IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 27, 2018, accepted June 11, 2018, date of publication June 15, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2847910

Virtual Network Embedding Using Node Multiple Metrics Based on Simplified ELECTRE Method

PEIYING ZHANG[1], **HAIPENG YAO**[2], **CHAO QIU**[2], **AND YUNJIE LIU**[2]
[1]College of Computer and Communication Engineering, China University of Petroleum (East China), Qingdao 266580, China
[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Peiying Zhang (25640521@qq.com)

This work was supported in part by the Fundamental Research Funds for the Central Universities of the China University of Petroleum (East China) under Grant 18CX02139A, in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2014FQ018, in part by the National Natural Science Foundation of China under Grant 61471056, in part by the National Basic Research Program (973) of China under Grant 2012CB315801, and in part by the China Research Project on Key Technology Strategy of Infrastructure Security for Information Network Development.

ABSTRACT The concept of network virtualization has attracted significant attention from academia to industry. One of the key challenges in network virtualization is the resource allocation problem, which is also termed the virtual network embedding (VNE) problem. It involved with mapping virtual networks onto a substrate network by adhering to some constraints, such as CPU capacity, on the nodes and bandwidth resources on the links. However, prior heuristic VNE algorithms mostly concentrate on measuring the embedding potential of substrate nodes using the multiplication of different nodes' resource metrics. Due to the fact that different resource metrics have different impacts on node ranking, these traditional methods have some limitations that would cause unbalanced embedding problems. Furthermore, the number of hops for the substrate paths that virtual links are mapped onto will have a large impact on the resource utilization of substrate links in a substrate network. In this paper, based on the topology analysis of six situations, we first propose a novel five-node ranking metric to quantify the importance of substrate nodes. Then, we give a comprehensive measurement method for substrate nodes using the simplified ELECTRE method to avoid an unbalanced embedding solution. We present a novel two-stage VNE algorithm, which chooses the substrate nodes with the maximum embedding potential to perform the node mapping procedure, and uses the shortest path algorithm to accomplish the link mapping procedure. Extensive simulation results demonstrated that our proposed method behaves better than the other state-of-the-art algorithms in terms of the long-term average revenue, the revenue-to-cost (R/C) ratio, and the VN request acceptance ratio.

INDEX TERMS virtual network embedding, node mapping procedure, link mapping procedure, node ranking metric.

I. INTRODUCTION

Over the past decades, the current Internet has achieved great successes. However, due to the coexistence of multiple Internet Service Providers (ISPs) with the contradictory purposes and strategies, the deployment and installation of new Internet services and protocols on the current Internet architecture are increasingly more and more difficult, which is called as Internet ossification. To fend off the Internet ossification and satisfy the demands of increasing number of diverse applications with various Quality of Service (QoS) requirements, virtual network embedding (VNE) has been propounded as a building block for the future Internet architecture, which has exerted a tremendous fascination on many researchers.

In the network virtualization environments (NVEs), traditional internet service providers (ISPs) are separated into infrastructure providers (InPs) and service providers (SPs). InPs are in charge of maintaining substrate network infrastructures, while SPs perform the role of providing the customized end-to-end network services. Specifically, SPs create heterogeneous virtual networks through aggregating distributed or centralized resources from multiple InPs with an aim to provide diverse services. VNE is a process in which mapping the virtual network requests onto the substrate network infrastructures with the constraints of CPU capacity on nodes and bandwidth resource on links. Due to the constraints of nodes and links, the VNE problem is proved to be an NP-hard problem. Due to its considerable runtime when in medium-size or large-size substrate network, a variety of heuristic algorithms have been proposed to address this issue, with the aim of finding the practical embedding solutions [1]–[3].

The VNE problem typically consist of two stages: 1) node mapping stage where virtual nodes from virtual network requests (VNRs) are mapped onto substrate nodes meanwhile satisfying the constraints such as CPU capacity on nodes; 2) link mapping stage where virtual links connecting to these virtual nodes are mapped onto the substrate paths meanwhile satisfying the constraints such as bandwidth demand on links. Prior works mainly concentrate on the greedy node mapping algorithms with the aim of giving priority to these substrate nodes with more embedding potential. However, these traditional VNE algorithms measure the node importance simply by the product of their CPU capacity and the total amount of bandwidth resources for their directly connected links. Some heuristic methods incorporate the topological attributes of the substrate network into the node importance ranking process. The aim of these methods is to give embedding priority to the substrate node with the biggest embedding potential [4]–[6]. This would lead to the imbalance problem of these metrics and decrease the resource utilization of substrate network.

ELECTRE is a family of multi-criteria decision analysis methods that originated in Europe in the mid-1960s. The simplified ELECTRE method can increase the computation efficiency and reduce the order complexity, without compromising the algorithm performance. Therefore, we adopt the simplified ELECTRE method to choose the most appropriate substrate node for virtual node in our node mapping process. The detailed description can be found in Section 4.3.

The main contributions and our main ideas are summarized as follows:

1. We define five metrics of node ranking using multiple attributes of substrate nodes in the substrate network. These five metrics can reflect the different aspects of substrate nodes in the substrate network, and facilitate the node mapping procedure from different perspectives.

2. Based on the simplified ELECTRE method, we devise a two-stage VNE algorithm, which is called ELECTRE-VNE, with multiple metrics of node ranking. In the stage of node mapping, based on simplified ELECTRE method, we use multiple node importance ranking metrics to address the issue of the imbalance problem on different evaluation metrics. In the stage of link mapping, we employ the shortest path algorithm to perform the link mapping procedure.

3. Extensive simulations demonstrated that our method is better than the other traditional methods in terms of the long-term average revenue, the revenue to cost (R/C) ratio, and the VN request acceptance ratio.

The reminder of this paper is organized as follows. Section 2 reviews the existing methods for VNE. Section 3 introduces the network model and problem statement. In Section 4, based on the multiple attributes of substrate nodes in the substrate network, we present the multiple metrics of node ranking. In Section 5, we describe our proposed method ELECTRE-VNE in detail. The performance of our method and other methods is evaluated in Section 6. Section 7 concludes this paper.

II. RELATED WORKS

Due to the constraints of nodes and links, the VNE problem is an NP-hard problem even when the topologies of virtual network requests (VNRs) are known in advance. Fischer *et al.* [7] presented a survey of current studies in the VNE area and introduced a VNE taxonomy. Based on whether the topologies of VN requests are known or not in advance, the VNE algorithms can be classified into two categories. One is the online VN embedding algorithms, the other is the offline VN embedding algorithms. Due to the fact that knowing all the VNRs in advance is not practical in the real situation, most of researchers advocate the online VN embedding algorithms. Given that the VN embedding problem is NP-hard [8], existing approaches can be roughly divided into three categories: (i) the optimal algorithms based on solving the integer linear programming (ILP) formulation; (ii) the heuristic algorithms based on various node resource estimation methods; and (iii) the meta-heuristic algorithms based on particle swarm optimization or memetic algorithms. In this section, we will review some prior studies in terms of these three aspects.

A. OPTIMAL ALGORITHMS

The most typical optimal algorithms is a VNE algorithm based on subgraph isomorphism detection [9]. The authors embedded nodes and links during the same stage. In the same year, Chowdhury *et al.* [10] addressed the VN embedding problem with the aim of coordinating two mapping stages including node mapping and link mapping. They constructed an augmented substrate graph based on the node location constraints, and formulated the VN embedding problem as a mixed integer linear programming problem. The authors designed two VN embedding algorithms *D-ViNE* and *R-ViNE* using deterministic and randomized techniques, respectively. Wang *et al.* [11] presented a compact path-based integer linear programming model to tackle with the VNE problem. Additionally, they proposed a branch-and-price framework that embeds a column generation process to address the formulated model. The simulation results demonstrated that the proposed framework can lead to the optimal or near optimal solution. However, this type of algorithms can only deal with the small size of topology for the substrate network and virtual networks due to its exponentially increasing computation time. The studies in [12] formulated an integer linear programming model to address the online VNE issue with the purpose of minimizing the resource consumption and balancing the network load. Melo *et al.* [12] presented three cost functions striving for the minimization of resource consumption and load balancing. Experimental results indicated that the Weighted Shortest Distance Path (*WSDP*) was the one which considered to be the optimal cost function. The work in [13] formulated an integer programming model to deal with the energy aware VNE for the purpose of coordinating node mapping and link mapping. The authors proposed two different objective functions to address the resource consumption and energy consumption, respectively. The classical and exact VNE algorithm was proposed by the work in [14], where the authors utilized the max-flow/min-cut approach to address the two InPs cases, and then extended them to multiple InPs cases. In their work, the authors employed the branch and bound algorithm, so as to solve the MIP program to provide the exact VN embedding solution with simultaneous mappings of nodes and links.

Due to its considerable runtime when in medium-size or large-size substrate networks, our work mainly concentrates on the uncoordinated VNE heuristic algorithm to address the VNE problem. Therefore, we do not make comparison with these exact or optimal algorithms [10], [14], [15]. Instead, we will address these issues in our future work.

B. HEURISTIC ALGORITHMS

Due to the fact that the optimal VN embedding algorithms would consume a large amount of computation time, some works on VNE mainly focus on the heuristic algorithms that consider the local resources of the nodes or the topological information of the substrate network more or less. The most classical heuristic VN embedding algorithm is *Greedy-VNE* proposed in [16]. The authors used the local resources of nodes to measure the node importance and employed the shortest path algorithm to perform the link mapping. Subsequently, inspired by the PageRank theory, Cheng *et al.* [17] presented a novel node ranking method using Markov Random Walk (RW) to improve the node ranking method. The authors devised two VN embedding algorithms, which were called RW-MaxMatch and RW-BFS. Extensive experimental results demonstrated that the topology-aware node ranking method was better than the classical resource evaluation method. The work in [5] exploited the topological information of substrate network and virtual networks, and introduced the network centrality analysis and the closeness analysis into the VNE process, by proposing two embedding algorithms to deal with the node ranking problem of substrate nodes. The improved closeness algorithm can dynamically measure the importance of substrate nodes in the substrate network and can increase the revenue of InPs in the long run. Botero and Hesselbach [18] modified and improved the exact existing energy aware VNE algorithms where the objective is to power off as many network nodes and interfaces as possible by consolidating the virtual networks into a subset of active physical networking equipment.

Wang and Hamdi [19] presented an efficient online VNE algorithm and formulated a new multiple objective linear programming optimization problem, and divided it into two stages including node mapping and link mapping. Wang and Hamdi [19] used Blocking Island (*BI*) to address the efficient resource allocation problem. The main aim of the proposed method *Presto* was to maximize the revenue of InPs and minimize the embedding cost of the VNRs. Hesselbach *et al.* [20] defined an optimization strategy using paths algebra method to address the linear or non-linear parameters of substrate nodes and links in substrate network, and proposed two novel algorithms called *NPA* and *I-NPA* to deal with the VNE problem in a coordinated node and link mapping manner.

Our work is similar to one in [21], the major difference is that our method present five metrics of node importance estimation based on the topology analyses of the substrate network, these metrics are different from those in the work [21]. In addition, we take use of simplified ELECTRE method to calculate the node ranking values of substrate nodes in the substrate network, whose computation complexity is lower than the TOPSIS based on the computation time complexity analysis. Different from [22], where the authors mainly used the product of multiple metrics of substrate nodes in the substrate network to estimate the node importance, thereby determining the embedding sequences of substrate nodes. Our work takes use of multiple metrics of node importance estimation to give each substrate node a comprehensive estimation value using simplified ELECTRE method, hence, they are essentially different.

C. META-HEURISTIC ALGORITHMS

As the optimal solution for large instances is difficult to find, meta-heuristics such as simulated annealing [23], genetic algorithm [24], ant colony optimization [25] or particle swarm optimization [2] can be used to find near optimal solutions by iteratively improving a candidate solution. The authors of [26] employed ant colony optimization meta-heuristic algorithm to deal with the VNE problem with the aim of minimizing the rejection rate of requests and maximizing returns for the substrate network provider. Zhang et al. [2] presented a unified enhanced particle swarm optimization method to address the VNE issue aiming to increase the acceptance ratio of VNs and the revenue of InPs by optimizing VN embedding costs. Zhang *et al.* [27] utilized multi-objective enhanced particle swarm optimization method to minimize the energy consumption of the substrate network by consolidating the major load into the a small number of active substrate nodes and links. Infhr and Raidl [28] introduced the memetic algorithm into the VNE problem, and studied the influence of diverse kinds of hybrid techniques on the VNE problem.

The advantage of these meta-heuristic algorithms is that it leverages the particle swarm optimization algorithm to improve a candidate solution. However, the weakness of these algorithms is that it would consume a large amount of running time. This type of algorithms is a trade-off between exact VNE algorithms and heuristic VNE algorithms. Our work only focuses on the heuristic VNE algorithm, therefore the comparison with these algorithms is beyond our research work. We intend to do this work in our future work.

III. NETWORK MODEL AND PROBLEM STATEMENT

A. SUBSTRATE NETWORK MODEL

We model the substrate network as a weighted undirected graph and denote it by $G^s = \{N^s, L^s\}$, therein, N^s and L^s represent the set of substrate nodes and the set of substrate links, respectively. Each substrate node $n^s \in N^s$ is characterized by its functional or non-functional attributes such as CPU capacity, storage capacity, and geographic location. Each substrate link $l_s \in L^s$ is characterized by its communication capacity such as bandwidth capacity. For each substrate link $l_s(i,j) \in L^s$, therein, i and j represent the two ends of the substrate link $l_s(i,j)$, we use $BW(l_s)$ to denote the total amount of available bandwidth resources. We denote the set of all the substrate paths by P_s, and denote the set of substrate paths from the source node s to destination node t by $P_s(s,t)$.

The left part of Fig. 1 illustrates a substrate network. The numbers over the links represent the total bandwidth resources and the residual bandwidth resources separating by a vertical line. The numbers aside the nodes represent the total CPU capacity in the first rectangular box and residual CPU capacity in the second rectangular box.

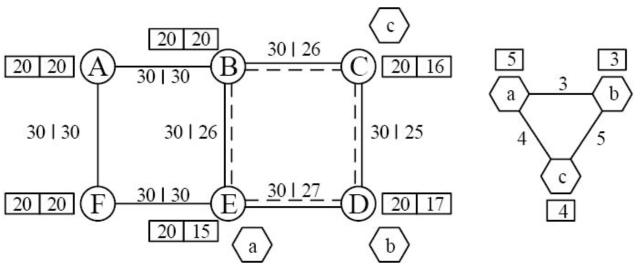


FIGURE 1. The diagram of substrate network and a virtual network request.

B. VIRTUAL NETWORK MODEL

Similarly, a virtual network can also be modeled as a weighted undirected graph and denoted by $G^v = \{N^v, L^v\}$, therein, N^v and L^v represent the set of virtual nodes and the set of virtual links in each VNR, respectively. For a virtual node $n_v \in N^v$, its computing demand can be expressed by $CPU(n_v)$. For a virtual link $l_v \in L^v$, its required bandwidth resource can be expressed by $BW(l_v)$. A virtual network request which consists of three virtual nodes and three virtual links is illustrated in the right part of Fig. 1. The number in the rectangular box which aside the node represents the demanded CPU capacity of the node, and the number over the link represents the required bandwidth resource of the link.

C. VIRTUAL NETWORK EMBEDDING PROBLEM DESCRIPTION

The VNE problem can be modeled as a mapping $M : G^v\{N^v, L^v\} \rightarrow G^s\{N^{s'}, P_s\}$ from G^v to a subset of G^s, therein, $N^{s'} \subset N^s$. The mapping process is typically decomposed of two mapping steps: (i) node mapping stage which assigns the virtual nodes to the heterogeneous substrate nodes meanwhile satisfying the resource constraints on the nodes; and (ii) link mapping stage which assigns the virtual links to a loop-free substrate paths on the substrate links meanwhile satisfying the resource constraints on the links.

The left part of Fig. 1 indicates a VNE solution for a VNR which depicted in the right part of Fig. 1. The mapping solution can be represented by node mapping solution $\{a \rightarrow E, b \rightarrow D, c \rightarrow C\}$ and link mapping solution $\{P_v(a,b) \rightarrow P_s(E,D), P_v(b,c) \rightarrow P_s(D,C), P_v(a,c) \rightarrow P_s(E-B-C)\}$.

D. OBJECTIVES

The main goal of VNE is how to make efficient use of the limited substrate network resources to accommodate as many VNRs as possible so as to obtain more revenue from the InPs' point of view. Generally, there are three main objectives to measure the performance of VNE algorithms, i.e., the long-term average revenue, the long-term revenue to cost (R/C) ratio and the VN request acceptance ratio. Similar to the previous studies [15], [16], for the InPs, the obtained revenue of accepting a VNR at time t can be defined as the total amount of network resources that VN request required, which can be formulated as follows:

$$R(G^v, t) = \sum_{n_v \in N^v} CPU(n_v) + \sum_{l_v \in L^v} BW(l_v), \tag{1}$$

where $CPU(n_v)$ represents the CPU capacity for the virtual node n_v, $BW(l_v)$ represents the amount of bandwidth resource requirement for the virtual link l_v.

The embedding cost of accommodating a VN request G^v at time t can be defined as the total amount of substrate network resources that allocated to the VNR, which can be formulated as follows:

$$C(G^v, t) = \sum_{n_v \in N^v} CPU(n_v) + \sum_{l_v \in L^v} BW(l_v) \times Hops(l_v), \tag{2}$$

where $Hops(l_v)$ represents the number of hops for the substrate path corresponding to the virtual link l_v.

Similar to the previous literature [16], the long-term average revenue can be defined as the limit of the average revenue when T trends to infinity, which can be formulated as follows:

$$R = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} R(G^v, t)}{T}. \tag{3}$$

The long-term average cost can be defined as the limit of the average cost when T trends to infinity, which can be formulated as follows:

$$C = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} C(G^v, t)}{T}. \tag{4}$$

The long-term revenue to cost (R/C) ratio can be formulated as follows:

$$R/C = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} R(G^v, t)}{\sum_{t=0}^{T} C(G^v, t)}. \tag{5}$$

The VN request acceptance ratio can be defined as follows:

$$acceptance\ ratio = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} VNR_{success}}{\sum_{t=0}^{T} VNR_{request}}, \tag{6}$$

where $VNR_{success}$ represents the number of VN requests which are successfully mapped onto the substrate network, $VNR_{request}$ represents the number of arrived VN requests.

Generally, we take use of the long-term average revenue to measure the performance of the algorithms with the purpose of maximizing the revenue of InPs and increasing the resource utilization of substrate network in the long run. If the long-term average revenue of the VNE algorithms is almost the same, we make use of the long-term revenue to cost (R/C) ratio to quantify the efficiency of resource utilization of substrate network. In addition to these above two metrics of the algorithms, we can also take use of the VN request acceptance ratio to distinguish the compared methods.

IV. THE EVALUATION METRICS OF NODE RANKING BASED ON MULTIPLE ATTRIBUTES

A. MOTIVATIONS

In the node mapping stage, most of heuristic algorithms are based on the resource evaluation metric of node ranking to give mapping priority to the substrate nodes with the larger metric values. Several greedy node mapping methods measure the embedding potential for each substrate node aiming to determine the mapping sequence of the substrate nodes. Therefore, the evaluation metric of node ranking has a significant influence on the performance of VNE algorithms. The authors of [16] proposed a classical evaluation metric of node ranking which measures the node resource availability by the product of the node CPU capacity and the total amount of the available bandwidth resources of its outgoing links. The evaluation metric of node ranking can be formulated as Eq. (7), and most of VNE methods utilize the same node ranking method to perform the node mapping procedure.

$$H(n) = CPU(n) \times \sum_{l \in neighbor(n)} BW(l), \tag{7}$$

where $H(n)$ represents the resource evaluation metric of node ranking for the node n, $CPU(n)$ represents the CPU capacity of the node n, $BW(l)$ represents the available bandwidth resource of the link l, and $neighbor(n)$ represents the set of links which directly connect to the node n.

However, this evaluation metric of node ranking has the following drawbacks.

First, it only takes the local resource metrics of the nodes into consideration while ignoring the resource metrics of their neighborhood nodes, which may lead to the mapping failure during the subsequent link mapping process. For example, as illustrated in Fig. 2, where the numbers in the rectangular boxes next to the nodes represent the CPU capacity of the nodes and the numbers over the lines represent the available bandwidth resources of the links. As demonstrated in Fig. 2, the evaluation metric of node B is calculated as $H(B) = 30*(40+40+40) = 3600$, and the evaluation metric of node F is calculated as $H(F) = 30*(40+40+40) = 3600$, hence, the node B and node F have the same resource evaluation metric of node ranking measured by Eq. (7). Nevertheless, the selection of substrate node F could have more opportunity

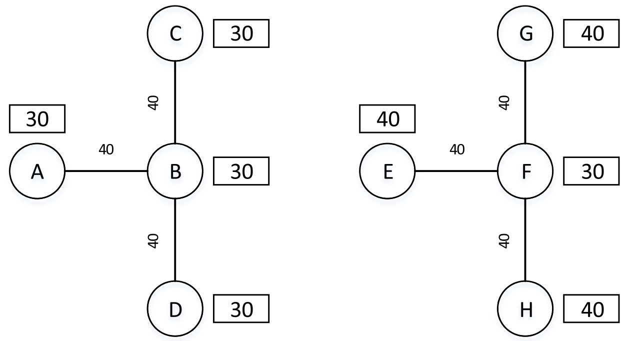


FIGURE 2. A motivational example 1 to illustrate the drawback of the metric $H(n)$.

to obtain the success of subsequent link mapping procedure, due to the fact that the CPU capacity of its neighborhood nodes for node F is more than the corresponding CPU capacity of its neighborhood nodes for node B. Therefore, we assume that the embedding potential of node F is more than that of node B.

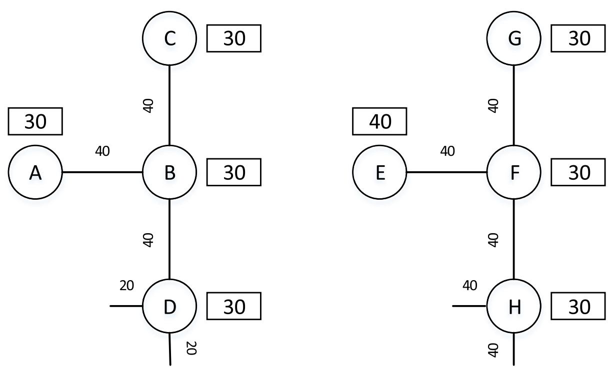


FIGURE 3. A motivational example 2 to illustrate the drawback of the metric $H(n)$.

Second, as illustrated in Fig. 3, we suppose that the CPU capacity of neighborhood nodes for node B is the same as node F, but the node D has less bandwidth resources than the node H. Apparently, mapping a virtual node onto the node F is better than mapping it onto the node B since the node H has more local resources than the node D in terms of its bandwidth resources.

Third, the aforementioned evaluation metric of node ranking ignores the bandwidth resource constraints on the links and will be prone to cause the failure of the subsequent link mapping procedure. As demonstrated in Fig. 4, a VN request is shown in the left side of Fig. 4, provided that the virtual node a is already mapped onto the substrate node A, the substrate nodes B and E are two nodes with the second largest evaluation metric of node ranking, both of their evaluation metric values are 1800, i.e., $H(B) = 30 * (20 + 20 + 20) = 1800$, $H(E) = 60 * (10 + 10 + 10) = 1800$, if we map the virtual node b onto the substrate E, the subsequent

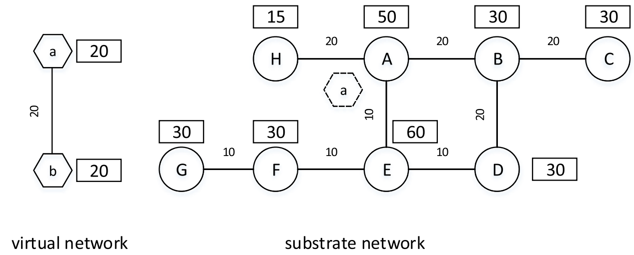


FIGURE 4. A motivational example 3 to illustrate the drawback of the metric $H(n)$.

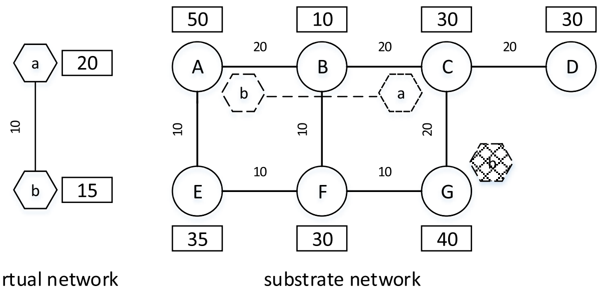


FIGURE 6. A motivational example 5 to illustrate the drawback of the metric $H(n)$.

link mapping will be fail due to the fact that the available bandwidth resources of substrate path denoted by $P_s(A, E)$ are lesser than the required bandwidth resources of the virtual link denoted by $l_v(a, b)$. Hence, not only the CPU capacity of the node and the amount of available bandwidth resources of its outgoing links should be taken into consideration, but also the minimum bandwidth requirement of virtual link which connects the already mapped virtual node and the mapping virtual node should be emphasized.

Fourth, the classical evaluation metric of node ranking only considers the local resources of the nodes while regardless of the topological attribute influence of the substrate network, which cannot give each node a comprehensive metric value. As illustrated in Fig. 5, in addition to the resource evaluation metrics of its neighborhood nodes and its required bandwidth resources, the node degree should also be taken into consideration. For instance, the substrate nodes B and F have the same evaluation metric values based on the above two metrics, but the degree of substrate node B is $degree(B) = 3$, the degree of substrate node F is $degree(F) = 4$, it means that mapping the virtual node onto the substrate node F has more opportunity to obtain the success during the subsequent node mapping and link mapping process. Therefore, with the purpose of improving the performance of VNE algorithm, the degree of substrate nodes should also be incorporated into the evaluation metric computation process.

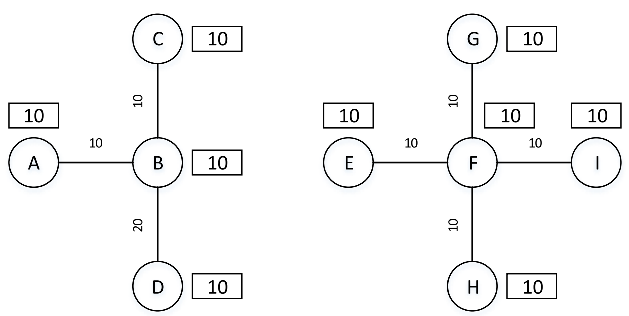


FIGURE 5. A motivational example 4 to illustrate the drawback of the metric $H(n)$.

Fifth, for each solution of VN request, the allocated CPU capacity over the substrate nodes for virtual nodes is constant, the difference between two mapping solutions is the allocated bandwidth resource over the substrate paths for virtual links from each VNR. To achieve a higher resource utilization of the substrate network aiming at accommodating more VNRs, and thereby improving the profitability of the InPs, we should map the two adjacent virtual nodes onto the two substrate nodes which are not far away from each other so as to reduce the unnecessary bandwidth consumption of substrate links and decrease the resource fragmentation of the substrate network. As illustrated in Fig. 6, the node mapping sequences of virtual nodes and substrate nodes computed by Eq. (7) in the virtual network and substrate network are $n_a^v > n_b^v$ and $n_C^s > n_A^s > n_G^s > n_F^s > n_E^s > n_D^s > n_B^s$, respectively. Based on the greedy node mapping strategy, the node mapping result is $M_N(n_a^v) = n_C^s$, $M_N(n_b^v) = n_A^s$, therein, $M_N()$ represents the node mapping function. When the virtual node n_a^v has been mapped onto the substrate node n_C^s, the next virtual node n_b^v will be mapped onto the substrate node n_A^s because the substrate node n_A^s is the unmapped substrate node with the largest node ranking value measured by Eq. (7). However, the substrate node n_G^s is the most appropriate candidate substrate node although it has the lower node ranking value than the substrate node n_A^s. Mapping the virtual node n_b^v onto the substrate node n_G^s will consume less bandwidth resource for virtual link $l_v(a, b)$ during the subsequent link mapping process. The prerequisite is that the substrate node n_G^s has enough CPU capacity resources to satisfy the computing demand of virtual node n_b^v. Therefore, the number of hops between the mapping substrate node and the set of already mapped substrate nodes has a significant effect on the resource utilization of substrate network and should be incorporated into the calculating resource evaluation metric of node ranking.

Sixth, apart from these above mentioned aspects which need to be considered in the node mapping stage, the CPU utilization ratio of the substrate node is crucial to the performance of VNE algorithm, as depicted in Fig. 7. Provided that we are mapping the virtual node n_a^v, the substrate nodes n_B^s and n_E^s are two candidate substrate nodes that can be mapped onto, the CPU demand of the virtual node n_a^v is $CPU(n_a^v) = 10$, mapping the virtual node n_a^v onto the substrate node n_B^s will consume all of its CPU capacity resources and almost cut off the connection between substrate nodes n_A^s and n_C^s, and will generate network resource fragmentation. This situation

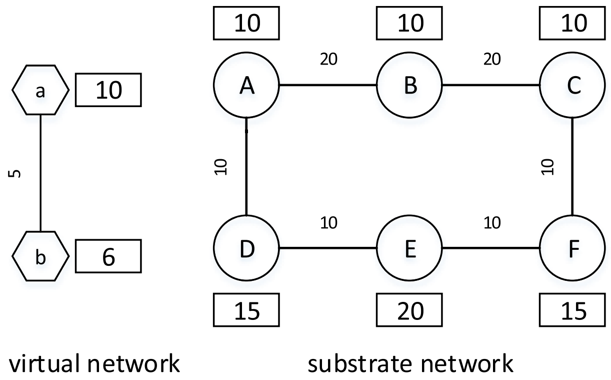


FIGURE 7. A motivational example 6 to illustrate the drawback of the metric $H(n)$.

may lead to the failure of subsequent link mapping. Therefore, the CPU utilization ratio of the substrate node has a significant effect on the greedy node mapping algorithm, and must be taken into consideration to avoid this situation.

B. THE EVALUATION METRIC OF NODE RANKING ANALYSIS

Since different evaluation metrics of node ranking will lead to different node mapping sequences, single evaluation metric of node ranking or the multiplication by several evaluation metrics will cause imbalanced evaluation of embedding potential for the substrate nodes, and result in the lower resource utilization of the substrate network. In this section, we introduce some definitions to measure substrate node importance aiming to facilitate the node mapping process.

Definition 1: Resource Capacity can be defined as the sum of node resource evaluation metric and the resource evaluation metric values obtained from its neighborhood nodes.

The resource capacity value of substrate node n_i can be formulated as follows:

$$RC(n_i) = H(n_i) + \sum_{n_j \in nbr(n_i)} H(n_i) \times \frac{BW(l_{ij})}{\sum_{n_k \in nbr(n_j)} BW(l_{jk})}, \quad (8)$$

where $RC(n_i)$ represents the resource capacity of the node n_i, $nbr(n_i)$ represents the set of its neighborhood nodes which directly connected to the node n_i. If $n_i \in N^v$, $H(n_i)$ represents the resource evaluation metric value for the virtual node n_i, $BW(l_{ij})$ represents the required bandwidth resource of the virtual link l_{ij}. If $n_i \in N^s$, $H(n_i)$ represents the available resource evaluation metric value for the substrate node n_i, $BW(l_{ij})$ represents the available bandwidth resources of the substrate link l_{ij}. Note that we take the bandwidth resource account for the proportion of total bandwidth resource as the probability, and incorporate the probability information into the Eq. (8). The definition 1 can deal with the first and second situations.

Definition 2: Modified Resource Evaluation Value can be defined as the multiplication of the node CPU capacity and the total amount of bandwidth resource of its adjacent links whose available bandwidth resources are more than the minimum demand of the virtual links.

The modified resource evaluation value of the node can be formulated as follows:

$$MREV(n_i) = CPU(n_i) \times \sum_{l \in nbr(n_i) \wedge BW(l) \geq \delta} BW(l), \quad (9)$$

where $MREV(n_i)$ denotes the modified resource evaluation value of the node n_i, $CPU(n_i)$ denotes the available CPU capacity of the node n_i, $nbr(n_i)$ represents the set of adjacent links for the node n_i, $BW(l)$ represents the available bandwidth resources of the link l, δ represents a threshold which is set in advance or the minimum bandwidth resource requirement. The definition 2 can deal with the third situation.

Definition 3: Node Degree can be defined as the number of its outgoing links.

The node degree for the node n_i can be formulated as follows:

$$ND(n_i) = degree(n_i), \quad (10)$$

where $ND(n_i)$ represents the degree of the node n_i, $degree(n_i)$ refers to the number of the outgoing links of the node n_i. There are some studies [22], [29] that resort to the other centrality metrics such as closeness centrality, betweenness centrality, eigenvector centrality, Katz centrality and so on. The definition 3 can deal with the fourth situation.

Definition 4: The HOPS between the mapping substrate node and the set of already mapped substrate nodes can be defined as the minimum number of hops between the mapping substrate node and any mapped node from the set of already mapped substrate nodes.

The HOPS between the mapping substrate node and the set of already mapped substrate nodes can be expressed as follows:

$$HOPS(n_i, \Omega) = min_{n \in \Omega} HOPS(n_i, n), \quad (11)$$

where n_i represents the mapping substrate node, Ω represents the set of already mapped substrate nodes, $HOPS(n_i, \Omega)$ represents the hops between the mapping substrate node n_i and the set of already mapped substrate nodes Ω. The definition 4 can deal with the fifth situation.

Definition 5: The CPU Utilization Ratio of the substrate node can be defined as the ratio between the demanded CPU capacity of each virtual node and the available CPU capacity of the substrate node n_i^s.

$$UR(n_i^s) = \frac{CPU(n_i^v)_{required}}{CPU(n_i^s)_{available}}, \quad (12)$$

where $UR(n_i^s)$ represents the CPU utilization ratio of the substrate node n_i^s, the subscript $_{required}$ and $_{available}$ represent the required CPU capacity for the virtual node n_i^v and the available CPU capacity for the substrate node n_i^s, respectively. The definition 5 can deal with the sixth situation.

C. SIMPLIFIED ELECTRE ALGORITHM

The ELECTRE is a notable classical method of multiple attribute decision making which was first proposed in [30], and Roy and Vanderpooten [31] reported on the works of the European consultancy company SEMA with respect to a specific real world problem. There are many multiple attribute decision-making methods such as TOPSIS [32], ELECTRE and so on. The reason we choose simplified ELECTRE is that its time complexity can satisfy our needs. In this section, the substrate node is regarded as a solution, the multiple evaluation metric values of substrate nodes are regarded as solution attributes, and then the comprehensive metric of substrate node importance can be transformed into a multiple attribute decision making problem.

The steps of the simplified ELECTRE algorithm are as follows:

Step1: Evaluate each solution denoted by $a_1, a_2, \ldots, a_n$, where each solution has m evaluation criteria. We calculate the metric values of evaluation criteria for each solution a_{ij}, and we can obtain a decision making matrix and formulate it as follows:

$$A_{n\times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}, \tag{13}$$

where each element a_{ij} represents the metric value of *j-th* evaluation criterion of *i-th* solution.

Step2: Normalize the decision making matrix. The normalized matrix denoted by R can be obtained through the normalization of column vectors, aiming to eliminate the impacts of different metric values in different dimensionalities.

$$R_{n\times m} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}, \quad therein,\ r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{n} a_{ij}^2}}. \tag{14}$$

Step3: Calculate the weighted normalized decision making matrix. The weight of each criterion can be formulated as a weighted vector denoted by $\vec{W}(j = 1, 2, \ldots, m)$, where w_j represents the weight coefficient of *j-th* evaluation criterion to measure the importance of each metric, and must satisfy the constraint equation $\sum_{j=1}^{m} w_j = 1$. The weighted normalized decision making matrix V can be defined as follows:

$$V_{ij} = r_{ij} \cdot w_j. \tag{15}$$

Step4: Calculate the consistent matrix and non-consistent matrix.

(i) To compare the two element values in any two different rows in the weighted normalized matrix V, if the *i-th* row value is larger than the *j-th* row value in *k-th* column, the k can be grouped into a consistent set C_{ij}, otherwise the k can be grouped into a non-consistent set D_{ij}, where $k = 1, 2, \ldots, m$. The consistent set and non-consistent set can be formulated as follows:

$$C_{ij} = \{k | v_{ik} \geq v_{jk}\} \& D_{ij} = \{k | v_{ik} \leq v_{jk}\}. \tag{16}$$

(ii) To calculate the consistent matrix. The consistent matrix C can be obtained by adding the weighted value of each element in each consistent set. The formulation of the consistent matrix C can be defined as follows:

$$C = [c_{ij}]_{n\times n}, c_{ij} = \frac{\sum_{k\in C_{ij}} w_k}{\sum_{k=1}^{m} w_k}, \tag{17}$$

where c_{ij} represents the relative dominating index of the solution a_i compared to the solution a_j.

(iii) To calculate the non-consistent matrix. We first compute the maximum value of the differences between any pair of the weighted values for each element corresponding to two solutions. Then we divided it by the maximum value of the differences between any pair of the weighted values for each element corresponding to all of the solutions. Finally, we can obtain the relative inferior value of the two solutions. The non-consistent matrix can be expressed as follows:

$$D = [d_{ij}]_{n\times n}, d_{ij} = \frac{\max\limits_{k\in D_{ij}} |w_k(a_{ik} - a_{jk})|}{\max\limits_{k\in S} |w_k(a_{ik} - a_{jk})|}, \tag{18}$$

where d_{ij} represents the relative inferior index of the solution a_i compared to the solution a_j. We take the weight information of the index into consideration when calculating the relative inferior index. Relative to c_{ij} which only contains the weight information of the index, d_{ij} increases the differences between two index values, thus, not only contains the weight information, but also has the index information. Furthermore, the relative dominating index and the relative inferior index are not complementary. d_{ij} can reflect the relative inferiority of the solution a_i compared to the solution a_j. The smaller value means the lower inferiority degree of the solution a_i compared to the solution a_j.

(iv) To calculate the modified non-consistent matrix. Hwang and Masud [33] redefined the non-consistent matrix, the formulation can be defined as follows:

$$D' = [d'_{ij}]_{n\times n}, d'_{ij} = 1 - d_{ij}. \tag{19}$$

Step5: Calculate the modified weighted summation matrix. The non-consistent matrix in the traditional ELECTRE method can be modified in order to make the element value in the modified non-consistent matrix and the element value in the modified consistent matrix have the same value. The greater the value, the higher the preference degree is. Therefore, we can take use of the product of these two element values between the element value in consistent matrix and the corresponding element value in non-consistent matrix to obtain the modified weighted summation matrix, and formulate it as follows:

$$E = [e_{ij}]_{n\times n}, e_{ij} = c_{ij} \cdot d'_{ij}. \tag{20}$$

Step6: Calculate the net dominating value. The concept of net dominating value is put forward by Van Delft and

Nijkamp in 1976. The net dominating value can be defined as follows:

$$C_k = \sum_{i=1 \wedge i \neq k}^{n} e_{ki} - \sum_{j=1 \wedge j \neq k}^{n} e_{jk}, \tag{21}$$

where C_k represents the weighted sum of the solution a_k to the other solutions minus the weighted sum of the solution a_i, it can reflect the weighted sum of the net dominating value for the solution C_k. The bigger the value of C_k is, the better the solution a_k is.

Step7: Sort the solutions by their weighted sum of the net dominating values. We sort the solutions by their weighted sum of the net dominating values, and can obtain the sequences of the solutions from the best to the worst.

In this paper, we take the resource capacity (RC), modified resource evaluation value (MREV), node degree (ND), hops (HOPS) and CPU utilization ratio (UR) as the evaluation criterions of substrate node importance. We take the reciprocals of HOPS and UR in order to make all of these evaluation criterions the larger the better. Then, we can obtain the ranking orders for substrate nodes through the simplified ELECTRE method.

D. A EXAMPLE FOR SIMPLIFIED ELECTRE

Considering the following topologies of the virtual network and the substrate network as illustrated in Fig. 8. We assume that the virtual node a has already mapped onto the substrate node A, the subsequent step is to choose another substrate node for virtual node b. For the sake of simplification, here we only annotate the available bandwidth resource for substrate links.

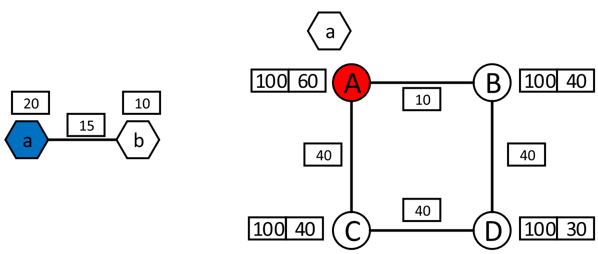


FIGURE 8. A illustration demo for simplified ELECTRE.

The computation of resource evaluation metrics for substrate nodes: $H(A) = 3000, H(B) = 2000, H(C) = 3200, H(D) = 2400$.

The computation of resource capacity for substrate nodes: $RC(A) = 5960, RC(B) = 4520, RC(C) = 5900, RC(D) = 5000$.

The computation of modified resource evaluation value for substrate nodes: $MREV(A) = 2400, MREV(B) = 1600, MREV(C) = 3200, MREV(D) = 2400$. Note that when mapping the virtual node b, the minimum required bandwidth resource value is 15. Therefore, when we compute the modified resource evaluation value for substrate nodes A and B, the virtual link $l_s(A, B)$ should be removed.

The computation of node degree for substrate nodes: $ND(A) = 2, ND(B) = 2, ND(C) = 2, ND(D) = 2$.

The computation of HOPS for substrate nodes: $HOPS(A) = 0, HOPS(B) = 3, HOPS(C) = 1, HOPS(D) = 2$. Note that the number of HOPS between substrate node B and the set of already mapped substrate nodes $\Omega = \{A\}$ is 3, due to the fact that the bandwidth resource of virtual link $l_s(A, B)$ cannot satisfy the minimum requirement.

The computation of CPU utilization ratio for substrate nodes: $UR(A) = \frac{1}{6}, UR(B) = \frac{1}{4}, UR(C) = \frac{1}{4}, UR(D) = \frac{1}{3}$.

The construction of decision making matrix: Since the substrate node A is allocated to virtual node a, the candidate substrate nodes do not contain the substrate node A. Note that each row represents five metrics of substrate nodes B, C, and D. Each column represents RC, MREV, ND, reciprocal of HOPS, and reciprocal of UR, respectively.

$$A_{3\times5} = \begin{bmatrix} 4520 & 1600 & 2 & 0.33 & 4 \\ 5900 & 3200 & 2 & 1 & 4 \\ 5000 & 2400 & 2 & 0.5 & 3 \end{bmatrix}. \tag{22}$$

The normalized the decision making matrix:

$$R_{3\times5} = \begin{bmatrix} 0.50 & 0.37 & 0.58 & 0.28 & 0.62 \\ 0.66 & 0.74 & 0.58 & 0.86 & 0.62 \\ 0.56 & 0.56 & 0.58 & 0.43 & 0.47 \end{bmatrix}. \tag{23}$$

The computation of weighted normalized decision making matrix: We set every weight coefficient to 0.2.

$$V_{3\times5} = \begin{bmatrix} 0.100 & 0.074 & 0.116 & 0.056 & 0.124 \\ 0.132 & 0.148 & 0.116 & 0.172 & 0.124 \\ 0.112 & 0.112 & 0.116 & 0.086 & 0.094 \end{bmatrix}. \tag{24}$$

The consistent matrix and non-consistent matrix are formulated as follows:

$$C_{3\times5} = \begin{bmatrix} - & 0.4 & 0.4 \\ 0.6 & - & 0.8 \\ 0.6 & 0.2 & - \end{bmatrix}. \tag{25}$$

$$D_{3\times5} = \begin{bmatrix} - & 1 & 1 \\ 0 & - & 0 \\ \frac{1}{800} & 1 & - \end{bmatrix}. \tag{26}$$

The modified non-consistent matrix can be formulated as follows:

$$D'_{3\times5} = \begin{bmatrix} - & 0 & 0 \\ 1 & - & 1 \\ 0.99875 & 0 & - \end{bmatrix}. \tag{27}$$

The modified weighted summation matrix can be formulated as follows:

$$E_{3\times5} = \begin{bmatrix} - & 0 & 0 \\ 0.6 & - & 0.8 \\ 0.5993 & 0 & - \end{bmatrix}. \tag{28}$$

The computation of the net dominating value for substrate nodes B, C, and D. $C_B = -.11993, C_C = 1.4, C_D = -0.2007$. We sort the solutions by their weighted sum of the net dominating values, and can obtain the sequences of the solutions from the best to the worst is: $C_C > C_D > C_B$. Therefore, we should map the virtual node b onto the substrate node C.

V. HEURISTIC ALGORITHM DESIGN

Based on the multiple attribute decision making method and five resource evaluation metrics, we propose a novel heuristic algorithm called as ELECTRE-VNE to deal with the VNE problem, which consists of node mapping process based on simplified ELECTRE method and link mapping process based on the shortest path algorithm.

A. NODE MAPPING ALGORITHM

The detailed steps of node mapping algorithm are illustrated in Algorithm 1. The proposed method is similar to the greedy node mapping approach, the only difference between them is the node ranking method for substrate nodes in the substrate network.

Algorithm 1 The Node Mapping Algorithm Based on Simplified ELECTRE Method

```
Sort the virtual nodes by H in non-increasing order.
for all the unmapped virtual nodes in VNR do
    choose a virtual node n_v with the highest H;
    for each substrate node n_s in substrate network do
        Calculate RC(n_s), MREV(n_s), ND(n_s), HOPS(n_s),
        and UR(n_s);
        Calculate the node ranking values for substrate
        nodes using simplified ELECTRE method;
        Sort the substrate nodes by their node ranking values
        in descending order and denote it by Ω;
        for each n_s in Ω do
            if n_s is not mapped and CPU(n_s) ≥ CPU(n_v)
            then
                M_N(n_v) = n_s;
                break;
            end if
        end for
    end for
end for
return true.
```

Note that we ranking the virtual nodes according to H not ELECRRE, the reason is that through conducting experiments we found that when the number of virtual nodes is less, using ELECTRE method would not improve the performance of node ranking than using H method but increasing the computation time. Therefore, we consist on ranking the virtual nodes according to H.

B. LINK MAPPING ALGORITHM

Here we use the shortest path algorithm to perform the link mapping procedure. The larger number of hops will consume a large amount of bandwidth resource but increase the VN request acceptance ratio. The less number of hops will cause the virtual network mapping failure due to the bandwidth resource bottleneck of substrate links but decrease the ratio of virtual network request acceptance. The detailed steps of link mapping algorithm are demonstrated in Algorithm 2.

Algorithm 2 The Link Mapping Algorithm Based on the Shortest Path Algorithm

```
Sort the virtual links by the required bandwidth in non-
increasing order.
for all the unmapped virtual links in VNR do
    fetch two corresponding substrate nodes n^s_start and n^s_end
    for the virtual link l_v;
    remove all the substrate links whose bandwidth
    resource is lesser than the required amount of band-
    width resource;
    choose a loop-free substrate path between n^s_start and
    n^s_end using shortest path algorithm.
end for
return true.
```

C. TIME COMPLEXITY ANALYSIS

We denote the number of virtual nodes and virtual links in each VN request by $|N^v|$, $|L^v|$, respectively; We denote the number of substrate nodes and substrate links in the substrate network by $|N^s|$, $|L^s|$, respectively. The average times of iteration in RW-VNE is denoted by $|itertimes|$.

The time complexity of Greedy-VNE: The time complexity of node mapping process is $O(|N^s|^2)$, and the time complexity of link mapping process is $O(|L^v||N^s|^2)$.

The time complexity of RW-VNE: The time complexity of node mapping process is $O(|itertimes| \times |N^s|^2)$, and the time complexity of link mapping process is $O(|L^v||N^s|^2)$.

The time complexity of IC-VNE: The time complexity of node mapping process is $O(|N^s|^3)$, and the time complexity of link mapping process is $O(|L^v||N^s|^2)$.

The time complexity of ELECTRE-VNE: The time complexity of node ranking computation for each VNR is $O(|N^v|^2)$. The time complexities of computing $RC(n_s)$, $MERV(n_s)$, $ND(n_s)$, $HOPS(n_s)$, and $UR(n_s)$ are $O(|N^s|^2)$, $O(|N^s|^2)$, $O(|N^s||L^s|)$, $O(|N^s|^3)$, and $O(|N^s|^2)$, respectively. Therefore, the time complexity of Algorithm 1 is $O(|N^s|^3)$. The time complexity of Algorithm 2 is $O(|L^v||N^s|^2)$.

VI. PERFORMANCE EVALUATION

In this section, we describe the settings of our simulation environment in detail, and then present our experimental results. We use the aforementioned three evaluation criteria including the long-term average revenue, the long-term R/C ratio and the VN request acceptance ratio to measure the performance of our method compared with the other methods.

A. ENVIRONMENT SETTINGS

Similar to the prior works [6], [15], [16], we use the GT-ITM [34] to generate the topologies of the substrate network and virtual networks. The number of substrate nodes in substrate network is set to 100, and the connectivity probability between any two substrate nodes is set to 0.5. The CPU capacities of substrate nodes in substrate network are real numbers which follow the uniform distribution between 50 and 100, the bandwidth resources of substrate links in substrate network are real numbers which follow the uniform distribution between 50 and 100. We assume that the arrival of VNR is a Poisson process, and the mean arrival rate is 5 VNRs/100 time units. The duration of each VNR follows negative exponential distribution with an average of 1000 time units. The demanded CPU capacities of virtual nodes in each VNR are real numbers which uniformly distributed between 0 and 50, the required bandwidth resources of virtual links in each VNR are real numbers which uniformly distributed between 0 and 50. The number of nodes in each VNR is uniformly distributed between 2 and 20, and the connectivity probability between any two virtual nodes is assigned to 0.5.

Our simulation experiments evaluate four methods, which are listed in Table 1. The Greedy-VNE algorithm [16] is the classical VN embedding algorithm, the RW-VNE algorithm [17] is a topology-aware node ranking VN embedding algorithm, the IC-VNE algorithm [5] is an approach of VN embedding based on network centrality analysis and closeness centrality, and the ELECTRE-VNE algorithm is our proposed method. In our work, we do not take into consideration the situation where link mapping solution supports the path splitting.

TABLE 1. The compared four methods.

Notation	Description
Greedy-VNE	Node mapping process based on classical node resource evaluation metric and link mapping process based on shortest path algorithm.
RW-VNE	Node mapping process based on the random walk (RW) algorithm and link mapping process utilizing shortest path algorithm.
IC-VNE	Node mapping process based on the node closeness and link mapping process utilizing shortest path algorithm.
ELECTRE-VNE	Node mapping process using node multiple metrics based on the simplified ELECTRE method and link mapping process utilizing shortest path algorithm.

B. EVALUATION RESULTS

The existing work generally set the maximum number of hops range from 3 to 7. In our work, we want to maximum the ratio of virtual network request acceptance and avoid unnecessary consumption of bandwidth resource in substrate network. Therefore, we set the maximum number of hops to 5 according our experiences. In order to fully evaluate our method compared with the other three methods from different perspectives, we carried out two experiments aiming to validate the effectiveness and feasibility of our method.

1) SIMULATION EXPERIMENT 1

The first experiment aims to evaluate the performance of our method compared with the other methods. We use the long-term average revenue, the long-term R/C ratio and the VN request acceptance ratio to measure the performances of these compared methods.

Fig. 9 shows the long-term average revenue of the compared four methods. From the Fig. 9, we can observe that our method ELECTRE-VNE is the highest one among these compared methods. According to quantitative analysis from the specific data, the long-term average revenue of our method ELECTRE-VNE is 22.01% higher than GREEDY-VNE, 14.66% higher than RW-VNE, and 6.83% higher than IC-VNE. The main reason is that ELECTRE-VNE incorporates the hops between the mapping substrate node and the set of already mapped substrate nodes into substrate node ranking computation process. Then we can avoid unnecessary bandwidth resource consumption so as to save more bandwidth resource to accommodate more virtual network requests, which leads to the largest long-term average revenue.

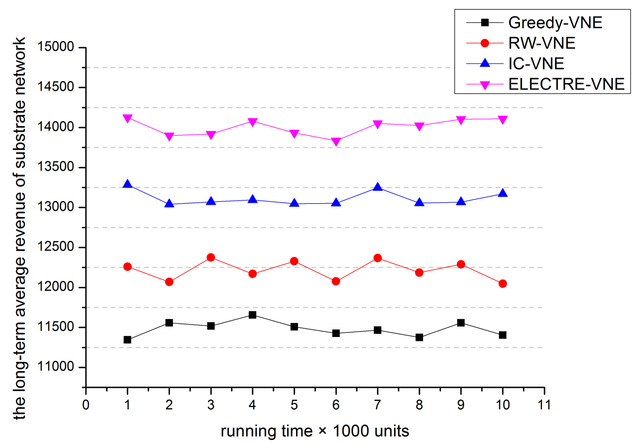


FIGURE 9. The long-term average revenue of substrate network.

Fig. 10 presents the R/C ratio of the compared four methods. From the Fig. 10, we can observe that our method ELECTRE-VNE is the optimal one among these four methods. According to the quantitative analysis from the obtained data, the R/C ratio of our method ELECTRE-VNE is almost 13.67% higher than GREEDY-VNE, 9.14% higher than RW-VNE, and 5.15% higher than IC-VNE. The main reason is that our method takes into consideration the number of the hops between the mapping substrate node and the set of already mapped substrate nodes, embeds the adjacent virtual nodes onto the two substrate nodes that are not far away from each other with the aim of reducing unnecessary bandwidth resource consumption. Therefore, it can make the R/C ratio higher than the other three methods.

Fig. 11 illustrates the VN request acceptance ratio of the compared four methods. From the Fig. 11, we can observe that the VN request acceptance ratio of our method

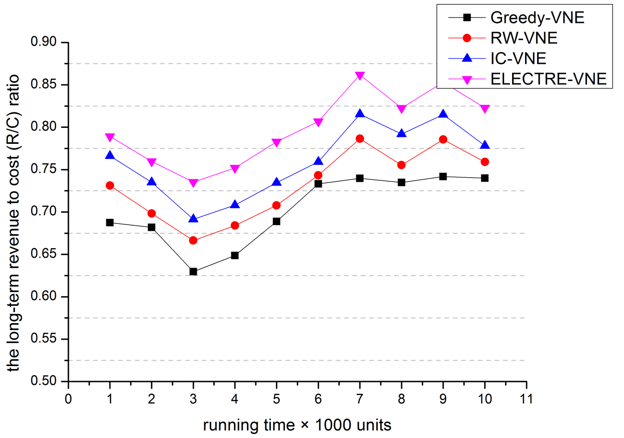


FIGURE 10. The long-term revenue to cost (R/C) ratio of substrate network.

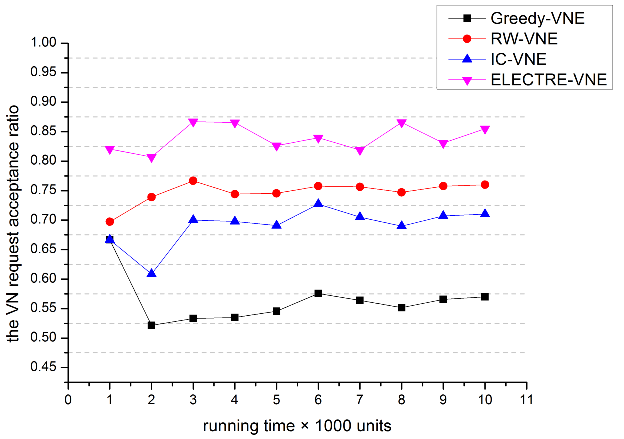


FIGURE 11. The VN request acceptance ratio.

ELECTRE-VNE is the highest one among these compared four algorithms. Based on the quantitative analysis from the concrete data, the VN request acceptance ratio of our method ELECTRE-VNE is 27.67% higher than GREEDY-VNE, 14.93% than RW-VNE, and 9.24% than IC-VNE. Due to the fact that our method ELECTRE-VNE incorporates the multiple metrics of node ranking into the node importance computation process, which would increase the VN request acceptance ratio from the long run. The main reason is that node degree, modified resource evaluation value and CPU utilization ratio can give a comprehensive node ranking value, the number of hops between the mapping substrate node and the set of already mapped substrate nodes can save bandwidth resources to accommodate more virtual network requests. Both of them can increase the acceptance ratio of virtual network requests.

2) SIMULATION EXPERIMENT 2

The second experiment aims to measure the performance of our method on the VN requests with different CPU capacity requirements. We carried out the second experiment, and let the required CPU capacity of virtual nodes uniformly distribute between 0 and C_n, therein C_n increases from 10 to 100 step 10. We evaluate our method compared with the other three methods in terms of the long-term average revenue, the long-term R/C ratio and the VN request acceptance ratio.

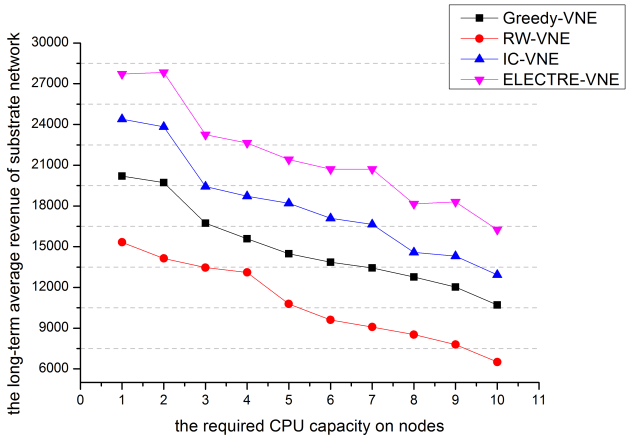


FIGURE 12. The long-term average revenue with increasing CPU capacity.

Fig. 12 shows the long-term average revenue of the compared four methods with increasing demand of CPU capacity on nodes. We can observe that our method ELECTRE-VNE is the best one among these compared four methods. According to the quantitative analysis from the concrete data, the long-term average revenue of our method ELECTRE-VNE is almost 50% higher than GREEDY-VNE, 31.07% higher than RW-VNE, and 16.98% higher than IC-VNE. From the overall trend, the long-term average revenue is decreasing with the increasing demanded CPU capacity on nodes due to the fact that substrate resources are diminishing with the increasing requirements of CPU capacity on nodes. From the overall trend of four lines, it can be seen that our proposed five definitions do contribute to the performance of our method. There is no obvious fluctuation with the increasing demanded CPU capacities on nodes which can justify our proposed method.

Fig. 13 presents the R/C ratio of the compared four methods with increasing demand of CPU capacity on nodes. We can observe that our method ELECTRE-VNE outperforms the other three methods, the main reason is that our method ELECTRE-VNE takes into consideration the hops between the mapping substrate node and the set of already mapped substrate nodes with an aim to reduce the unnecessary bandwidth resource consumption of substrate links. Through incorporating the multiple metrics of node ranking into the node importance computation process, we can choose the substrate node with the most embedding potential to perform the node mapping process, and thereby increase the R/C ratio of VNE algorithm. According to the quantitative analysis from the obtained data, our method ELECTRE-VNE is 17.93% higher than GREEDY-VNE, 13.97% higher than RW-VNE, and 11.56% higher than IC-VNE.

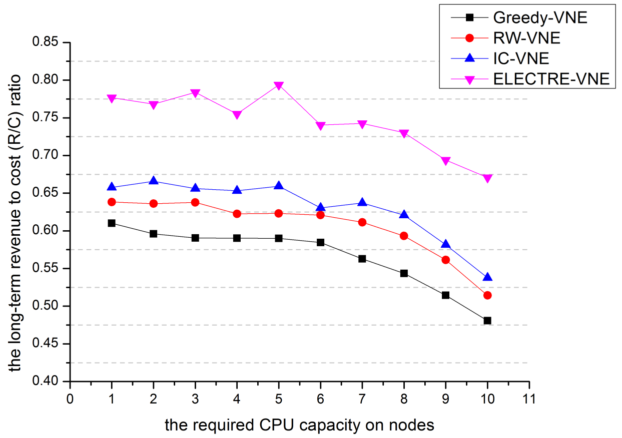


FIGURE 13. The long-term revenue to cost (R/C) ratio with increasing CPU capacity.

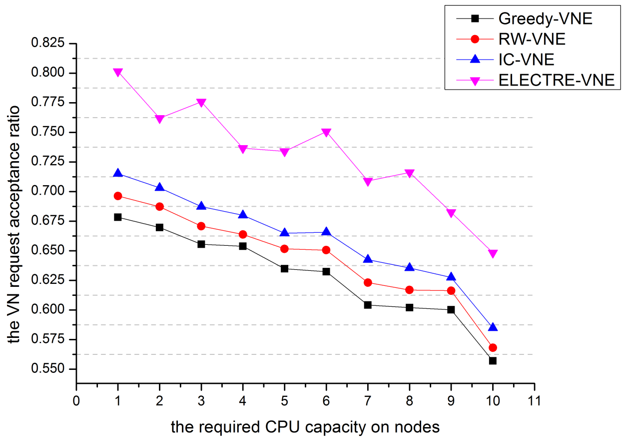


FIGURE 14. The VN request acceptance ratio with increasing CPU capacity.

Fig. 14 illustrates the VN request acceptance ratio of the compared four methods with increasing demand of CPU capacity on nodes. We can observe that our method ELECTRE-VNE is the highest one among these compared four methods. The main reason is that our method takes into account the multiple metrics of node ranking based on the topological analyses of substrate network, and employ comprehensive multiple attribute decision method ELECTRE to measure the node importance of substrate nodes in substrate network. The resource capacity can give a comprehensive node ranking value for each substrate node. The modified resource evaluation value can reduce the failure probability of subsequent link mapping process. The node degree can improve the node ranking of substrate nodes. The CPU utilization ratio can eliminate the bottleneck substrate node to increase the acceptance ratio of virtual network requests. According to the quantitative analysis from the concrete data, our method ELECTRE-VNE is 10.28% higher than GREEDY-VNE, 8.72% higher than RW-VNE, and 7.10% higher than IC-VNE.

In addition, through the vertical comparison Fig. 10 vs Fig. 13, Fig. 11 vs Fig. 14, we can see that our proposed method on increasing demand of CPU capacity performs better than on normal case. The reason is that our proposed five definitions are all associated with the node importance metric.

VII. CONCLUSIONS

Virtual network embedding is a promising technique to fend off the ossification of the current Internet architecture in network virtualization environments. In this paper, we introduce six situations to demonstrate the main drawbacks of the classical resource evaluation metric of node ranking, and give five definitions to evaluate the node importance with the aim of addressing these issues. We presented a novel VNE algorithm called ELECTRE-VNE, which uses the simplified ELECTRE method for five evaluation metrics to rank the importance of the substrate nodes, and utilizes the shortest path algorithm to perform the link mapping procedure. Two kinds of simulation experiments have shown that our method outperforms the other state-of-the-art methods in terms of the long-term average revenue, the long-term R/C ratio, and the VN request acceptance ratio.

Our method mainly uses different node ranking metrics that are jointly considered using simplified ELECTRE decision making method to perform the node mapping process. Therefore, the proposed method has always worked particularly in the case of distributed scientific computation service, where requiring more CPU capacity on nodes to process data but less bandwidth resource on links to transfer the computation results. In our future work, we will focus on the different link ranking metrics with the aim of increasing universality of our method, such as the case of the live network service requiring more bandwidth resources on links to transport packets but less CPU capacity on nodes to forward packets. In addition, we will concentrate on the energy-aware, security-aware, and QoS-aware VNE algorithms to satisfy diverse kinds of services for virtual networks.

ACKNOWLEDGMENT

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Y. Zhu and M. H. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

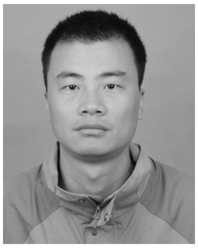
[2] Z. Zhang, X. Cheng, S. Su, Y. Wang, K. Shuang, and Y. Luo, "A unified enhanced particle swarm optimization-based virtual network embedding algorithm," *Int. J. Commun. Syst.*, vol. 26, no. 8, pp. 1054–1073, 2013.

[3] N. F. Butt, M. Chowdhury, and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding," in *Proc. Int. IFIP TC Conf. (NETWORKING)*, Chennai, India, May 2010, pp. 27–39.

[4] H. Cui, S. Tang, X. Huang, J. Chen, and Y. Liu, "A novel method of virtual network embedding based on topology convergence-degree," in *Proc. IEEE Int. Conf. Commun. Workshops*, Jun. 2013, pp. 246–250.

[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.*, Dec. 2012, pp. 2603–2608.

[6] J. Ding, T. Huang, J. Liu, and Y.-J. Liu, "Virtual network embedding based on real-time topological attributes," *Frontiers Inf. Technol. Electron. Eng.*, vol. 16, no. 2, pp. 109–118, 2015.
[7] 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.
[8] E. Amaldi, S. Coniglio, A. M. C. A. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electron. Notes Discrete Math.*, vol. 52, pp. 213–220, Jun. 2016.
[9] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proc. ACM Workshop Virtualized Infrastruct. Syst. Archit.*, 2009, pp. 81–88.
[10] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM*, vol. 20, Apr. 2009, pp. 783–791.
[11] Y. Wang, Q. Hu, and X. Cao, "A branch-and-price framework for optimal virtual network embedding," *Comput. Netw.*, vol. 94, pp. 318–326, Jan. 2016.
[12] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, "Optimal virtual network embedding: Node-link formulation," *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 4, pp. 356–368, Dec. 2013.
[13] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, "Optimal virtual network embedding: Energy aware formulation," *Comput. Netw.*, vol. 91, pp. 184–195, Nov. 2015.
[14] I. Houidi, W. Louati, W. B. Ameur, and D. Zeghlache, "Virtual network provisioning across multiple substrate networks," *Comput. Netw.*, vol. 55, no. 4, pp. 1011–1023, 2011.
[15] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.
[16] 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, Apr. 2008.
[17] X. Cheng *et al.*, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38–47, Apr. 2011.
[18] J. F. Botero and X. Hesselbach, "Greener networking in a network virtualization environment," *Comput. Netw.*, vol. 57, no. 9, pp. 2021–2039, 2013.
[19] 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.
[20] X. Hesselbach, J. R. Amazonas, S. Villanueva, and J. F. Botero, "Coordinated node and link mapping VNE using a new paths algebra strategy," *J. Netw. Comput. Appl.*, vol. 69, pp. 14–26, Jul. 2016.
[21] S. Gong, J. Chen, S. Zhao, and Q. Zhu, "Virtual network embedding with multi-attribute node ranking based on TOPSIS," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 2, pp. 522–541, 2016.
[22] M. Feng, J. Liao, J. Wang, S. Qing, and Q. Qi, "Topology-aware virtual network embedding based on multiple characteristics," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 1, pp. 2956–2962, 2014.
[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, p. 671, May 1983.
[24] P. Zhang, H. Yao, M. Li, and Y. Liu, "Virtual network embedding based on modified genetic algorithm," *Peer-to-Peer Netw. Appl.*, vol. 2017, no. 2, pp. 1–12, 2017.
[25] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Progetto Intell. Artificiale Robot., Dipartimento Elettron. Informazione, Polytech. Univ. Milan, Milan, Italy, 1992.
[26] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2011, pp. 1–6.
[27] P. Zhang, H. Yao, C. Fang, and Y. Liu, "Multi-objective enhanced particle swarm optimization in virtual network embedding," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–9, 2016.
[28] J. Inführ and G. Raidl, "A memetic algorithm for the virtual network mapping problem," *J. Heuristics*, vol. 22, no. 4, pp. 475–505, 2016.
[29] W. Wenzhao, W. Binqiang, W. Zhiming, and W. Bo, "Virtual network embedding algorithm based on analysis of network centrality," *Appl. Res. Comput.*, vol. 32, no. 2, pp. 565–568, 2015.
[30] M. Zeleny, P. Nijkamp, and A. Van Delft, "Multi-criteria analysis and regional decision-making," *J. Amer. Statist. Assoc.*, vol. 75, no. 372, p. 1035, 1977.
[31] B. Roy and D. Vanderpooten, "An overview on 'the European school of MCDA: Emergence, basic features and current works,'" *Eur. J. Oper. Res.*, vol. 99, no. 1, pp. 26–27, 1997.
[32] C.-L. Hwang and K. Yoon, *Multiple Objective Decision Making—Methods and Applications: A State-of-the-Art Survey*. Berlin, Germany: Springer, 1981.
[33] C. L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making—Methods and Applications*. Berlin, Germany: Springer, 1979.
[34] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE 15th Joint Conf. IEEE Comput. Soc. Netw. Next Gener. (INFOCOM)*, vol. 2, Mar. 1996, pp. 594–602.



PEIYING ZHANG is currently pursuing the Ph.D. degree in information and communication engineering from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He is currently an Associate Professor with the College of Computer and Communication Engineering, China University of Petroleum (East China). His research interests include natural language processing, semantic computing, future Internet architecture, network virtualization, and artificial intelligence for networking.

HAIPENG YAO is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His main research interests include future network architecture, artificial intelligence for networking, and the architecture and key technology of the new generation mobile communication system.

CHAO QIU received the B.S. degree in communication engineering from China Agricultural University, Beijing, China, in 2013. She is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. Her current research interests include deep learning, reinforcement learning, and future Internet architecture.

YUNJIE LIU received the B.S. degree in technical physics from Peking University, Beijing, China, in 1968. He is currently the Academician of the China Academy of Engineering, the Chief of the Science and Technology Committee of China Unicom, and the Dean of the School of Information and Communications, Beijing University of Posts and Telecommunications. His current research interests include next-generation network, and network architecture and management.