Cross-Domain Virtual Network Mapping Based on Discrete Particle Swarm Optimization and Kruskal' s Algorithm in 5G Network Slicing: Postprint

Authors: Wang Xiaolei, Chen Qiang, Liu Caixia

Date: 2018-05-02T00:00:00+00:00

Abstract

5G mobile communication networks will lease resources such as data centers from multiple infrastructure providers to collaboratively construct network slices. To address the key issue of how to efficiently perform cross-domain virtual network mapping in the lifecycle management of network slices, a two-stage cross-domain mapping strategy called DPSO-K is proposed. First, based on resource bidding, node resources and inter-domain bandwidth resources are considered in a coordinated manner. Then, a cross-domain virtual network mapping based on an optimized discrete particle swarm optimization algorithm is proposed, which can effectively improve the optimization capability. For intra-domain mapping with relatively lower overhead, a fast algorithm based on Kruskal' s minimum spanning tree is proposed, aiming to shorten slice instantiation time and accelerate service deployment. Compared with the traditional method of first performing virtual network mapping to partition requests and then uniformly mapping links, this strategy considers inter-domain bandwidth overhead in request partitioning, focuses on the mapping of critical links in link mapping, and adopts a centralized management and distributed control approach to achieve effective utilization of physical network resources. Experimental results show that the algorithm can achieve higher acceptance rates with smaller additional overhead and shorter partitioning time.

Cross-Domain Virtual Network Mapping Based on Discrete Particle Swarm Optimization and Kruskal Algorithm in 5G Network Slicing

Wang Xiaolei, Chen Qiang†, Liu Caixia
(National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450001, China)

Abstract

The 5G mobile communication network will lease resources such as data centers from multiple infrastructure providers to collaboratively construct network slices. To address the critical challenge of efficiently performing cross-domain virtual network mapping in the lifecycle management of network slicing, this paper proposes a two-stage cross-domain mapping strategy called DPSO-K. First, based on resource bidding, the algorithm comprehensively considers both node resources and inter-domain bandwidth resources. Then, it introduces an optimized discrete particle swarm optimization algorithm for cross-domain virtual network mapping, which can effectively improve optimization capability. For intra-domain mapping, which incurs relatively lower overhead, a fast algorithm based on Kruskal' s minimum spanning tree is proposed to shorten slice instantiation time and accelerate service deployment. Compared with traditional methods that first partition virtual network mapping requests and then uniformly map links, this strategy considers inter-domain bandwidth overhead during request partitioning and focuses on mapping critical links during link mapping, achieving efficient utilization of physical network resources through centralized management and distributed control.

Experimental results demonstrate that the proposed algorithm can achieve higher acceptance rates with lower overhead and shorter partitioning time.

Keywords: 5G; network slicing; virtual network embedding; cross-domain mapping

---

0 Introduction

Network slicing is an ideal network architecture for the 5G era, 被视为应对互联网 OTT (Over The Top) 业务价值碾压、实现移动网价值提升的重要途径。Network slicing enables operators to partition multiple virtual end-to-end networks that share physical network resources in isolation, or to construct a logical end-to-end network based on hardware facilities from multiple infrastructure providers to jointly provide transparent service 承载. For a specific service, the underlying infrastructure can be jointly provided by multiple equipment vendors, and the geographical deployment is no longer concentrated in a single machine room, exhibiting heterogeneous and distributed characteristics [?].

Virtual Network Embedding (VNE) is a crucial step in the network slicing generation process, aiming to find physical nodes and physical paths that satisfy mapping constraints for virtual nodes and virtual links, respectively [?]. Virtual network mapping in single-domain scenarios generally assumes complete knowledge of underlying resources within that network, and relevant mapping strategies have matured [?, ?, ?, ?, ?]. However, in 5G scenarios, an end-to-end network slice traverses access networks, core networks, and the Internet. For customized networks in vertical industries, Service Providers (SPs) must lease network infrastructure from multiple Infrastructure Providers (InPs) due to resource type and price constraints, then collaboratively construct slices using resources from multiple InPs. The subnets of different InPs constitute multiple autonomous domains. Due to commercial interests and other reasons, information is not shared between different autonomous domains, rendering single-domain mapping algorithms unsuitable for cross-domain virtual network mapping in network slicing. Compared with single-domain virtual network mapping, multi-domain virtual network mapping presents several challenges: (a) Due to the potentially large scale of multi-domain networks, algorithms designed for single-domain problems can incur enormous signaling and memory overhead; (b) Due to confidentiality requirements, local controllers may only selectively provide partial intra-domain information, forcing global controllers to perform virtual network mapping based on limited information.

Researchers have conducted numerous beneficial studies on cross-domain mapping. Existing cross-domain virtual network mapping approaches mainly adopt distributed or centralized methods. Distributed cross-domain mapping is achieved through resource negotiation among SPs and InPs, and between InPs themselves, which can fully respect the intentions of both SPs and InPs and offers good scalability. However, distributed methods introduce additional network transmission costs during negotiation and cannot obtain optimal cross-domain VN mapping solutions due to the lack of global information. Centralized methods feature the emergence of a Virtual Network Provider (VNP) [?], whose main function is to simplify the information interaction and supply-demand matching process between SPs and InPs. Based on collected infrastructure information, the VNP solves cross-domain mapping schemes according to predetermined rules and strategies. Reference [?] proposes a metaheuristic-based virtual network request partitioning strategy according to the price conditions of boundary nodes and inter-domain links in multi-domain networks. Dietrich et al. propose an exact virtual network request partitioning algorithm based on underlying network boundary node information [?]. This method' s innovation lies in proposing a more refined problem model and effectively reducing additional overhead, but the increased algorithm complexity makes it difficult to apply in large-scale network scenarios.

These centralized cross-domain VN mapping approaches generally consist of two phases: (a) Virtual network partitioning phase, where InPs select the most suitable mapping target for each virtual resource from their matching sets to minimize mapping overhead, partitioning the virtual network into multiple virtual subnets and determining the InPs that carry each virtual subnet. Houidi et al. [?] first theoretically proved that virtual network partitioning is NP-Hard and proposed a heuristic algorithm for virtual network partitioning based on a fully connected physical network topology among boundary nodes; (b) Virtual subnet mapping phase, where the VNP forwards each virtual subnet request to the corresponding InPs, which can then solve the virtual subnet mapping using existing single-domain mapping methods, primarily including integer linear programming, intelligent algorithms, and heuristic algorithms.

Existing centralized cross-domain mapping strategies assume that instantiating virtual network functions consumes the same amount of node resources in different data centers and that InPs offer identical resource prices. They also abstract the cross-domain virtual network partitioning problem as a bin packing problem, simplifying each subdomain as a point. These assumptions are not well-suited for 5G network scenarios. From the node resource perspective, resource quotes from different InPs to SPs are often different in practice. From the link resource perspective, each subdomain generally has multiple egress nodes, and selecting different egress nodes affects resource costs, so subdomains cannot be simply abstracted as points.

To address these cross-domain mapping problems, this paper proposes a cross-domain mapping algorithm DPSO-K that comprehensively considers both node resources and link resources. The mapping scheme consists of two parts: inter-domain mapping and intra-domain mapping. Since information between different autonomous domains is not public, the algorithm first considers only node resources and inter-domain link costs for resource allocation and function deployment, then adopts a distributed mapping mechanism for intra-domain mapping. The rationality of this two-step simplification lies in the fact that inter-domain link costs constitute the main cost in this problem, while intra-domain mapping costs are relatively small [?]. Therefore, the intra-domain mapping portion can be solved using a concise and fast algorithm. The main contributions of this paper are: (a) Proposing an inter-domain mapping algorithm based on particle swarm optimization that encodes VN partitioning schemes in matrix form and iteratively searches from multiple initial solutions to gradually approach the global optimum; (b) Proposing an intra-domain approximation algorithm based on Kruskal’ s minimum spanning tree that quickly obtains mapping results conforming to virtual network topology constraints through modifications to the minimum spanning tree.

1 Cross-Domain Mapping Architecture and Model for Network Slicing

1.1 Problem Description

In traditional multi-domain virtual network mapping problems, service providers need to continuously negotiate with multiple InPs when establishing a cross-domain virtual network request. The 5G core network will adopt an SDN architecture, so in software-defined 5G network slicing, the control plane and data plane will be completely separated. This architectural advancement facilitates improved forwarding efficiency and unified resource management. The centralized cross-domain mapping network architecture for network slicing is shown in Figure 1 [Figure 1: see original paper]:

a) **Infrastructure Providers (InPs):** Including access network InPs, core network InPs, Internet InPs, etc.

b) **Management and Orchestration (MANO):** Receives virtual network requests from service providers of various autonomous domains and performs virtual network request partitioning and mapping based on network information and underlying physical resource status uploaded from each autonomous domain.

c) **Virtual Network Provider (VNP):** Composed of multiple Virtual Machines (VMs) and Virtual Infrastructure Monitors (VIMs). A local VIM manages one autonomous domain, calculates the information that needs to be uploaded to MANO, and allocates virtual resources to virtual network requests. VNP and InPs are collectively called virtual network function infrastructure (NFVI).

d) **Service Providers (SPs):** Send virtual network requests to the physical network' s MANO through Operation/Business Support Systems (OSS/BSS).

To facilitate description of the virtual network mapping problem, Figure 1 is simplified and abstracted. As shown in Figure 2 [Figure 2: see original paper], this is a schematic diagram of a cross-domain mapping physical network with three subdomains. Assuming three InPs construct three network domains, where N-type nodes are internal nodes that can build VMs (or containers) to carry virtual network functions, and B-type nodes are boundary nodes that serve as gateways for information exchange between different network domains. The goal of cross-domain mapping is to select appropriate N-type nodes to carry virtual network functions and appropriate B-type nodes and links to carry virtual network links, minimizing physical network node computation costs and link bandwidth costs while satisfying virtual network requests.

Due to the practical requirements of slice function deployment in data centers, some node mappings have additional constraints beyond resource constraints. If an embedding satisfies the constraints, it is called a valid mapping. For a virtual request, multiple valid mappings may exist. The optimization objective of this paper is to minimize overall cost and improve resource utilization. Based on the model in reference [**?**], this paper defines the cost of cross-domain virtual network mapping as:

1.2 Model Establishment

The underlying physical resource network is represented by a weighted undirected graph $G^s = (N^s, E^s)$. The total computing resources on underlying physical node $n^s \in N^s$ are $c(n^s)$, and the available bandwidth of the physical link between underlying physical nodes n_i^s and n_j^s is $b(n_i^s, n_j^s)$. Similar to the underlying physical resource network, the virtual network, as a subgraph of the underlying physical network, is also represented by a weighted undirected graph $G^v = (N^v, E^v)$, where the set of underlying physical network nodes is represented as $\{n_1^v, n_2^v, ..., n_{|N^v|}^v\}$. The set of underlying physical network links is represented as E^v.

The mapping of a virtual network request onto the underlying physical resource network can be expressed as $M : G^v(V^v, E^v) \rightarrow G^s(V^s, E^s)$. During the mapping process, the remaining available CPU resources are represented as $R_c(n^s) = c(n^s) - \sum_{n^v \in N^v} S_{n^v,n^s} \cdot M_R^c(n^v, n^s)$, where $c(n^s)$ represents the total CPU resources, and $\sum_{n^v \in N^v} S_{n^v,n^s} \cdot M_R^c(n^v, n^s)$ represents the occupied CPU resources. Here, S_{n^v,n^s} indicates whether virtual node n^v is mapped to physical node n^s. Similarly, the remaining available bandwidth resources are $R_b(e^s) = b(e^s) - \sum_{e^v \in E^v} S_{e^v,e^s} \cdot M_R^b(e^v, e^s)$, which is the difference between total link bandwidth and occupied bandwidth.

Virtual network mapping has resource constraints, meaning the underlying network must have sufficient remaining resources to carry the virtual network, characterized as:

For node resources: $\sum_{n^v \in N^v} M_R^c(n^v, n^s) \leq c(n^s)$

For link resources: $\sum_{e^v \in E^v} M_R^b(e^v, e^s) \leq b(e^s)$

Additionally, some nodes have location constraints characterized as:

$$Loc(n^s, n^v) = \begin{cases} 1, & \text{if deployable} \\ 0, & \text{if not deployable} \end{cases}$$

If an embedding satisfies the constraints, it is called a valid mapping. For a virtual request, there may exist multiple valid mappings. This paper’ s optimization objective is to minimize overall cost and improve resource utilization. Based on the model in reference [**?**], this paper defines the cost of cross-domain virtual network mapping as:

$$Cost = \alpha \sum_{n^v \in N^v} G_C(n^v) + (1 - \alpha) \sum_{e^v \in E^v} B_{length}(e^v)$$

where $G_C(n^v)$ represents the node resource cost, $B_{length}(e^v)$ represents the bandwidth cost of the virtual link, and α is the weighting factor.

2 DPSO-K Algorithm Description

The DPSO-K algorithm performs inter-domain mapping first, followed by intra-domain mapping. The inter-domain mapping phase constitutes the main source of overhead and is the primary optimization target of the DPSO-K algorithm. Based on the inter-domain mapping results, resources within each autonomous domain are then mapped. Since intra-domain mapping incurs relatively small overhead, an approximation algorithm is adopted for fast solution to ensure rapid service deployment.

2.1 Inter-Domain Mapping Algorithm

The inter-domain algorithm primarily addresses the virtual network partitioning problem. Based on an optimized Discrete Particle Swarm Optimization (DPSO) algorithm, this algorithm completes the partitioning of cross-domain virtual network mapping, comprehensively considering inter-domain link resources and node resource costs to achieve global resource optimization. The algorithm inputs are the physical network G^s and virtual network request G^v, and the output is the inter-domain mapping scheme, including both node mapping and link mapping components. This optimization scheme serves as part of the input for the intra-domain algorithm described in Section 2.2.

Figure 3 [Figure 3: see original paper] shows the global network view obtained by MANO. MANO does not understand the specific network topology of each subdomain but can learn about resource types and approximate distributions within subdomains through node bidding and other methods. The inter-domain algorithm primarily considers inter-domain link resources and node resource costs for resource allocation and function deployment.

For inter-domain link cost, this problem has been proven to be NP-hard [?]. Therefore, metaheuristic algorithms are typically used for such problems, aiming to find approximate optimal solutions within acceptable time. For node resource cost, node bidding is a business game between InPs and SPs. The main process is: SP requests are sent to MANO, which records the node and link requirement information of SP requests. Controllers in each InP send their domain's available node resource information and unit prices to the information processing center. For example, InP1 submits to MANO: can provide physical nodes v1, v2, v3 with resource amounts of 20, 5, 5 respectively, and resource unit price of 12. The processing center analyzes the information from various InPs and SP requests to formulate a Bidding Results Table (BRT), and MANO calculates the optimal virtual node mapping results based on the algorithm.

2.1.1 Encoding Method Each particle is set as a vector where each component represents the physical node to which a virtual node is mapped. For a given virtual network, the number of virtual nodes determines the vector length, the numbering of virtual nodes determines component positions, and the physical node number to which a virtual node is mapped determines the component value. Thus, each particle represents a virtual network node mapping scheme. When historical factors guide the process, whether to adjust the position vector is determined by the direction vector. Components in the direction vector correspond one-to-one with components in the position vector, taking values of 0 or 1. A value of 1 indicates adjustment of that component in the position vector, while 0 indicates no adjustment. During perturbation, the position vector can be randomly generated.

2.1.2 Fitness Function The quality of each node mapping is measured by a fitness function, defined as:

$$f(L) = \sum_{(u,v)\in E^v} LEN(u,v) \cdot BW(u,v)$$

where $LEN(u,v)$ refers to the length (in hops) of the physical path to which virtual link (u,v) is mapped, and $BW(u,v)$ refers to the bandwidth of the virtual link. The fitness value is determined by the link mapping results obtained from the node mapping scheme.

2.1.3 Iterative Optimization The optimized discrete particle swarm algorithm is a multi-solution iterative process. The basic idea is to find optimal solutions through collaboration and information sharing among individuals in the population. The specific iterative optimization process is:

a) Initialize the position vector and other variables, set iteration count $z = 0$.

b) Obtain the direction vector through direction adjustment operations, then use position adjustment operations to adjust the position vector according to the direction vector, generating a new position vector.

c) Calculate all available shortest paths between each pair of physical nodes corresponding to virtual links using the K-shortest path algorithm. If link mapping fails and the maximum backtrack count has not been exceeded, return to step (2); if the maximum backtrack count is reached, return link mapping failure.

d) Obtain the fitness value for this position vector using Equation (1).

e) Evaluate the fitness value. If it is smaller than a certain individual in the current elite group and the new node vector is not in the elite group, update the current elite group.

f) Compare the new fitness value with the particle’ s previous fitness value. If the new fitness value is lower and the number of flights in this direction is less than the maximum direction flight count N_s, proceed to step g); otherwise, enter the next iteration and return to step (2).

g) Call the position adjustment operation based on the determined direction vector.

h) Calculate all available shortest paths between each pair of physical nodes corresponding to virtual links using the K-shortest path algorithm. If physical link mapping succeeds, proceed to step i); if link mapping fails and the maximum backtrack count has not been exceeded, return to step g).

i) $z = z + 1$. Determine whether the iteration count has been reached. If not, return to step b).

2.1.4 Parameter Operations The position adjustment operation changes particle i’ s position, i.e., re-determines the value of each component j. The physical meaning is that virtual node j reselects a physical node for matching. The position adjustment operation is executed through multiple direction adjustment operations. The direction adjustment operation determines the value of particle i’ s direction component j according to an adjustment probability. The value is $P_{ij} = \frac{r_h}{m}$, where r_h is a random adjustment factor introduced by the algorithm to escape local optima, and m is the number of individuals in the historical optimal set. The algorithm analyzes historical data by examining the top m optimal individuals. For component j of each optimal individual, it compares with particle i’ s component j, counting the number of differences as r_h. Then, a random number p in the range [0,1] is generated. If $p > P_{ij}$, component j is adjusted. The direction adjustment operation is affected by perturbation operations, meaning the adjustment of component j is inverted according to the perturbation probability P_{disp}.

2.2 Intra-Domain Mapping Algorithm

Based on the inter-domain mapping results, InPs perform intra-domain virtual network mapping within their respective domains. Figure 4 [Figure 4: see original paper] shows the view of each subdomain. The regional controller understands the resource distribution and network topology within its domain, and boundary nodes have already been selected.

This paper proposes a fast mapping algorithm based on Kruskal’ s minimum spanning tree algorithm. According to the underlying physical network resource status, the algorithm aims to minimize cost and operates as follows:

a) Based on inter-domain mapping results, input the location information and bandwidth information of egress nodes.

b) Calculate the set of available physical nodes for each virtual node based on virtual node resources, location constraints, etc.

c) Calculate the minimum weight paths between candidate physical nodes in the underlying physical network using a minimum weight routing algorithm according to the network resource status in the cross-domain environment.

d) Based on Kruskal' s minimum spanning tree algorithm, dynamically select the minimum weight physical path from the minimum weight path set in each iteration. After obtaining the minimum spanning tree, map the virtual nodes and adjust the virtual links to output the final mapping result.

3 Simulation and Analysis

3.1 Simulation Environment

The simulation experiments were conducted on a computer configured with 3 GB memory, 32-bit Windows 7 operating system, and Intel Core 2 T9550 processor. MATLAB was used for programming, and both physical network and virtual network topologies were randomly generated using the GT-ITM tool.

The preset parameters are as follows: the entire underlying network consists of 5 different InPs, the number of virtual node types is set to $s = 8$, and the number of VN requests is $n = 1000$. For each InP, the resources r provided for each virtual node type follow a uniform distribution $[0, r_{max}]$, and the resource unit price P follows a uniform distribution $[10, 20]$. The control resource information from each InP is centralized in the management center to form a unified resource table. The number of virtual nodes in VN requests follows a uniform distribution $[1, s]$, where the resource demand r of each node follows a uniform distribution $[0, 20]$. The optimization objective is to minimize the total mapping cost defined in Equation (3), with both node resource cost weight α and cross-domain cost weight β set to 1.

The main parameters of the particle swarm algorithm are set as shown in Table 1 .

Table 1: Simulation Parameters

Parameter	Value
Particle count S	50
Maximum iterations N_z	100
Particle velocity upper limit	5
Particle velocity lower limit	-5
Perturbation operation probability P_{disp}	0.15
Elite group size $BestN$	10

Parameter	Value

Table 2: Average Mapping Time of Each Algorithm (s)

Algorithm	Average Mapping Time
HTF	12.3
MC-VNM	15.7
DPSO-K	18.5

As shown in Table 2, although the DPSO-K algorithm can reduce overhead, its computation time is longer because the particle swarm-based cross-domain mapping component introduces resource bidding, increasing algorithm complexity. However, the increased computation time remains in the same order of magnitude, and the additional time overhead is acceptable in practical applications.

3.2 Simulation Results and Analysis

By analyzing the shortcomings of existing cross-domain virtual network mapping algorithms, this paper proposes a hierarchical resource management model for cross-domain network environments to dynamically manage network resources within each autonomous domain. Under the premise of fully considering cross-domain network environments and differentiated physical link bandwidth, different link weights are set for intra-domain link bandwidth and inter-domain backbone link bandwidth. Based on the physical network resource status in all autonomous domains, cross-domain virtual network mapping operations are implemented with the optimization objective of minimum-cost virtual network mapping. With primary consideration of inter-domain overhead and the goal of minimizing global resource total overhead subject to resource constraints, a discrete particle swarm optimization model for virtual network mapping is established. Theoretical analysis and experimental results demonstrate that the proposed DPSO-K algorithm effectively reduces the resource cost of virtual network mapping, achieves good network performance, and maximizes infrastructure provider revenue.

Figure 5 [Figure 5: see original paper] compares the total mapping overhead of the three algorithms under different node quantities. It can be observed that the DPSO-K algorithm reduces total mapping overhead to some extent by comprehensively considering node resources and link resources and utilizing discrete particle swarm optimization. This verifies the algorithm' s effectiveness. Additionally, as the number of virtual request nodes increases, the cost of virtual network mapping increases at a constant rate. When the problem scale is larger, such as in large-scale network environments, this solution remains applicable.

Figure 6 [Figure 6: see original paper] shows the impact of different numbers of data domains on total mapping overhead for the three algorithms. In this scenario, the number of physical network nodes and virtual network nodes remains consistent, but the number of data domains in the physical network changes. It can be seen that as the number of data domains increases, inter-domain mapping introduces additional bandwidth overhead. The proposed algorithm can ensure smaller total overhead compared with contrast algorithms under different numbers of data domains, verifying the algorithm' s stability.

Many issues in virtual network mapping require further research and resolution. Current research primarily focuses on effectively mapping virtual network requests with static network resource requirements. Due to the scarcity of network resources, dynamic migration and scheduling of service function chains in 5G networks will be considered in future work.

References

[1] Mano T, Inoue T, Ikarashi D. Efficient virtual network optimization across multiple domains without revealing private information [C]// Proc of the 23rd International Conference on Computer Communication and Networks. 2014: 1-8.

[2] Shen Meng, Xu Ke, Yang Kun. Towards efficient virtual network embedding across multiple network domains [C]// Proc of the 22nd International Symposium of Quality of Service. 2014: 61-70.

[3] 肖蔼玲，王颖，孟洛明，等．基于知识描述和遗传算法的跨域虚拟网络映射 [J]．软件学报，2014, 25 (10): 2189-2205.

[4] Ye Z, Cao X, Wang J. Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization [J]. IEEE Network, 2016, 30 (3): 81-87.

[5] Yousaf F Z, Taleb T. Fine-grained resource-aware virtual network function management for 5G carrier cloud [J]. IEEE Network, 2016, 30 (2): 110-115.

[6] Garg S, Dwivedi R K, Chauhan H. Efficient utilization of virtual machines in cloud computing using synchronized throttled load balancing [C]// IEEE Next Generation Computing Technologies. 2015: 77-80.

[7] Baumgartner A, Reddy V S, Bauschert T. Mobile core network virtualization: a model for combined virtual core network function placement and topology optimization [C]// Proc of IEEE Conference on Network Softwarization. 2015: 1-9.

[8] Gonzalez A, BarraA E, Beghelli A. A sub-graph mapping-based algorithm for virtual network allocate on over flexible grid networks [C]// IEEE Transparent Optical Networks. 2015: 1-4.

[9] Guo K, Wang Y, Qiu. Particle swarm optimization based multi-domain virtual network embedding [C]// Proc of International Federation for Information Processing (IFIP) //IEEE International Symposium on Integrated Network Management. 2015: 798-801.

[10] 贾伟, 夏靖波, 跨域虚拟网络映射问题研究 [J]. 电子与信息学报, 2016, 38 (3): 728-734.

[11] Gong S, Chen J, Yin X, et al. Survivable virtual network embedding across multiple domains [C] [C]// Proc of IEEE International Conference on Computer & Communications. 2017: 2391-2396.

[12] Lv Bo, Wang Zhenkai, Huang Tao, et al. Virtual resource organization and virtual network embedding across multiple domains [C]// Proc of International Conference on Multimedia Information Networking & Security. 2010: 725-728.

[13] Jia W, Xia J. Research on virtual network embedding across multiple domains [J]. Journal of Electronics & Information Technology, 2016, 38 (3): 728-734.

[14] Shen M, Xu K, Yang K, Chen H. Towards efficient virtual network embedding across multiple network domains [J]. Quality of Service, 2014: 61-70.

[15] Dahir M, Alizadeh H, Gözüpek D. Energy efficient virtual network embedding in federated software defined networks [C]// Proc of Signal Processing & Communications Applications Conference. 2017.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.