Research Article

An Efficient Resource and Topological Attribute-Based Configuration Algorithm of Virtual Networks

Xinyan Wang, Shunbin Li, and Longju Bai

1Intelligent Science and Information Engineering College, Xi’an Peihua University, Xi’an 710125, China
2Research Center for Industrial Internet, Zhejiang Lab, Hangzhou 311121, China
3College of Engineering, Beijing Forestry University, Beijing 100091, China

Correspondence should be addressed to Shunbin Li; lishunbin@zhejianglab.com

Received 9 August 2021; Revised 15 September 2021; Accepted 20 September 2021; Published 28 September 2021

Copyright © 2021 Xinyan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the future Internet operation and service model, users put forward network application requirements, and the service provider allocates the corresponding virtual network for users according to the demands. Virtual network resource allocation is an NP-hard problem; hence, the existing research work is mainly based on heuristic algorithms to solve the above problem. However, these traditional algorithms based solely on resource attributes are not sufficient to configure virtual networks. We design a virtual network configuration algorithm (ERTA-CA) based on the two-stage heuristic resource utilization that considers both local and global resource attributes and topological attributes of the network. The experimental results verify that our proposed algorithm can improve the acceptance rate and configuration revenue of the virtual network, which further indicates that ERTA-CA can achieve the virtual network configuration with efficient use of resources.

1. Introduction

The history of virtualization technology can be traced back to the 1960s. In 1967, IBM first introduced the host CP-40 that supports running virtual machines [1]. Virtualization is a technology that abstracts the underlying physical hardware to provide a simple virtualized interface to upper-level applications. Virtualization makes it possible to share computing resources and dynamically provide virtual resources to services and applications, realizing the full use of physical resources. Therefore, when using virtualization technology to deploy new applications, there is no need to purchase and install new server equipment. Virtualization technology has also become a key enabling technology [2–6] for the booming cloud computing technology [7–9] in the last decade. Two virtualization technologies that are currently widely used are host-level virtualization and container-level virtualization [10]. The host-level virtualization architecture usually consists of a physical host, a virtual machine monitor (named Hypervisor), and a virtual machine running on Hypervisor. Container-level virtualization is “lightweight” virtualization, which is very different from host-level virtualization. Host-level virtualization is resource isolation at the operating system level, while containers are essentially process-level resource isolation. Docker container technology [11] is currently the most popular container-level virtualization technology. Docker packages the application and the class library that the program depends on into a file and runs directly on the host operating system to generate a virtual container. The program runs in the container as if it were running on a real physical machine.

The core of network virtualization technology is abstraction and isolation. By abstracting and isolating the physical resources of the infrastructure, virtualization can realize the sharing of the physical network and ensure the independent operation of the services carried by the virtual network. Compared with the traditional network architecture, the biggest advantage of network virtualization is that service providers do not need to know the specific operation requirements of the physical facility but only needs to directly configure the virtual network according to the user’s needs. When current network operators deploy new
services, they need to install special middleware equipment in the network, such as intrusion detection systems (IDS), wide area network accelerator (WANA), and firewalls [12]. These types of middleware are usually dedicated and expensive equipment manufactured by different equipment providers and only provide limited control interfaces to network service providers. In addition, middleware equipment brings huge expenses to network service providers, such as the purchase cost of special equipment, the training cost of managing equipment staff, and the cost of maintaining and expanding this network equipment. Network function virtualization based on virtualization technology is proposed to solve the above shortcomings [13–15]. Network function virtualization uses virtualization technology to implement virtual network functions (VNFs) to replace traditional dedicated network functions, which can decouple network functions and dedicated hardware platforms design and deploy and manage network services in a whole new way. The planned virtual network needs to be mapped to the physical network. Various specific operations of the network are implemented to provide corresponding network services. Therefore, the configuration of the virtual network also determines the specific physical devices involved in a network application and the usage overhead of the device resources. A good virtual network configuration strategy should not only meet the QoS requirements of users but also carry as many virtual network users as possible and maximize the utilization of network resources to maximize operating benefits. In this paper, the virtual network configuration algorithm with the goal of maximizing the number of carrying virtual networks and maximizing the benefits of virtual network configuration under the given resource constraints is named the virtual network configuration algorithm for efficient use of resources. This paper focuses on the virtual network configuration algorithm aiming at efficient utilization of resources.

The virtual network resource allocation is an NP-hard problem [16], so most of the existing research work [17–20] is to propose heuristic algorithms to solve the problem. Because the heuristic algorithm can solve the suboptimal virtual network configuration scheme in a relatively fast time, a good balance is achieved between the complexity and optimization of the solution. One type of these heuristic algorithms only considers the resource attributes of the network and ignores the topological attributes [19] of the network to allocate physical resources to achieve efficient resource utilization. However, this type of traditional virtual network configuration algorithm based solely on resource attributes is not sufficient to configure the virtual network. Although there is a small amount of research work that considers the resource attributes and topological attributes of the network, these studies have not reasonably defined resources and topological attributes, so that the virtual network cannot be configured efficiently. To solve the problems left by the previous research, we design a virtual network configuration algorithm based on the two-stage heuristic resource utilization. When the virtual network request reaches the resource allocation system, in the first stage, the resource allocation system uses the resource attributes (CPU and bandwidth) and topology attributes of the network to comprehensively evaluate and sort the nodes. Then, the resource allocation system configures the virtual nodes based on the sorting results. In the second stage, the virtual link is configured using a link configuration algorithm based on the kth shortest path. The experimental results indicate that ERTA-CA can improve the performance in terms of virtual network acceptance rate and virtual network configuration revenue, which further verifies that the algorithm can realize the virtual network configuration with efficient use of resources.

2. Related Work

According to whether the virtual network configuration algorithm is suitable for large-scale networks, the virtual network configuration algorithm is divided into exact solutions and heuristic algorithms. The exact solution can effectively solve the virtual network configuration of small-scale problem instances. In the existing literature, the exact solution to solve the virtual network configuration problem is usually through the optimization theory to perform integer linear programming or mixed-integer linear programming modeling and then use Gurobi, Cplex, GLPK, and other tools to derive the optimal solution. Literature [21] proposed an integer linear programming model to solve the problem of online virtual network configuration to achieve minimal resource consumption and load balancing. The model considers the configuration of nodes and links to be completed in one stage at the same time. The authors in [21] compared the performance of the proposed exact solution with that of the heuristic algorithms and found that the exact solution can achieve better resource utilization and virtual network acceptance ratio. Houdi et al. [17] presented and introduced an accurate algorithm for virtual network mapping based on integer linear programming. The authors considered mapping the virtual network request to multiple physical networks and used the branch and bound method to solve the mapping problem. The solution results show that the exact solution is very effective for the mapping of small-scale virtual networks. The authors in [22] established a mixed-integer linear programming model for the survivability virtual network configuration problem of multinode failures. They use the GLPK tool to solve the model and the experimental results show that the exact solution can improve the survivability of the virtual network and increase the long-term configuration benefits.

Because the virtual network configuration problem is proved to be NP-hard, the heuristic algorithms for solving virtual network configuration have received more attention than exact solutions. The reason is that it can solve the suboptimal virtual network configuration scheme in a relatively short time period, and it is a good balance between the complexity and the optimal solution. It is more practical for large-scale virtual network configuration problems. Literature [23] proposed an efficient online heuristic virtual network embedding (VNE) algorithm (Presto) based on the blocking Island model to maximize the acceptance ratio of the virtual network, which enhances the computational
efficiency and reduces the search domain. The authors in [24] designed an effective heuristic method in view of the fact that most of the existing heuristic methods cannot cope with physical network node or link failure. Considering the physical network single link failure survivability virtual network configuration problem, an efficient heuristic method is designed. In addition, many researchers have adopted metaheuristic algorithms that simulate random phenomena in nature to solve virtual network configuration problems. Some research used the ant colony optimization algorithm, such as literature [25] which used ant colony optimization algorithm to solve virtual network configuration. Literature [26] designed an energy-saving virtual network configuration based on ant colony optimization. Some works have adopted genetic algorithms. Literature [27] proposed two virtual network configuration methods CB-GA and RW-GA based on genetic algorithms. The simulation results show that these two algorithms are superior to virtual network configuration based on particle swarm optimization in terms of configuration benefits and virtual network acceptance rate. The authors in [28] used the genetic algorithm to solve the multidomain virtual network configuration problem, and the simulation results show that the proposed algorithm is better than the latest reference methods.

Based on the above literature review, the study of this paper will use a single-domain physical network. Although the research on virtual network scenarios is attracting more researchers’ attention, most of the research is still based on single-domain physical networks. This is because the subproblem of virtual network configuration in the multidomain physical network is intradomain configuration, which is equivalent to single-domain virtual network configuration. Hence, the single-domain virtual network configuration algorithm can be better used for intradomain configuration in multidomain scenarios. Similar to most research, in order to be consistent with the actual network operation process, the research of this manuscript considers the online virtual network configuration problem. Furthermore, given that most of the current network scales reach medium and large scales, the design of heuristic virtual network configuration algorithms has become the mainstream method to solve virtual network configuration problems.

This paper designs a virtual network configuration algorithm based on the two-stage heuristic resource efficient utilization that comprehensively considers local and global resources and topological attributes of the network. When the virtual network request is allocated to the resource allocation system, in the first stage, the resource allocation system uses network resource attributes and topology attributes to comprehensively evaluate and sort nodes and then configure virtual nodes based on the sorting results. In the second stage, the virtual link is configured using a link configuration algorithm based on the k shortest path. Compared to previous algorithms, our method considers local and global resources and local and global topology attributes, so that the virtual network can be configured more efficiently.

3. Problem Description and Model

3.1. Problem Description. Virtual network resource configuration is based on virtual network request to consider certain goals, such as maximizing virtual network configuration. To set revenue, map the virtual node to the physical network node that meets the demands including resource and deployment location, and map the virtual link to the physical link or path that satisfies the bandwidth request. Only when all nodes and links of the virtual network whose demands are met, the virtual network request is accepted; otherwise, it is rejected. It is reasonably assumed about the virtual network resource configuration process. (1) During the life cycle of a virtual network request, the network topology of the virtual network remains unchanged; that is, the virtual network reconfiguration problem is not considered. (2) Virtual nodes requested from the same virtual network can only be mapped to different physical nodes; that is, node mapping is a one-to-one mapping. (3) Virtual links cannot be divided; that is, virtual links can only be carried by one physical network link or path.

3.2. Virtual Network Model. Since the online virtual network configuration is closer to the actual environment, the dynamic interaction process between users and service providers is modeled. Therefore, we consider online virtual network configuration, that is, virtual network requests dynamically reach the resource configuration management system. For virtual network requests that continue to reach the resource configuration management system over time, the virtual network resource configuration process contains two subprocesses, virtual node mapping and virtual link mapping. Once the resource configuration of the virtual network request is successful, then within the lifetime of the virtual network, the allocated resources will be exclusive to the virtual network. When the lifetime of the virtual network ends, the allocated resources are released for use by virtual network requests that arrive next. Here, the virtual node mapping function, virtual link mapping function, virtual network configuration function, and resource update in the virtual network configuration process are defined.

Definition 1 (virtual node mapping function). Given a physical network \( G^p \) and virtual network \( G^v \) and the corresponding node sets \( N^p \) and \( N^v \), the virtual node mapping function \( M \) is a one-to-one mapping function, that is, when \( M(N): N^v \rightarrow N^l, N^l \subseteq N^i \) is a one-to-one mapping function, that is, when \( n^v_i = n^l_j, M(n^v_i) = M(n^l_j) \). The virtual node set \( N^v \) is mapped to the subset \( N^l \) of the physical node set.

Definition 2 (virtual link mapping function). Given a physical network \( G^p \) and virtual network \( G^v \) and their corresponding link sets \( E^p \) and \( E^v \), the virtual link mapping function is \( M(E): E^v \rightarrow P^l, P^l \subseteq P^i \). The virtual link set \( E^v \) is mapped to the subset \( P^l \) of the physical node set. It is worth noting that the same physical link may accompany multiple virtual links in the same virtual network, which is different.
from that the same physical node can only carry at most one virtual node in the same virtual network.

**Definition 3** (virtual network configuration function). Given a physical network \( G^p=(N^p, E^p) \) and virtual network \( G^v=(N^v, E^v) \), virtual network configuration function is \( M(V) : (N^v, E^v) \rightarrow (N^p, P^p) \).

**Definition 4** (resource update). \( c_v(n^i, t) \) represents the total computing resources of the physical node \( n^i \) allocated to all virtual nodes from different virtual networks at the specific time \( t \); if \( c_v(n^i, t) = \sum_{n^i} c_v(n^v) \), then \( c_v(n^i, t) = c_v(n^i) - c_v(n^i, t) \). Similarly, use \( b_v(n^i, t) \) to denote the bandwidth resources allocated to all virtual links by the physical link \( e^i \) at a specific time \( t \). The physical path mapped by these virtual links passes through \( e^i \); if \( b_v(e^i, t) = \sum_{e^i} c_v(e^i) \), then \( b_v(e^i, t) = b_v(e^i, t) - b_v(e^i, t) \).

### 3.3. Problem Formulation.

Define indicator variable \( x^i_k \) and \( y^ij_{kl} \) as the mapping of nodes and links, respectively. \( x^i_k \) means whether the virtual node \( n^v_k \) is mapped to the physical node \( n^i \); \( y^ij_{kl} \) means whether the physical link \( e^i_{ij} \) carries the virtual link \( e^v_{ij} \):

\[
x^i_k = \begin{cases} 1, & \text{if physical node } n^i \text{ carries virtual node } n^v_k, \\ 0, & \text{otherwise}, \end{cases}
\]

\[
y^ij_{kl} = \begin{cases} 1, & \text{if physical link } e^i_{ij} \text{ carries virtual link } e^v_{ij}, \\ 0, & \text{otherwise}. \end{cases}
\]

In order to ensure the correct configuration of the virtual network, the following constraints must be met.

**Virtual node mapping:** ensure that each virtual node is mapped to a physical node:

\[
\sum_{n^i} x^i_k = 1, \quad \forall n^v_k \in N^v. \quad (2)
\]

**On-to-one node mapping:** an arbitrary physical node carries one virtual node requested from the same virtual network:

\[
\sum_{n^i} x^i_k \leq 1, \quad \forall n^i \in N^i. \quad (3)
\]

**Computing resources:** the virtual node’s request for CPU computing resources should be less than the available CPU resources of the physical node:

\[
\sum_{n^i} x^i_k \cdot c_v(n^v_k) \leq c_v(n^i), \quad \forall n^i \in N^i. \quad (4)
\]

**Deployment location:** the distance between the virtual node’s actual deployment location and its expected deployment location should be less than the maximum allowable deployment deviation:

\[
x^i_k \cdot \text{dis}(\text{loc}(n^v_k), \text{loc}(n^i)) \leq r(n^v_k). \quad (5)
\]

**Herein,**

\[
\text{dis}(\text{loc}(n^v_k), \text{loc}(n^i)) = \sqrt{(x(n^v_k) - x(n^i))^2 + (y(n^v_k) - y(n^i))^2}. \quad (6)
\]

**Current conservation constraint:** the flow conservation constraint ensures that each virtual link is mapped to a physical path and ensures that the network traffic passing through other intermediate physical nodes except for the physical node carrying the virtual node is zero:

\[
\sum_{n^i} \left( y^ij_{ij} - y^ij_{ik} \right) = x^i_k - x^i_k, \quad \forall e^i_{ij} \in E^v, n^i \in N^i. \quad (7)
\]

**Bandwidth resources:** the sum of bandwidth requests of all virtual links carried by the physical link cannot exceed the available bandwidth resources of the physical link:

\[
\sum_{e^i_{ij} \in E^v} y^ij_{ij} \cdot b(e^i_{ij}) \leq b_v(e^i_{ij}), \quad \forall e^i_{ij} \in E^i. \quad (8)
\]

When configuring a virtual network, the virtual network can be configured successfully only when the above constraints are met. For a successfully configured virtual network, the service provider needs to pay the physical network provider based on the resource request amount of the virtual network as the operating income of the physical network provider. Assuming that the unit price of resources is 1, the benefits of configuring each virtual network can be defined as

\[
\text{REV}(G^v) = \sum_{n^i \in N^i} c_v(n^i) + \sum_{e^i \in E^i} b(e^i). \quad (9)
\]

From the definition of configuration revenue, it can be seen that since the amount of resources requested by a virtual network is fixed, the revenue obtained from successful configuration is fixed. Then, in the online configuration process, the physical network provider needs to accept more virtual network requests in order to obtain a greater long-term operating income. Therefore, the virtual network configuration algorithm needs to be able to efficiently use limited physical resources to accept more virtual network requests.

### 4. The Proposed Solution

The problem of virtual network resource allocation is NP-hard. Therefore, similar to most existing studies, this research designs a heuristic algorithm to solve the above-modeled problems. The advantage of the heuristic algorithm is that it can be solved with lower complexity and obtain satisfactory results. In view of the fact that existing research fails to fully consider the impact of local and global resources and topological attributes on the evaluation of nodes, define the local and global resources and topological attributes of nodes to evaluate node strategies and design a heuristic algorithm based on node evaluation and ranking.

#### 4.1. Complex Network Topology Measurement Metrics

This paper uses node degree centrality to characterize the local topological attribute measurement of the network. For
the global topological attributes, because the eigenvector centrality is used for directed graphs, this study does not consider it. Therefore, only the node degree centrality and proximity centrality are introduced here.

4.1.1. Degree Centrality. Degree centrality is defined as the number of connected edges of a node, which characterizes the local popularity of the node or the influence of its neighbors. The more important a node is, the more direct connected edges are. For directed graphs, out-degree centrality and in-degree centrality can be defined similarly. Degree centrality is only a partial measurement of nodes. Since degree centrality cannot reflect the importance of nodes in the entire network, it is not sufficient to fully describe the topological properties of nodes.

4.1.2. Closeness Centrality. Closeness centrality is defined as the reciprocal of the average shortest distance from a node to other nodes. The greater the closeness centrality of the node, the easier it is to reach other nodes to obtain information or have a more direct impact on other nodes.

4.2. Evaluate the Network Resources and Topological Properties of Nodes. The virtual network configuration process is divided into two phases: virtual node configuration and virtual link configuration. In the virtual node configuration phase, it is necessary to reasonably determine the configuration order of the virtual nodes and reasonably select physical nodes to carry the virtual nodes to meet the requirements of the virtual node’s CPU requirements and configuration position. This research believes that the configuration of virtual nodes needs to comprehensively consider the resources and topological attributes in the network to evaluate the importance of nodes, and then configure virtual nodes in order of importance. Hence, this research jointly considers resource and topological attributes to measure the significance of nodes.

4.2.1. Local Resource Attribute Measurement. We can obtain the local resource of a node by the product of the node’s CPU and the bandwidth of all its adjacent links. It is defined as follows:

\[
LR(n_i) = c(n_i) \sum_{e \in E(n_i)} b(e),
\]

where \(E(n_i)\) represents all adjacent links of \(n_i\).

4.2.2. Global Resource Attribute Measurement. Considering the minimum bandwidth of the shortest path between the current node and the rest nodes and the minimum CPU resources of the current node on this path, the sum of them is used as the global resource measurement, and the normalization is defined as follows:

\[
GR(n_i) = \frac{\sum_{i \neq j} [b(p(n_i, n_j)) + c(p(n_i, n_j))] + c(p(n_i, n_j))}{|N| - 1},
\]

in which \(b(p(n_i, n_j))\) means the minimum link bandwidth on the shortest path between nodes \(n_i\) and \(n_j\) and \(c(p(n_i, n_j))\) represents the minimum CPU resources of nodes other than \(n_i\) on the shortest path between nodes \(n_i\) and \(n_j\).

4.2.3. Local Topological Attribute Measurement. In the undirected network graph, the local topological attributes of nodes are expressed by degree centrality, which is defined as the ratio of the number of connected edges of a node to the maximum possible number of connected edges of the node, that is, normalized degree centrality:

\[
DC(n_i) = \frac{\sum_{j} a_{ij}}{|N| - 1},
\]

Among them, if the node \(n_i\) and the node \(n_j\) have an edge, \(a_{ij}\) takes 1; otherwise, it is 0. Degree centrality measures the local topological characteristics of nodes in the network. The greater the degree of a node is, the more adjacent nodes it has and the greater its influence in the network is.

4.2.4. Global Topological Attribute Measurement. The global topological properties of a node are represented by closeness centrality. We can compute the sum of the shortest paths between the node and the rest nodes, taking the reciprocal as the closeness centrality. The normalized closeness centrality is expressed as follows:

\[
CC(n_i) = \frac{|N| - 1}{\sum_{j} d(n_i, n_j)},
\]

where \(d(n_i, n_j)\) represents the shortest distance between \(n_i\) and \(n_j\). Closeness centrality can be used to describe the topological properties of nodes.

4.3. The Proposed Heuristic Virtual Network Resource Allocation Algorithm ERTA-CA. This section describes the two-phase heuristic virtual network resource allocation algorithm proposed in this paper. In the first phase, the heuristic virtual node is configured according to resource and topological attributes. In the second phase, the virtual link is configured based on the \(k\) shortest path.

4.3.1. Virtual Node Configuration. Similar to earlier literature, a node evaluation strategy that integrates network topology attributes and resource attributes is proposed in the form of product and sum. Specifically, the node is evaluated by multiplying the local resource measurement result and the local topology measurement result plus the global resource measurement results and the global topology measurement result, as the following formula:

\[
S(n_i) = LR(n_i) \cdot DC(n_i) + GR(n_i) \cdot CC(n_i).
\]

In this strategy, local and global resources and local and global topological attributes are considered at the same time to systematically evaluate physical nodes and virtual nodes:
When a virtual network request reaches the virtual network resource configuration system, each virtual node in the network request is scored according to formula (15), and the virtual nodes with higher scores are preferentially configured.

In the virtual network configuration process, if the physical network nodes are sorted by formula (14), the physical nodes selected eventually may be far away, which will cause the second phase of the link configuration to produce a very long physical path to carry the virtual link, causing insufficient utilization of physical network resources. To cope with such a problem, the following collaborative configuration method is adopted. When configuring the current virtual node, a set of candidate physical nodes that bear the virtual node is obtained. The smaller the sum of the shortest path hops between these candidate physical nodes and the physical nodes of all neighbor nodes that have carried the current virtual node, the more preferentially it is selected to carry the current virtual node. This collaborative configuration method is conducive to obtaining a shorter physical path to carry the virtual link during the second phase of virtual link configuration, thereby increasing the utilization of bandwidth resources. Therefore, the collaborative configuration coefficient is defined as follows:

\[
H(n_i^v) = \sum_{n_i^p \in M(\text{Adj}(n_i^v))} h(n_i^p, n_i^v),
\]

(16)

Herein, \(n_i^v\) represents a candidate physical node that meets the CPU requirements and location requirements of the virtual node \(n_i^v\), \(M(\text{Adj}(n_i^v))\) represents the set of physical nodes carrying all neighbor nodes of the virtual node \(n_i^v\). \(h(n_i^p, n_i^v)\) means the number of hops of the shortest path between physical nodes \(n_i^p\) and \(n_i^v\). The collaborative configuration coefficient is introduced into the scoring evaluation of physical nodes, and the scoring method of physical nodes is obtained as in formula (17). The value of \(E\) is \(10^{-7}\) to avoid the divisor being zero:

\[
S(n_i^v) = \frac{LR(n_i^v) \cdot \text{DC}(n_i^v) + \text{GR}(n_i^v) \cdot \text{CC}(n_i^v)}{H(n_i^v) + \varepsilon}.
\]

(17)

The detailed description of the virtual node configuration algorithm is shown in Figure 1.

4.3.2. Virtual Link Configuration. In the virtual link configuration phase, because of the virtual link with the larger bandwidth resource request, it is more difficult to configure. Therefore, it is needed to sort the virtual links according to the size of the bandwidth demand from large to small. We run the \(k\) shortest path algorithm on the sorted virtual links to obtain \(k\) candidate physical paths that satisfy the bandwidth resource request of each virtual link. In our scheme, in order to further improve the acceptance rate and resource utilization of virtual networks, a new path selection strategy is proposed based on the original \(k\) shortest path algorithm.

The product of the maximum link bandwidth utilization and the number of path hops in the path is defined as the path factor, which is expressed as

\[
P_j = \left(1 - \frac{b_j(e^j)}{b_j(e^j)^\text{max}} \right) \cdot |L(p^j)|.
\]

(18)

The proposed strategy selects the path with the smallest path factor from the candidate paths obtained by the \(k\) shortest path algorithm to carry the virtual link. The virtual link configuration is as in Figure 2.

4.3.3. Virtual Network Configuration. When the \(i\)th virtual network request arrives at the virtual network configuration system, the configured virtual networks whose life cycle ends at time \(t\) is judged. And then the system releases the physical resources the configured virtual networks occupy and updates the physical resources. Then, it configures virtual nodes and virtual links according to the above heuristic virtual node configuration and virtual link configuration algorithm. If and only if both the virtual node configuration and the virtual link configuration are successful, then the virtual network is accepted by the virtual network configuration system and resources are allocated. The virtual network configuration algorithm is described in Figure 3.

4.3.4. Time Complexity Analysis. Here is a brief analysis of the time complexity of the algorithm ERTA-CA. ERTA-CA is composed of virtual node configuration Algorithm 1 and virtual link configuration Algorithm 2. Its time complexity is the sum of the time complexity of Algorithms 1 and 2. For the virtual network node configuration algorithm, its complexity is mainly focused on traversing the resources and topological properties of physical network nodes and virtual network nodes, and the calculation of collaborative configuration coefficients. The total time complexity of Algorithm 1 is \(O(|N^V|^3) + O(|N^V|^3)\). For the virtual link configuration algorithm, its time complexity is \(O(|E^V| \log |E^V|) + O(k|E^V||N^V|^3(|E^V| + |N^V| \log |N^V|))\). In summary, the complexity of algorithm ERTA-CA is \(O(|N^V|^3) + O(|N^V|^3) + O(|E^V| \log |E^V|) + O(k|E^V| |N^V|^3(|E^V| + |N^V| \log |N^V|))\).

5. Experimental Results and Performance Analysis

5.1. Introduction to the Simulation Process. To evaluate the proposed algorithm, we developed a virtual network configuration simulator VNP-sim (Virtual Network Provisioning Simulator). VNP-sim uses JAVAtodevelop inconformance with object-oriented design principles and adopts a modular design method to make the simulator have a flexible architecture. The three core modules of the simulator include a virtual network configuration problem instance generation module, a virtual network configuration algorithm module, and a performance evaluation module of a virtual network configuration algorithm. These modules correspond to the three subprocesses of the simulation process: the virtual network configuration
Algorithm 1 virtual node configuration
Input: physical network \(G_I\), virtual network request \(G^V\)
Output: virtual node configuration strategy

1: for each physical node \(n^I \in N^I\) do
2: compute \(LR(n^I), DC(n^I) + GR(n^I), CC(n^I)\) and store it
3: end for
4: for each virtual node \(n^V \in N^V\) do
5: scoring \(S(n^V)\) according to formula (16)
6: end for
7: all virtual nodes are sorted according to \(S(n^V)\), and the sorting results stored in \(\text{virtualNdRkList}\)
8: for each virtual node \(n^V \in \text{virtualNdRkList}\) do
9: obtain candidate\((n^V)\) meet the virtual node \(n^V\) CPU resource request constraints and configuration location constraints
10: if candidate\((n^V)\) is not null then
11: obtain \(M(Adj(n^V))\)
12: for each \(n^I \in \text{candidate}(n^V)\) do
13: compute \(H(n^I)\) according to (17)
14: combine the result of step 2 and calculate \(S(n^I)\)
15: end for
16: configure \(n^V\) and save the configuration result to \(\text{nodeMapping}\)
17: else
18: return \(\text{nodeMappingFailed}\)
19: end if
20: end for
21: return \(\text{nodeMapping}\)

Figure 1: The pseudocode of virtual node configuration algorithm.

Algorithm 2 virtual link configuration
Input: physical network \(G_I\), virtual network request \(G^V\), and node configuration result \(\text{nodeMapping}\)
Output: virtual link configuration strategy

1: sort the virtual link \(E^V\) according to the virtual kink bandwidth request from large to small
2: the sorting results are stored in \(\text{virtualLinkRankList}\)
3: for each virtual link \(e^V \in \text{virtualLinkRankList}\) do
4: execute the k shortest path algorithm to obtain k candidate physical paths \(\text{subPathList}\)
5: if \(\text{subPathList}\) is not null then
6: for each physical link \(\text{subPath} \in \text{subPathList}\) do
7: calculate the path factor according to (19)
8: end for
9: configure the virtual link \(e^V\) on the candidate physical path with the smallest path factor
10: else
11: return \(\text{linkMappingFailed}\)
12: end if
13: return \(\text{linkMappingList}\)

Figure 2: The pseudocode of virtual link configuration algorithm.

Algorithm 3 virtual network configuration
Input: physical network \(G_I\) and i-th virtual network request \(VR^i = (G^V_i, t^\alpha_i, t^\tau_i)\)
Output: virtual network configuration results

1: check the virtual network request at \(t^\alpha_i\) and release the physical resource
2: use algorithm 1 to configure the virtual node of \(G^V_i\)
3: if virtual node configuration failed then
4: return virtual node configuration failed
5: else
6: use algorithm 2 to configure the virtual link of \(G^V_i\)
7: if virtual link configuration failed then
8: return virtual link configuration failed
9: else
10: allocate physical resources to \(G^V_i\) and update physical resources
11: return virtual link configuration succeeded
12: end if
13: end if

Figure 3: The pseudocode of ERTA-CA.
problem instance generation process, the operation process of virtual network configuration, and the experimental result evaluation process.

5.2. Performance Metrics. This section uses the virtual network acceptance rate, the long-term average revenue of virtual network resource allocation, and the long-term average revenue overhead ratio as performance metrics to measure the performance of the virtual network resource allocation algorithm.

The virtual network acceptance rate is defined as the ratio of the number of successfully configured virtual networks to the total number of virtual network requests that arrive in a period of time. It can be written as

\[
AR = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} S_m(t)}{\sum_{t=0}^{T} S(t)},
\]

where \( S(t) \) represents the total number of virtual network requests at time \( t \) and \( S_m(t) \) represents the total number of virtual networks successfully configured at time \( t \).

Long-term average revenue of virtual network resource allocation: configuration revenue of the virtual network request \( G^V \) at time \( t \) is defined as

\[
REV(G^V, t) = \sum_{n^V \in NV} c(n^V) + \sum_{e^V \in EV} b(e^V).
\]

Hence, the long-term average revenue is expressed as

\[
\mu = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} \sum_{G^V \in Sm(t)} REV(G^V, t)}{T}
\]

Virtual network resources allocation long-term average revenue-to-expense ratio: the configuration overhead of virtual network request \( G^V \) at time \( t \) is defined as

\[
COST(G^V, t) = \sum_{n^V \in NV} c(n^V) + \sum_{e^V \in EV} |[L(p^L(e^V))]|b(e^V),
\]

where \( p^L(e^V) \) represents the physical path carrying the virtual link \( e^V \) and \( L(p^L(e^V)) \) represents the set of all physical links contained in the physical path \( p^L(e^V) \). In this paper, when calculating configuration benefits and expenses, it is assumed that the unit price of computing power and bandwidth benefits and expenses are all 1. Therefore, the long-term average revenue-to-expense ratio of virtual network resource allocation is

\[
\eta = \frac{REV}{COST} = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} \sum_{G^V \in Sm(t)} REV(G^V, t)}{\sum_{t=0}^{T} \sum_{G^V \in Sm(t)} COST(G^V, t)}.
\]

5.3. Experiment Setup. All experiments in this section run on Intel Core i7-6820HQ CPU based on a laptop with 24G RAM. The operating system version is Windows 10. In the experiments, the Brite tool integrated into the simulator is used to generate the physical network topology and the virtual network request topology. The parameter settings are listed in Table 1.

5.4. Experimental Results and Analysis. The ERTA-CA algorithm designed in this paper configures the virtual network by evaluating nodes from the local and global resource attributes and their topological attributes. To evaluate the performance of the proposed ERTA-CA algorithm, we choose VNE-DCC, NRM-VNE, and CC as the reference algorithms. These reference algorithms are all existing research on configuring virtual networks considering network resources and topological attributes. To measure the effectiveness of the algorithms in this paper, we have compared with these algorithms. First, the main experiment evaluates the performance of the algorithms in the case where the virtual network arrival rate reaches 4 virtual networks per 100 time units on the physical network based on the Waxman and BA models. Then, we change the virtual network link connection probability on the physical network based on the Waxman model to verify the scalability of the proposed algorithm. In each case, the experiments are run 100 times, and the results are averaged [29–33].

5.4.1. The Virtual Network Arrival Rate Is 4 Virtual Network Requests Every 100 Time Units. In this scenario, the experiment is carried out on two physical networks based on the Waxman model and BA model. There are 100 physical nodes in the physical network. The virtual network acceptance rate, the long-term average revenue, and the long-term average revenue-to-expense ratio of virtual network resource allocation are shown in Figures 4–6.

Figure 4 shows that the proposed ERTA-CA algorithm peaks the highest value in terms of acceptance ratio of the virtual network on both physical networks during the entire simulation time period. At the beginning of the simulation, the acceptance ratio of various algorithms maintains a high value. The reason is that the physical network at this time owns an efficient resource to configure the virtual network. As time goes by, the number of active configured virtual networks in the resource allocation system increases, resulting in a decrease in available physical resources and a gradual decline in the acceptance ratio of virtual networks. What needs to be explained here is that Figure 4 shows the data structure after 1000 time units. Overall, the change trend of acceptance ratio under both Waxman and BA is consistent. After 10,000 time units, the acceptance ratio of the virtual network trends to convergence. This is because the arrival and departure of the virtual network are relatively balanced and the available physical resources are relatively stable. When approaching 40,000 time units, the acceptance ratio of the ERTA-CA algorithm under the Waxman is 83.46%, which is 3.32%, 8.19%, and 37.95% higher than VNE-DCC, NRM-VNE, and CC, respectively. In the BA network, the proposed ERTA-CA algorithm can also show similar performance advantages. ERTA-CA can systematically evaluate nodes from the perspectives of local and global resources and topological attributes to sort and configure nodes, which makes the configuration of virtual networks better, thereby increasing the acceptance ratio of virtual networks.
Figure 5 shows that the ERTA-CA algorithm can obtain the highest long-term average revenue of virtual network resources allocation. At the beginning of the simulation, the rapid decline in long-term average revenue is due to the continuous occupation of physical resources as virtual network requests arrive, and subsequent virtual network requests are more likely to be rejected. When the simulation time reaches about 10,000 time units, the long-term average revenue tends to stabilize due to the relatively stable arrival and departure of virtual network requests. In a steady state, the long-term average revenue of the ERTA-CA under the Waxman network is 9.65%, 11.7%, and 48.7% higher than that of VNE-DCC, NRM-VNE, and CC algorithm, respectively. It is 3.85%, 10.04%, and 30.03% higher than VNE-DCC, NRM-VNE, and CC under the BA network.

The same as the acceptance ratio and long-term average revenue, the long-term average revenue/cost ratio is relatively stable when the simulation time reaches 10,000 time units. Figure 6 shows the results of the long-term average revenue/cost ratio in the steady state. The results verify that ERTA-CA has better performance than other algorithms.

### 5.4.2. Different Virtual Link Connection Probability Scenarios

In the experimental scenario, change the virtual link connection probability to 0.2, 0.5, and 0.8 to study the impact on the performance. Figures 7 and 8 show the virtual network acceptance ratio and resource allocation revenue performance in the experimental scenario.

Figure 7 indicates that, as the virtual network link connection probability increases, the acceptance ratio decreases. The reason is that the virtual network requests with lower connection probabilities are more likely to be rejected.
**Figure 5:** Experimental results: (a) long-term average revenue under Waxman and (b) long-term average revenue under BA.

**Figure 6:** Experimental results: (a) revenue/cost ratio of resource allocation under Waxman and (b) revenue/cost ratio of resource allocation under BA.
more virtual links require more bandwidth resources, making it more difficult for the physical network to meet the bandwidth request, causing more virtual networks to be rejected. ERTA-CA can configure virtual networks more effectively so that it always holds the best performance under any virtual link connection probability. Figure 8(a) indicates that, as the virtual network link connection probability increases, the long-term average revenue of all other algorithms increases except for the CC algorithm. For algorithms other than CC, although the virtual network acceptance ratio is smaller in the case of a larger virtual network link connection probability, more virtual links are configured, which brings greater configuration benefits. For the CC algorithm, when the virtual link is connected with a probability of 0.2,
compared with 0.5 and 0.8, it can obtain a better acceptance ratio and thus obtain more configuration benefits. Among all algorithms, ERTA-CA always has the highest long-term configuration benefit under any virtual link connection probability. Figure 8(b) shows that, as the virtual network link connection probability increases, the long-term revenue/cost ratio of virtual network configuration decreases. However, $m$ in the case of high probability connection can increase the revenue but reduce the revenue/cost ratio. The reason is that when virtual network links are connected with high probability, due to the limited bandwidth resources, virtual links are easier to configure to longer physical links, which causes greater configuration overhead, and the increase in revenue is less than the increase in overhead.

6. Conclusions

The virtual network configuration problem is an NP-hard problem, and most of the existing research is solved by designing a heuristic configuration strategy in a short time to obtain an approximate optimal solution. Inspired by the existing research on designing node evaluation rules based on network resources or topological attributes, the research in this paper aims at virtual network configuration for efficient use of resources and proposes a two-phase heuristic algorithm based on local and global resources and topological attributes. ERTA-CA efficiently utilizes physical network resources to increase the acceptance ratio of virtual networks and the benefits of virtual network resource allocation. The experimental results further verify the effectiveness of the proposed ERTA-CA.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the present study.

Acknowledgments

This work was supported by National Key R&D Project, under Grant no. 2018YFB2100400.

References

[1] R. J. Creasy, “The origin of the VM/370 time-sharing system,” IBM Journal of Research and Development, vol. 25, no. 5, pp. 483–490, 1981.
[2] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, “Idle slots skipped mechanism based tag identification algorithm with enhanced collision detection,” KSII Transactions on Internet and Information Systems, vol. 14, no. 5, pp. 2294–2309, 2020.
[3] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, “Redundant rule detection for software-defined networking,” KSII Transactions on Internet and Information Systems, vol. 14, no. 6, pp. 2735–2751, 2020.
[4] J. Su, Z. Sheng, A. X. Liu, Z. Fu, and Y. Chen, “A time and energy saving-based frame adjustment strategy (TES-FAS) tag identification algorithm for UHF RFID systems,” IEEE Transactions on Wireless Communications, vol. 19, no. 5, pp. 2974–2986, 2020.
[5] J. Su, Y. Chen, Z. Sheng, Z. Huang, and A. X. Liu, “From M-ary Query to Bit Query: a new strategy for efficient large-scale RFID identification,” IEEE Transactions on Communications, vol. 68, no. 4, pp. 2381–2393, 2020.
[6] J. Su, Z. Sheng, A. X. Liu, Y. Han, and Y. Chen, “A group-based binary splitting algorithm for UHF RFID anti-collision systems,” IEEE Transactions on Communications, vol. 68, no. 2, pp. 998–1012, 2020.
[7] P. Mell and T. Grance, The NIST Definition of Cloud Computing, National Institute of Standards & Technology, Gaithersburg, MD, USA, 2011.
[8] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalasai, “Cloud computing - the business perspective,” Decision Support Systems, vol. 51, no. 1, pp. 176–189, 2011.
[9] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, “The rise of “big data” on cloud computing: review and open research issues,” Information Systems, vol. 47, pp. 98–115, 2015.
[10] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, “Container-based operating system virtualization,” ACM SIGOPS-Operating Systems Review, vol. 41, no. 3, pp. 275–287, 2007.
[11] D. Merkel, “Docker: lightweight Linux containers for consistent development and deployment,” Linux Journal, vol. 2014, pp. 1–5, 2014.
[12] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, “Making middleboxes someone else’s problem,” ACM SIGCOMM-Computer Communication Review, vol. 42, no. 4, pp. 13–24, 2012.
[13] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, “Network function virtualization: state-of-the-art and research challenges,” IEEE Communications surveys & tutorials, vol. 18, no. 1, pp. 236–262, 2015.
[14] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: challenges and opportunities for innovations,” IEEE Communications Magazine, vol. 53, no. 2, pp. 90–97, 2015.
[15] J. Martins, M. Ahmed, C. Raiciu et al., “ClickOS and the art of network function virtualization,” in Proceedings of the 11th Symposium on Networked Systems Design and Implementation, pp. 459–473, Seattle, WA, USA, April 2014.
[16] D. G. Andersen, “Theoretical approaches to node assignment,” 2002, https://www.cs.cmu.edu/~dga/papers/andersen-assign.ps.
[17] I. Houidi, W. Louati, W. Ben Ameur, and D. Zeghlache, “Virtual network provisioning across multiple substrate networks,” Computer Networks, vol. 55, no. 4, pp. 1011–1023, 2011.
[18] L. Wang, H. Qu, J. Zhao, and Y. Guo, “Virtual network embedding with discrete particle swarm optimisation,” Electronics Letters, vol. 50, no. 4, pp. 285–286, 2014.
[19] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding,” ACM SIGCOMM-Computer Communication Review, vol. 38, no. 2, pp. 17–29, 2008.
[20] X. Cheng, S. Su, Z. Zhang et al., “Virtual network embedding through topology-aware node ranking,” ACM SIGCOMM-Computer Communication Review, vol. 41, no. 2, pp. 38–47, 2011.
[21] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, “Optimal virtual network embedding: node-link formulation,” IEEE Transactions on Network and Service Management, vol. 10, no. 4, pp. 356–368, 2013.

[22] R. Guerzoni, R. Trivisonno, I. Vaishnavi et al., “A novel approach to virtual networks embedding for SDN management and orchestration,” in Proceedings of the IEEE Network Operations and Management Symposium (NOMS), pp. 1–7, Krakow, Poland, May 2014.

[23] T. Wang and M. Hamdi, “Presto: towards efficient online virtual network embedding in virtualized cloud data centers,” Computer Networks, vol. 106, pp. 196–208, 2016.

[24] M. R. Rahman and R. Boutaba, “SVNE: survivable virtual network embedding algorithms for network virtualization,” IEEE Transactions on Network and Service Management, vol. 10, no. 2, pp. 105–118, 2013.

[25] I. Fajjari, N. Aitssadi, G. Pujolle, and H. Zimmermann, “VNE-AC: virtual network embedding algorithm based on ant colony metaheuristic,” in Proceedings of the IEEE International Conference on Communications, pp. 1–6, Kyoto, Japan, June 2011.

[26] X. Guan, X. Wan, B. Y. Choi, and S. Song, “Ant colony optimization based energy efficient virtual network embeddings,” in Proceedings of the IEEE 4th International Conference on Cloud Networking, pp. 273–278, Niagara Falls, ON, Canada, October 2015.

[27] X. Mi, X. Chang, J. Liu, L. Sun, and B. Xing, “Embedding virtual infrastructure based on genetic algorithm,” in Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 239–244, Beijing, China, December 2012.

[28] I. Pathak and D. P. Vidyarthi, “A model for virtual network embedding across multiple infrastructure providers using genetic algorithm,” Science China Information Sciences, vol. 60, no. 4, pp. 1–12, 2017.

[29] J. Su, Z. Sheng, A. X. Liu, Y. Han, and Y. Chen, “Capture-aware identification of mobile RFID tags with unreliable channels,” IEEE Transactions on Mobile Computing, p. 1, 2020.

[30] J. Su, Z. Sheng, V. C. M. Leung, and Y. Chen, “Energy efficient tag identification algorithms for RFID: survey, motivation and new design,” IEEE Wireless Communications, vol. 26, no. 3, pp. 118–124, 2019.

[31] J. Su, A. X. Liu, Z. Sheng, and Y. Chen, “A partitioning approach to RFID identification,” IEEE/ACM Transactions on Networking, vol. 28, no. 5, pp. 2160–2173, 2020.

[32] J. Su, Y. Chen, Z. Sheng, and L. Sun, “Bit query based M-ary tree protocol for RFID tags identification,” in Proceedings of the IEEE Global Communications Conference, pp. 1–7, Singapore, December 2017.

[33] J. Su, Z. Sheng, A. X. Liu, Z. Fu, and C. Huang, “An efficient missing tag identification approach in RFID collisions,” IEEE Transactions on Mobile Computing, p. 1, 2021.