IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 21, 2020, accepted May 6, 2020, date of publication May 15, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994912

A Service Function Chain Deployment Method Based on Network Flow Theory for Load Balance in Operator Networks

XIAOYANG HAN[1], XIANGRU MENG[1], ZHENHUA YU[2], QIAOYAN KANG[1], AND YU ZHAO[3]

[1]College of Information and Navigation, Air Force Engineering University, Xi'an 710077, China
[2]College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China
[3]College of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China

Corresponding author: Zhenhua Yu (zhenhua_yu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873277, Grant 71571190, and Grant 61401499, and in part by the Key Research and Development Program of Shaanxi Province under Grant 2019GY-056.

ABSTRACT Network function virtualization technology makes the deployment and management of network service more flexible and elastic by decoupling network function from dedicated hardware. The service requests of network function virtualization are usually deployed in the form of a service function chain. In order to solve the problems of load imbalance, unreasonable utilization of substrate resources, and the high delay of the service function chain deployment in operator networks, a service function chain deployment method based on the network flow theory is proposed in this paper. First, on the basis of perceiving the substrate network resources and topology with a software-defined network controller in real time, a candidate node set is determined according to the resource constraints and the locations of ingress/egress switch nodes that service flow flows in/out. Second, the candidate node set, the ingress/egress switch nodes and the connection between them are used to form a directed network, and the service function chain deployment problem is transformed into an optimal path selection problem. Then, a node disassembling method is used to transform the directed network into a capacity-flow-cost network. Finally, a minimum-cost maximum-flow algorithm is used to find the optimal deployment path and complete the service function chain deployment. Experiments show that the method proposed in this paper can guarantee the load balance of operator networks, reduce the average transmission delay of service flow, and make the utilization of substrate resources more reasonable.

INDEX TERMS Network function virtualization, service function chain, node disassembling method, virtual network functions combination, minimum-cost maximum-flow.

I. INTRODUCTION

The original intention of network function virtualization technology (NFV) is to realize network functions by means of ''general hardware+software'', reduce the capital expenditure and operating expense of networks, and improve the flexibility of service deployment. The NFV has gradually become a key technology to promote the development of 5G and future network [1]–[3]. The use of a low-cost high-performance generic computing platform in network infrastructure can greatly reduce the cost of operators. Network functions can be virtualized into instances of plain software referred to virtual network function (VNF) in the NFV, which can greatly shorten the deployment cycle and improve the deployment flexibility and elasticity of network service [4], [5].

As a kind of new developing network technology, software-defined network (SDN) has been widely applied in recent years. Its characteristics, such as separating the control plane from the data plane and centralized control based on the global perspective, significantly improve the flexibility of network management and utilization of substrate resources [6]. The SDN and NFV are highly complementary: SDN can further simplify the NFV deployment and reduce the burden of operation and maintenance. On the other hand, NFV can provide SDN with infrastructure support. The combination of SDN and NFV technology can significantly improve the efficiency, programmability and flexibility of networks, which is widely used in 5G networks [7], [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Yue Zhang.

While NFV technology has many advantages, it also faces some challenges in the deployment process. In operator networks, different types of service flows are required to be served by different types of VNFs, and service function chain (SFC) is formed by these VNFs connected in a predefined order. In SFC deployment, the bandwidth resources, forwarding resources and computing resources of the substrate network should be occupied simultaneously. The SFC deployment is also an NP-hard problem like the virtual network mapping [9], [10]. Therefore, deciding how to design an efficient and reasonable SFC deployment strategy is particularly important. It is related not only to the revenue of operators, but also the quality of service (QoS) provided by operators. This demand makes the SFC deployment a hot spot in the field of industry and academia.

With respect to the issue of the SFC deployment, many researchers have proposed solutions. To deal with the load imbalance phenomenon in SFC deployment, the authors in [11] design two deployment methods to balance the load of the substrate network, and achieve good results. However, these methods only aim at minimizing the forwarding cost, and the consumption of substrate resource has not been fully considered. The work in [12] aims to improve the overall performance of the SFC deployment and uses a discrete particle swarm optimization algorithm to conduct the SFC deployment. In [13], a linear programming formulation is established with the goal of minimizing the resource scheduling delay, and a heuristic algorithm is designed based on a greedy algorithm to determine the resource scheduling method. In [14], the SFC deployment in 5G networks with the goal of resource optimization and QoS demand is studied. Sun *et al.* [15] propose a reliability-aware approach for SFC deployment and improve the resource utilization of the substrate network. The problem of resource orchestration in 5G networks with the goal of QoS and efficient utilization of network resources is studied in [16]. The authors in [17] propose an SFC deployment method to ensure efficient resource utilization and low delay. Wang *et al.* [18] study the SFC deployment in internet of things, and propose a new deployment method that can perceive resources and service in real time. In [26], the SFC deployment is studied and a novel algorithm is proposed to optimize the operating expense. The simulation results show that this method can significantly reduce service cost. Kim *et al.* [27] propose an SFC deployment strategy to improve QoS. However, unilaterally emphasizing QoS would affect the revenue of operators.

Most of the studies only optimize the single performance of SFC deployment, but they have not fully considered the load balance, QoS and reasonable utilization of the substrate resource. This work proposes an SFC deployment method based on the network flow theory (SFCD-NFT) that not only guarantees the load balance of the substrate network, but also considers the QoS and the revenue of operators. First, the global perspective of SDN controller is used to perceive substrate resources in real time, and a candidate node set is determined according to the resource demand of SFC and the locations of ingress/egress switch nodes. Second, the candidate node set is used to form a directed network, and the SFC deployment problem is transformed into a task of finding the optimal path in the directed network. Then a node disassembling method is used to construct a capacity-flow-cost network based on the obtained directed network. Finally, taking the link transmission delay as the cost and the SFC bandwidth demand as the constraint, a minimum-cost maximum-flow algorithm of network flow theory is used to find the optimal deployment path and complete the SFC deployment.

This work gives full consideration to VNF combination and eliminates bottleneck nodes in the candidate node selection stage. It chooses the substrate network path that has the least bottleneck links and bottleneck forwarding nodes as the optimal deployment path, and therefore guarantees the load balance of the substrate network, greatly reduces the link transmission delay and improves the performance of SFC deployment. The experiments show that the method can guarantee the load balance of the substrate network, and improve the overall QoS and resource utilization.

The main contributions of this paper can be summarized as follows.

(i) The SFC deployment problem is transformed into an optimal path selection problem. According to the locations of ingress/egress switch nodes and the resource demand of VNF, a candidate server node set of VNF is determined. Then the candidate server node set, the ingress/egress switch nodes and the connection between them are used to form a directed network, and thus the SFC deployment problem is transformed into a problem of finding the optimal path in the directed network. VNF combination is fully considered and bottleneck nodes are eliminated in the candidate node selection stage. It is beneficial to the load balance of the substrate network.

(ii) The obtained directed network is converted into a capacity-flow-cost network by a node disassembling method, and a minimum-cost maximum-flow algorithm is used to find the optimal deployment path of SFC in the capacity-flow-cost network. The path that has the least bottleneck nodes/links and required transmission delay is selected as the optimal deployment path. Taking the link transmission delay as the cost and using a minimum-cost maximum-flow algorithm to solve the optimal path selection problem can make the result closer to the optimal solution, reduce the average transmission delay of service flow and improve the resource utilization of the substrate network.

The rest of this paper is organized as follows. In Section II, some related works are reviewed. Section III gives the problem statement. In Section IV, a network model and evaluation indicators are provided. Section V illustrates the integer linear programming formulation of the SFC deployment. The SFCD-NFT is elaborated in Section VI. Section VII validates and evaluates the proposed method with extensive simulations and experiments. We conclude this paper in Section VIII.

II. RELATED WORK

The tight coupling of infrastructure hardware and network service in the traditional network leads to high capital expenditure and operating expense. To solve this problem, some researchers have tried to use NFV to decouple infrastructure hardware and network service, which can reduce costs of the service deployment and network operation and improve the flexibility of service deployment. In the NFV technology, an SFC is formed by multiple VNFs connected in a predefined order to provide users with various end-to-end network services. The SFC deployment problem has become a hot spot in recent years. As the stuides in [2], [7], [8] have made a relatively detailed overview of SFC deployment, this section simply summarizes the SFC deployment problem.

The work in [19] proposes an SFC deployment strategy with the goal of improving resource utilization, but the scenario setting is relatively simple. The work in [20] proposes a heuristic algorithm to solve the deployment problem of SFC with the goal of minimizing the overall delay, and achieves good effect, but the objective function of this method is incomplete. A heuristic algorithm is proposed in [21] for dynamic SFC deployment, which balances the minimum resource consumption and the minimum resource scheduling operation. In [22], a linear programming formulation with reliability perception and delay constraint is established to maximize the service reliability and minimize the delay of SFC. An optimal approximation algorithm is proposed in [23] to solve the problem of SFC deployment, which improves the reliability of SFC and reduces the deployment cost. However, this method cannot guarantee the deploying success ratio.

The emergence of many lightweight virtualization technologies, such as docker and linux container technologies, makes it possible for deploying multiple VNFs simultaneously on the same high-performance server. The work in [24] studies the SFC deployment in 5G networks, and proposes a method of VNF combination and mapping temporary link to reduce network resource consumption. Wang *et al.* [25] propose a method to conduct joint-optimization of SFC and resource allocation, and discuss the influence of virtual machine reusing and VNF combination on the performance of SFC deployment. Mechtri *et al.* [28] propose an SFC deployment method that aims at maximizing the revenue of network operators and satisfying the service demands of users. An optimal deployment method of SFC is proposed in [29], which is scalable to network topology. However, it does not take into account the reasonable using of substrate resources, and the overall deployment performance still has room for improvement. Sun *et al.* [30] propose a backtracking mechanism to solve the initial failure of SFC deployment, thus significantly improving SFC deploying success ratio.

The work in [31] proposes a heuristic algorithm that first builds a multicast routing tree between the source node and the destination nodes, and then places the VNFs along the built routing tree. He *et al.* [32] studies the optimal placement of VNFs with multiple instances to minimize cost, as well as guarantee the load balance. The work in [33] proposes an efficient deployment algorithm for online SFC requests, and the algorithm is expected to guarantee the load balance and improve the deploying success ratio. These studies allow for one-to-many and many-to-one VNF embeddings to improve the load balance. However, the splitting of service flow will cause performance degradation.

The above studies mostly pursue single performance, failing to fully consider the overall performance of SFC deployment, such as load balance, QoS and the reasonable use of substrate resources. This is also the motivation of this paper.

III. PROBLEM STATEMENT

Existing SFC deployment strategies mostly pursue single performance, leading to the unreasonable use of the substrate network resources, mainly including: (i) The use of the substrate network resources is unbalanced and there are some bottleneck nodes and links in the substrate network. (ii) The unreasonable use of the substrate network resources may also lead to the problems of increased transmission delay and reduced revenue. This paper studies the reasonable using of the substrate network resources and performance optimization of the SFC deployment from the perspective of operators.

A. VNF COMBINATION STRATEGY

Lightweight virtualization technologies make it possible to deploy multiple VNFs simultaneously on the same high-performance server. Therefore, multiple adjacent VNFs of an SFC can be deployed on the same server node under the condition of meeting resource constraints and VNF types constraints, called VNF combination. The VNF combination mechanism can effectively improve the utilization of substrate network resources, reduce link transmission delay and improve SFC deployment performance, as shown in Figure 1.

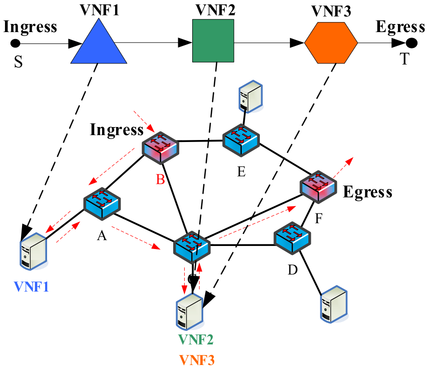


FIGURE 1. An example of VNF combination.

Assume that the number of VNF types that the server node can host is m. Then the adjacent VNFs whose number is m can be deployed in the same server node on the basis of meeting the VNF type constraint and resource constraint. Otherwise, the VNF combination is not allowed to avoid a Ping-pong routing problem. For example, if $m = 2$, the i-th VNF and the $(i-1)$-th VNF can be deployed on the same server node, and the i-th VNF cannot be combined with other VNFs except the $(i-1)$-th and $(i+1)$-th VNF. Otherwise, it is easy to generate the Ping-pong routing problem, as shown in Figure 2.

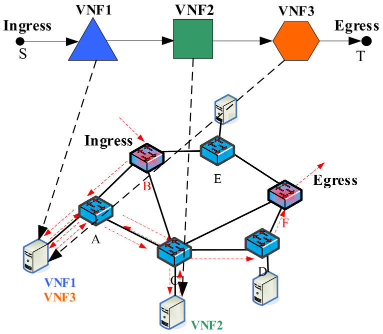


FIGURE 2. An example of Ping-pong routing problem.

B. SFC DEPLOYMENT STRATEGY

From the perspective of operators, SFC deployment should ensure the low delay demands of 5G services, enhance the utilization of substrate resources as much as possible and reduce resource fragmentation, so as to improve the revenue of operators.

In this paper, the SFC deployment problem is transformed into an optimal path selection problem that is solved by the network flow theory. First, the candidate server node set of VNFs is determined according to the locations of the ingress/egress switch nodes and the resource demand of SFC. Then the ingress/egress switch nodes, the candidate server node set and the connection between them are used to form a directed network, and the SFC deployment problem is transformed into an optimal path selection problem. Next, the directed network is transformed into a capacity-traffic-cost network by a node disassembling method. Finally, link transmission delay is taken as the cost of optimal path selection, and the bandwidth demand of service flow is taken as the constraint, the path that has the lower delay and the least bottleneck nodes/links is selected as the optimal deployment path by a minimum-cost maximum-flow algorithm.

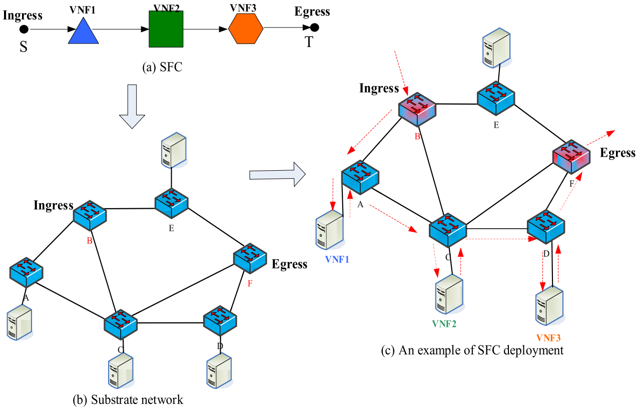
Due to the full consideration of the VNF combination and bottleneck nodes/links in the process of optimal path selection, the load balance of the substrate network is improved and the overall performance of SFC deployment can be guaranteed.

IV. NETWORK MODEL AND EVLUATION INDICATORS

A. NETWORK MODEL

1) SFC REQUEST

An SFC request can be represented by a weighted undirected graph $\boldsymbol{G_v} = (S, T, V, E_v, d_v, D_d)$. For simplicity, we assume service traffic comes from one single ingress switch node and ends at the single egress switch node, S and T represent the ingress switch node and egress switch node, respectively. $V = \{v_1, v_2, \ldots, v_P\}$ represents the set of VNFs that are arranged in a predefined order. When the VNFs are deployed on server nodes, the computing resources will be consumed. E_v represents the virtual link set of SFC, $e_v^{ij} \in E_v$ represents the virtual link between v_i and v_j, and $bw(e_v^{ij})$ represents the bandwidth demand of the virtual link e_v^{ij}. When a service flow traverses through switch nodes, it needs to consume the forwarding resources of switch nodes. When an SFC is deployed, it needs to occupy the computing and bandwidth resources of a substrate network. d_v represents the delay constraint of the SFC. D_d is defined as the actual transmission delay of the SFC that is successfully deployed, and it should be less than or equal to d_v, as shown in Figure 3(a).



(a) SFC

(b) Substrate network

(c) An example of SFC deployment

FIGURE 3. An example of SFC deployment.

2) SUBSTRATE NETWORK

This paper selects operator networks as the application scenario, which is composed of a core network and a number of servers (or data centers) connected to it, as shown in Figure 3(b). The weighted undirected graph $\boldsymbol{G_s} = (N_s, E_s, D_s)$ can be used to describe the substrate network, where $N_s = N_f \cup N_c$, $N_f = \{n_f^1, n_f^2, \ldots, n_f^n\}$ represents the switch nodes with fixed forwarding capacity, and $N_c = \{n_c^1, n_c^2, \ldots, n_c^n\}$ represents the server nodes with fixed computing resources and types of VNFs that can be hosted. Due to the high-speed fiber connection between a pair of switch node and server node, the link transmission delay between them is negligible. In this paper, E_s is used to represent the set of links between the switch nodes, and $e_s^{lm} \in E_s$ represents the substrate inter-switch link between the n_f^l and n_f^m. D_s represents the link transmission delay set of E_s, and $d_s^{lm} \in D_s$ represents the link transmission delay of the e_s^{lm}.

3) SFC DEPLOYMENT

When an SFC request arrives, the operator instantiates the VNFs according to the predefined order that the service flow traverses through the VNFs. SFC deployment needs to consume the computing resources, forwarding resources and bandwidth resources of the substrate network, as shown in Figure 3(c). In order to prevent performance degradation caused by the splitting of service flow, this paper stipulates that a flow cannot be split into more than two paths.

B. EVALUATION INDICATORS

1) DEPLOYING SUCCESS RATIO

Deploying success ratio is one of the main indicators to describe the deployment performance and overall resource utilization of SFC. It can be defined as follows:

$$r = \lim_{T_d \to \infty} \frac{\sum_{t=0}^{T_d} SFC_{suc}(t)}{\sum_{t=0}^{T_d} SFC(t) + \delta} \tag{1}$$

where $SFC(t)$ is the number of SFC requests at time t, $SFC_{suc}(t)$ is the number of SFC requests successfully deployed at time t, T_d is the total operating time of operators, and δ is a constant that is infinitely close to 0.

2) LONG-TERM AVERAGE REVENUE TO COST RATIO

For SFC request $\boldsymbol{G}_v$, define revenue $R(\boldsymbol{G}_v, t)$ and cost $C(\boldsymbol{G}_v, t)$ as follows:

$$\begin{aligned} R(\boldsymbol{G}_v, t) = \alpha_1 \sum_{v_i \in V} cpu(v_i) + \alpha_2 \sum_{v_i \in V} rforward(v_i) \\ + \alpha_3 \sum_{e_v^{ij} \in E_v} bw(e_v^{ij}) \end{aligned} \tag{2}$$

$$\begin{aligned} C(\boldsymbol{G}_v, t) = \beta_1 \sum_{v_i \in V} cpu(v_i) + \beta_2 \sum_{v_i \in V} cforward(v_i) \\ + \beta_3 \sum_{e_v^{ij} \in E_v} hops(D(e_v^{ij})) \times bw(e_v^{ij}) \end{aligned} \tag{3}$$

where $cpu(v_i)$ represents the required CPU resources of v_i, $rforward(v_i)$ denotes the required forwarding resources of v_i, and $cforward(v_i)$ stands for the actually consumed forwarding resources of v_i. $bw(e_v^{ij})$ represents the required bandwidth resources of e_v^{ij}, $D(e_v^{ij})$ denotes the actual deployment path of e_v^{ij}, and $hops(D(e_v^{ij}))$ is the links hops of $D(e_v^{ij})$. α_1, α_2, and α_3 are weighting coefficients of $cpu(v_i)$, $rforward(v_i)$ and $bw(e_v^{ij})$ in $R(\boldsymbol{G}_v, t)$, respectively. $\alpha_1 + \alpha_2 + \alpha_3 = 1$. β_1, β_2, and β_3 are weighting coefficients of $cpu(v_i)$, $cforward(v_i)$ and $bw(e_v^{ij})$ in $C(\boldsymbol{G}_v, t)$, respectively. $\beta_1 + \beta_2 + \beta_3 = 1$.

Usually, the long-term average revenue to cost ratio is used to represent the performance of SFC deployment method under steady state, and it can be defined as follows:

$$R/C = \lim_{T_d \to \infty} \frac{\sum_{t=0}^{T_d} \sum_{\boldsymbol{G}_v \in SFC_{suc}(t)} R(\boldsymbol{G}_v, t)}{\sum_{t=0}^{T_d} \sum_{\boldsymbol{G}_v \in SFC_{suc}(t)} C(\boldsymbol{G}_v, t)} \tag{4}$$

3) LINK LENGTH EXPANSION COEFFICIENT

In this paper, the link length expansion coefficient is defined as the expanded proportion of the total substrate path hops that virtual links deployed compared to the sum hops of SFC that are successfully deployed. Denoted by K_1, the link length expansion coefficient reflects the effective utilization of substrate link resources in SFC deployment, as defined as follows:

$$K_1 = \frac{\sum_{e_v^{ij} \in E_v} hops(D(e_v^{ij}))}{L_v} - 1 \tag{5}$$

where $\sum_{e_v^{ij} \in E_v} hops(D(e_v^{ij}))$ is the sum of substrate path hops deployed by virtual link e_v^{ij}, and L_v is the sum of SFC hops deployed successfully.

4) AVERAGE TRANSMISSION DELAY

The transmission delay of SFC is the time that it takes for packets to travel from an ingress switch node through the VNFs to an egress switch node. It includes queueing delay, processing delay, and link transmission delay, and it shall be less than or equal to the end to end delay demand [34]. In the scenario where flow splitting is not allowed, the processing delay and queueing delay are mainly related to the hardware performance and packet size. However, the link transmission delay is directly affected by the performance of SFC deployment algorithm. This paper defines the average transmission delay as the average value of link transmission delay of successfully deployed SFCs. It denoted as K_2 can reflect the performance of SFC deployment, as defined as follows:

$$K_2 = \frac{\sum_{p \in SFC_{suc}(t)} D_d^p}{\sum_{t=0}^{T_d} SFC_{suc}(t)} \tag{6}$$

where D_d^p represents the actual transmission delay of the p-th successfully deployed SFC.

5) PROPORTION OF BOTTLENECK NODES/LINKS

The substrate switch nodes and substrate inter-switch links with load over 95% are defined as bottleneck nodes and links, respectively. The proportion of bottleneck nodes and links can reflect the utilization of substrate resources, and they can be described as follows:

$$K_3 = \frac{L_{linkload-0.95}}{L_s} \tag{7}$$

$$K_4 = \frac{N_{nodeload-0.95}}{F_1} \tag{8}$$

where K_3 is the proportion of bottleneck links and K_4 is the proportion of bottleneck nodes. $L_{linkload-0.95}$ is the total number of bottleneck links, and L_s is the number of all substrate inter-switch links. $N_{nodeload-0.95}$ is the number of bottleneck nodes, and F_1 is the number of all substrate switch nodes.

6) PROPORTION OF IDLE NODES/LINKS

The substrate switch node and links with resource utilization less than 5% are defined as idle nodes and links, respectively. The proportion of idle nodes and links can reflect the utilization of substrate resources, and they can be described as follows.

$$K_5 = \frac{L_{linkload-0.05}}{L_s} \tag{9}$$

$$K_6 = \frac{N_{nodeload-0.05}}{F_1} \tag{10}$$

where K_5 is the proportion of idle links, K_6 is the proportion of idle nodes, $L_{linkload-0.05}$ is the number of idle links, and $N_{nodeload-0.05}$ is the number of idle nodes.

V. INTEGER LINEAR PROGRAMMING FORMULATIONS

The integer linear programming formulation of SFC deployment is formulated and the objective function and constraints are given in this section.

This paper aims at minimizing the average transmission delay of SFC in operator networks on the premise of ensuring a balanced utilization of substrate network resources. The objective function is as follows.

$$\min \frac{\sum_{p=1}^{N} D_d^p}{N}$$

$$N = \lim \left\{ \sum_{t=0}^{T_d} SFC_{suc}(t) \right\} \tag{11}$$

where N represents the number of all successfully deployed SFCs at time T_d.

$$x_i^l = \begin{cases} 1, & \text{if } v_i \text{ is deployed to } n_c^l \\ 0, & \text{otherwise,} \end{cases}$$
$$\forall v_i \in V, \forall n_c^l \in N_c \tag{12}$$

$$y_{ij}^{lm} = \begin{cases} 1, & \text{if } e_v^{ij} \text{ is deployed to } e_s^{lm} \\ 0 & \text{otherwise,} \end{cases}$$
$$\forall e_v^{ij} \in E_v, \forall e_s^{lm} \in E_s \tag{13}$$

Constraint (12) indicates that if the VNF v_i is deployed to the server node n_c^l, $x_i^l = 1$. Otherwise, $x_i^l = 0$. Constraint (13) denotes that if the virtual link e_v^{ij} is deployed to the substrate link e_s^{lm}, $y_{ij}^{lm} = 1$. Otherwise, $y_{ij}^{lm} = 0$.

$$\sum_{v_i \in V} x_i^l \leq m, \quad \forall n_c^l \in N_c \tag{14}$$

$$\sum_{n_c^l \in N_c} x_i^l = 1, \quad \forall v_i \in V \tag{15}$$

$$x_i^l \times cpu(v_i) \leq cpu(n_c^l),$$
$$\forall v_i \in V, \ \forall n_c^l \in N_c \tag{16}$$

$$\sum_{e_v^{ij} \in E_v} y_l^m \times bw(e_v^{ij}) \leq b(e_s^{lm}),$$
$$\forall e_v^{ij} \in E_v, \forall e_s^{lm} \in E_s \tag{17}$$

$$\sum_{e_v^{ij} \in E_v} y_{ij}^{lm} \times bw(e_v^{ij}) \leq forward(n_f^l),$$
$$\forall n_f^l \in N_f, \ \forall e_v^{ij} \in E_v, \ \forall e_s^{lm} \in E_s \tag{18}$$

$$z_i^l = \begin{cases} 1, & \text{if } f_i \text{ is satisfied by } n_c^l \\ 0, & \text{otherwise,} \end{cases}$$
$$\forall v_i \in V, \forall n_c^l \in N_c \tag{19}$$

Constraint (14) indicates that the number of VNF types hosted by the server node are m. Constraint (15) means that each VNF of an SFC can only be deployed to one server node. Constraint (16) denotes that the residual CPU computing resources of a server node cannot be less than the CPU resource demand of VNF. Constraint (17) represents that the residual bandwidth resources of a substrate network link should be larger than the link resource demand of an SFC. Constraint (18) shows that the residual forwarding capacity of a substrate switch node should be greater than the forwarding resource demand of a service flow to be deployed, and $forward(n_f^l)$ represents the residual forwarding capacity of the substrate switch node n_f^l. The constraint (19) indicates that the VNF type demand must be satisfied by the server node, and f_i represents the type demand of v_i.

$$\sum_{\mathrm{e}_s^{lm} \in E_s} y_{ij}^{lm} - \sum_{\mathrm{e}_s^{ml} \in E_s} y_{ji}^{ml} = x_i^l - x_j^m,$$
$$\forall e_s^{lm} \in E_s, \forall e_s^{ml} \in E_s \tag{20}$$

$$D_d^p \leq d_v^p, \quad \forall p \in \sum_{t=0}^{T_d} SFC_{suc}(t) \tag{21}$$

Constraint (20) means that the virtual link should be deployed to the loopfree path of the substrate network to prevent the Ping-pong routing problem. Constraint (21) indicates that SFCs deployed successfully must meet the delay demands of service flows, and d_v^p represents the delay constraint of the p-th successfully deployed SFC.

This integer linear programming formulation is known to be an NP-hard problem, and therefore the SFCD-NFT is proposed to solve it.

VI. SFCD-NFT

In order to improve the SFC deployment performance and guarantee the load balance of the substrate network, this paper proposes the SFCD-NFT that first transforms an SFC deployment problem into an optimal path selection problem and then uses a minimum-cost maximum-flow algorithm to solve it.

A. THE PROCESS OF SFCD-NFT

As shown in Figure 4, first, a candidate node set is obtained according to the locations of the ingress/egress switch nodes and resource demands of VNFs. The candidate node set, the ingress/egress switch nodes and the connections between them are used to form a directed network, and thus the SFC deployment problem is transformed into an optimal path selection problem. Then, the obtained directed network is transformed into a capacity-traffic-cost network by a node disassembling method. Finally, a minimum-cost maximum-flow algorithm is used to find the maximum flow path with

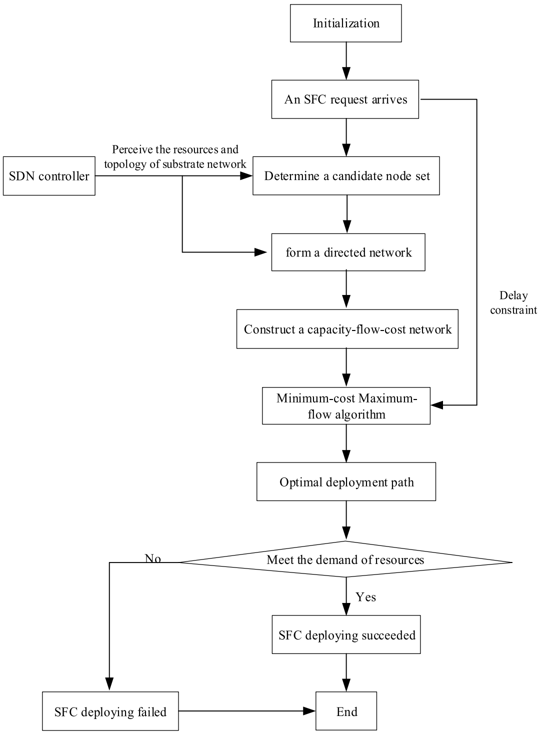

Initialization
An SFC request arrives
Perceive the resources and topology of substrate network
SDN controller
Determine a candidate node set
form a directed network
Delay constraint
Construct a capacity-flow-cost network
Minimum-cost Maximum-flow algorithm
Optimal deployment path
No
Meet the demand of resources
Yes
SFC deploying succeeded
SFC deploying failed
End

FIGURE 4. The process of SFCD-NFT.

the minimum cost in the capacity-traffic-cost network, and the result is employed to complete the SFC deployment.

Due to the full consideration of resource utilization and VNF combination in the SFC deployment, the load of the substrate network is more balanced and the SFC deployment path length is significantly reduced.

B. DETERMINE A CANDIDATE NODE SET

The candidate node set of VNFs can be determined by the following factors:

- the locations of the ingress/egress switch nodes.
- the given order of the VNFs in an SFC.
- the length L of an SFC request that is equal to the hops between the first VNF and the last VNF.
- the length R of the shortest path between an ingress switch node and an egress switch node.

In order to prevent the QoS problem caused by an excessive long deployment path, the deployment locations of VNFs should be restricted. The load parameter N_{load} of a substrate node is introduced to monitor the load of server nodes and the switch nodes in real time through an SDN controller. Define the hops between the ingress switch node and the l-th server node n_c^l as $I(n_c^l)$, and the hops between the egress switch node and the l-th server node n_c^l as $O(n_c^l)$.

For the i-th VNF v_i, the server nodes that meet the following constraints should be selected as the candidate nodes:

1. $I(n_c^l) >= i - 1$ & $I(n_c^l) <= i + 3$;

$O(n_c^l)$ >={max(R, L) − i-1} & $O(n_c^l)$ <={max(R, L) − $i + 3$};

$I(n_c^l)$ and $O(n_c^l)$ restrict the deployment location of v_i;

2. $cpu(v_i) <= cpu(n_c^l)$;

$cpu(v_i)$ represents the computing resource demand of v_i, and $cpu(n_c^l)$ represents the residual computing resource of the l-th server node n_c^l;

3. $f_i \in \mathrm{C}(n_c^l)$;

$\mathrm{C}(n_c^l)$ represents the service types that the n_c^l can host. This means that the type of v_i is contained in the types of VNF that can be hosted by the server node n_c^l;

4. $N_{load}(n_c^l) < 0.95$ & $N_{load}(n_f^l) < 0.95$;

$N_{load}(n_c^l)$ is the real-time load of the server node n_c^l, and $N_{load}(n_f^l)$ is the real-time load of the switch node n_f^l. This means that the candidate server node n_c^l and the switch node n_f^l connected to n_c^l are not bottleneck nodes.

The specific process of determining a candidate node set is shown in **Algorithm 1**.

Algorithm 1 Determine a Candidate Node Set

```
Input: the real-time resource and topology of the substrate network, $L$, $R$
Output: the candidate node set $N_{LIST}$
an SDN controller perceive the resource and topology of the substrate network in real time
for $i = 1: q$
    for $l = 1: r$
        if $N_{load}(n_c^l) < 0.95$ & $N_{load}(n_f^l) < 0.95$ & $f_i \in \mathrm{C}(n_c^l)$
          & $cpu(v_i) <= cpu(n_c^l)$ & $i - 1 <= I(n_c^l) <= i+1$
          & $\{\max(R, L) - i - 1\} <= O(n_c^l) <=$
          $\{\max(R, L) - i + 3\}$
            $n_c^l$ is an eligible server node, and put it in the candidate set of $v_i$;
            return a candidate set $n(i)_{LIST}$ of $v_i$
        end if
    end for
end for
$N_{LIST} = \{n(1)_{LIST}; n(2)_{LIST}; ...; n(q)_{LIST}\}$
```

According to the result of **Algorithm 1**, a candidate node set of VNFs can be obtained. Use the candidate server node set N_{LIST}, the ingress/switch nodes and the inter-switch links between them to form a directed network $\boldsymbol{G_d}$, as shown in Figure 5. The SFC deployment problem is converted to an optimal path selection problem of finding a path from the ingress switch node through the candidate node to the egress

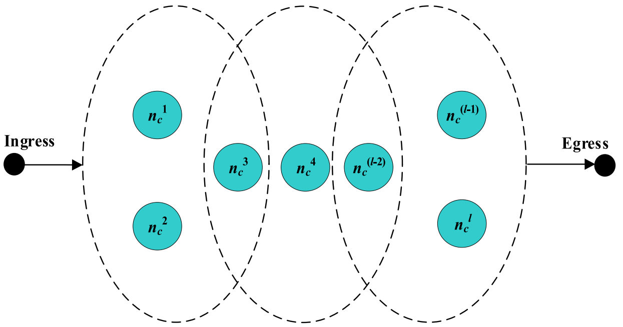
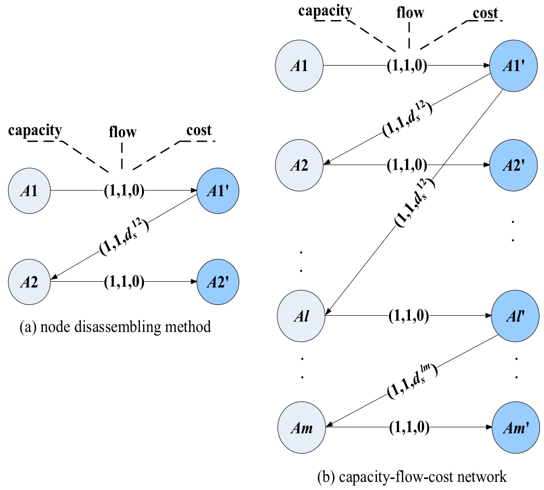


FIGURE 5. The example of directed network Gd.

switch node. This path must pass all the candidate node set of VNFs, and each candidate node set can be passed only once. The overlapping nodes in the candidate node set can be used as VNF combination deployment nodes to reduce the link delay.

C. CONSTRUCT A CAPACITY-FLOW-COST NETWORK

In order to find the optimal deployment path in the directed network $\boldsymbol{G_d}$, it is necessary to convert the directed network $\boldsymbol{G_d}$ into a capacity-flow-cost network by using a node disassembling method, as shown in Figure 6. First, nodes of $\boldsymbol{G_d}$ are numbered, each node Al is disassembled into a node pair Al and Al', and an edge connects Al to Al'. Then, the capacity and the flow of the edge are defined to be both equal to 1, and the cost between a node pair is set 0 (equivalent to the cost from oneself to oneself). If node Al is directly connected to Am, Al' is connected to Am by an edge. Finally, the link delay d_s^{lm} between switch nodes of the substrate network is used as the cost, and the directed network $\boldsymbol{G_d}$ is thus converted to a capacity-flow-cost network $\boldsymbol{G_c}$, as shown in Figure 6.



(a) node disassembling method

(b) capacity-flow-cost network

FIGURE 6. The example of a node disassembling method.

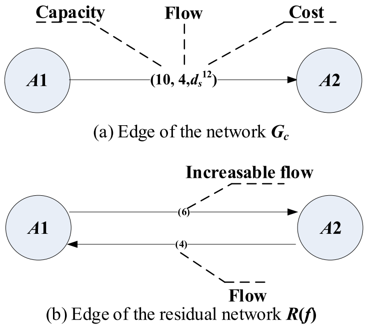
The specific process of node disassembling method is shown in **Algorithm 2**.

Algorithm 2 Node Disassembling Method

```
Input: A directed network $\boldsymbol{G_d}$
Output: A capacity-flow-cost network $\boldsymbol{G_c}$
(1) number the nodes of $\boldsymbol{G_d}$
(2) disassemble the node $Al$ into a node pair $Al$ and $Al'$
(3) use an edge to connect the node $Al$ and node $Al'$
(4) define the capacity and flow of an edge to be both equal
to 1, the cost between a node pair $Al$ and $Al'$ is 0
(5) if node $Al$ is directly connected to $Am$
(6)     $Al'$ is connected to $Am$ by an edge
(8)     use $d_s^{lm}$ as the cost
(9) end if
(10)    return the capacity-flow-cost network $\boldsymbol{G_c}$
```

D. MINIMUM-COST MAXIMUM FLOW ALGORITHM

Using the link transmission delay of the substrate network as the cost, a minimum-cost maximum-flow algorithm is adopted to solve the problem of the SFC deployment, which can ensure that the deployment path has the minimum transmission delay and maximum flow, and guarantee the performance of the SFC deployment. In order to state the problem clearly, the concept of residual network is introduced. Each capacity-flow-cost network $\boldsymbol{G_c}$ and a flow on it correspond to a residual network $\boldsymbol{R(f)}$ whose nodes are the same as the network $\boldsymbol{G_c}$. The residual network reflects the residual link capacity, as shown in Figure 7.



(a) Edge of the network $\boldsymbol{G_c}$

(b) Edge of the residual network $\boldsymbol{R(f)}$

FIGURE 7. Example of a residual network.

The specific process of minimum-cost maximum-flow algorithm is shown in **Algorithm 3**.

Finally, the SFC deployment is completed after occupying the computing resource, forwarding resource and link bandwidth resource on the optimal deployment path.

E. COMPLEXITY ANALYSIS

For each SFC flow, the time complexity of SFCD-NFT includes the time complexity of determining a candidate node set, the node disassembling method and the minimum-cost maximum-flow algorithm. F_2 is the number of all VNFs in an SFC request, and the complexity of determining a candidate node set is at most $O(F_1{*}F_2$. The time complexity of the node disassembling method is $O(F_1)$. The complexity of the minimum-cost maximum-flow algorithm is $O(L_s{*}\log F_1)$. Thus, the total time complexity of SFCD-NFT at the worst situation is $O(L_s{*}\log F_1 + F_1{*}F_2 + F_1)$. Comparing with other SFC deployment algorithms, the deployment method proposed in this paper has lower time complexity.

VII. SIMULATION

In this paper, MATLAB is used for simulation and the SFCD-NFT proposed in this paper is evaluated in a large-scale network scenario and compared with three other algorithms.

A. SIMULATION ENVIRONMENT

The substrate network topology and SFC topology used in the simulation experiments are generated by the improved Salam network topology random generation algorithm. Set the substrate switch node and the server node to be deployed in the

Algorithm 3 Minimum-Cost Maximum-Flow Algorithm

```
Input: An SFC request $G_v$, a capacity-flow-cost network $G_c$
Output: the optimal deployment path $ODpath$
k = 0, the minimum-cost maximum-flow path $f^k = 0$(minimum cost flow with zero traffic), and the set of minimum-cost maximum-flow path $MCMFpathlist = 0$
for i = 1: $k$
    construct the residual network $R(f^k)$
    the link bandwidth demand is used as the constraint, the link transmission delay as the cost, and the depth-first search algorithm is applied to find the minimum cost path from the ingress switch node to the switch egress node in $R(f^k)$
    if the minimum cost path $P$ is nonexistent
        $f^k$ is path of the maximum flow with the minimum cost, end the calculation
        put the $f^k$ into the $MCMFpathlist$
    else
        update $f^k$ to $f^k + P$, update the $R(f^k)$
    end if
end for
return $MCMFpathlist$
if $MCMFpathlist$ is empty or $MCMFpathlist$ cannot meet the delay constraint
        return deploying failed
    else
        choose the path with the least bottleneck nodes/links as the optimal deployment path
end if
    return optimal deployment path $ODpath$
if $ODpath$ meet SFC computing resource constraint
    return deploying success
else
    return deploying failed
end if
```

same location of operator networks, their numbers are both 100, and the link connection probability between the switch nodes is 0.5. The computing resources of server nodes and the forwarding resources of switch nodes obey the uniform distribution of [50]–[60], and the link bandwidth between the substrate network switch nodes obeys the uniform distribution of [20]–[25]. The link transmission delay of the substrate network obeys the uniform distribution of [1], [2]. It is assumed that each server node can host any two types of $\{v_1, v_2, v_3, v_4\}$.

The ingress switch node and egress switch node of service flow are randomly determined according to an SFC request. The number of VNFs contained in an SFC request obeys the uniform distribution of [2], [4], and the types of VNFs are different. The computing resource demands of VNFs obey the uniform distribution of [7], [10], and the bandwidth demands of SFCs obey the uniform distribution of [5], [8]. The maximum allowable transmission delay of SFC obeys the uniform distribution of [6], [8]. The arrival ratio of SFC requests obeys the Poisson distribution with the parameter 0.1 and the life time obeys the exponential distribution with parameter 1500.

The duration of simulation experiments is 10,000 time units. In order to eliminate the effect of random factors on the experimental results, the simulation experiment is conducted for 10 times, and its average value is taken as the final simulation result.

B. EXPERIMENTAL ANALYSIS

Two sets of experiments are set up to verify the SFCD-NFT proposed in this paper. In the first set of experiments, the SFCD-NFT is compared with three latest research methods under the same experimental conditions, as shown in Table 1. The second set of experiments analyzes the performance of the SFCD-NFT in the utilization of the substrate network resources.

TABLE 1. Comparision of SFC deployment methods.

Method	Description
SFCD-NFT	With the aim of load balance, the deployment problem of SFC is transformed into an optimal path selection problem, the VNF combination strategy and a minimum-cost maximum-flow algorithm is used to find the optimal deployment path to complete the SFC deployment.
SFCD-FOCL	With the aim of minimizing link resources and computing resources consumption, the virtual machine reusing strategy, VNF combination strategy and temporary link mapping strategy are applied in the SFC deployment. And the two-level game theory is used to achieve Nash equilibrium between minimizing the link resources and minimizing computing resources [24].
NFV-CG	With the aim of improving resource utilization ratio, the decomposition method is used to establish an exact mathematical model, and the column generation algorithm is used to solve the problem. The result is used to complete the SFC deployment [29].
JoraNFV	With the aim of minimizing the resource consumption, this paper proposes a three-stage joint-optimization deployment method of NFV allocation, which includes a one-hop optimal traffic scheduling algorithm and a multi-path greedy algorithm for VNF composition and forwarding graph embedding, respectively. The SFC deployment is completed according to the result [25].

1) EXPERIMENT 1: COMPARISION OF SFC DEPLOYMENT METHODS

This set of experiments mainly compares and verifies the performance of the SFCD-NFT from four evaluation indicators: deploying success ratio, long-term average revenue to cost ratio, average transmission delay and average link expansion coefficient.

As shown in Figures 8 and 9, the SFCM-FOCL utilizes the virtual machine reusing strategy, VNF combination strategy and temporary link mapping strategy to complete the SFC deployment. The deploying success ratio remains at 0.5 and the long-term average revenue to cost ratio at 0.55. The JoraNFV makes the joint-optimization for the three stages

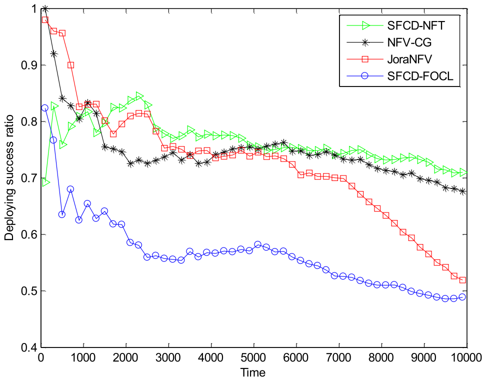


FIGURE 8. The deploying success ratio.

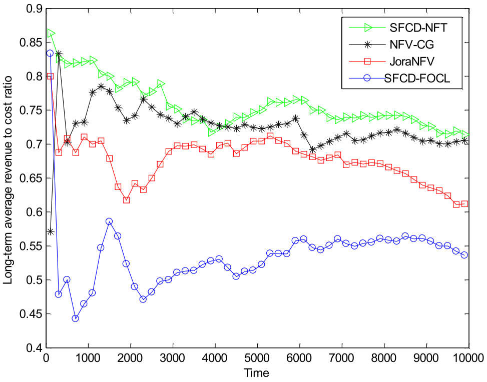


FIGURE 9. The long-term average revenue to cost ratio.

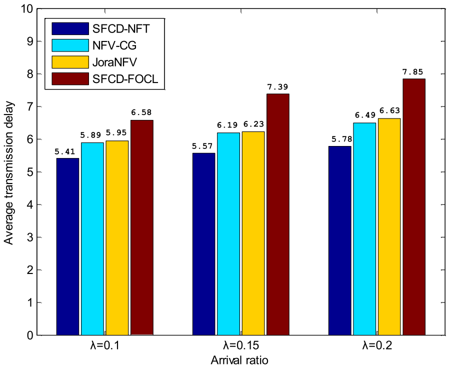


FIGURE 10. The average transmission delay.

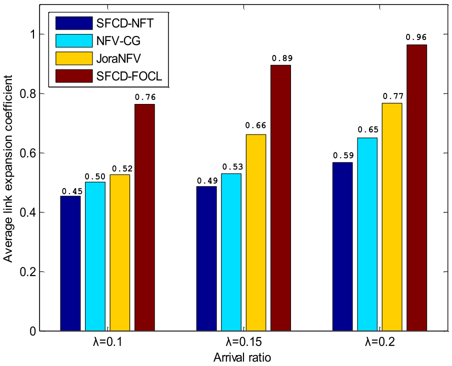


FIGURE 11. The average link expansion coefficient.

of NFV resource allocation with the goal of optimizing resource utilization. The deploying success ratio remains above 0.5 and the long-term average revenue to cost ratio above 0.6, which are slightly higher compared with the SFCM-FOCL. The NFV-CG uses the decomposition method to establish an exact mathematical model of the SFC deployment, and solves it by the column generation algorithm. Due to the excellent performance of column generation algorithm in solving large-scale linear programming problem, the NFV-CG obtains relatively good results, the deploying success ratio and the long-term average revenue to cost ratio remain at 0.67 and 0.7, respectively.

The above three methods all aim at improving resource utilization and achieve good results. However, there is still room for improvement. The three methods are prone to cause the excessive resource consumption and resource fragmentation, because the reasonable utilization of the substrate network resources is not fully considered. In contrast, the SFCD-NFT proposed in this paper takes into account the balanced utilization of the substrate network resources. The deploying success ratio is maintained above 0.7 and the long-term average revenue to cost ratio is maintained around 0.72. Overall, the SFCD-NFT has the best performance in the four SFC deployment methods.

As shown in Figures 10 and 11, the average transmission delay and average link expansion coefficient of the four methods rise with the increased SFC request arrival ratio steadily. This is because the average length of deployment path and the transmission delay of SFC increase with the consumption of the substrate network resources. However, when the arrival ratio λ is equal to 0.1, 0.15 and 0.2, the average transmission delays of the SFCD-NFT are equal to 5.41, 5.57, 5.78 and the average link expansion coefficients are equal to 0.45, 0.49, 0.59, respectively. The average transmission delay and average link expansion coefficient of the SFCD-NFT are lower than the three other methods. With the increase of the arrival ratio, the variations of the average transmission delay and average link expansion coefficient are not large. These verify the performance of the SFCD-NFT in terms of the average transmission delay and the average link expansion coefficient.

2) EXPERIMENT 2: ANALYSIS OF THE SUBSTRATE NETWORK RESOURCE UTILIZATION

This set of experiments mainly analyzes the substrate network resource utilization of the SFCD-NFT, and compares it with the three other methods from four evaluation indicators: proportions of bottleneck nodes and idle nodes, proportions of bottleneck links and idle links.

As seen from Figures 12 and 13, because the JoraNFV and SFCM-FOCL fail to fully consider the node real-time load of the substrate network in the SFC deployment, the proportions of bottleneck nodes and idle nodes are both high,

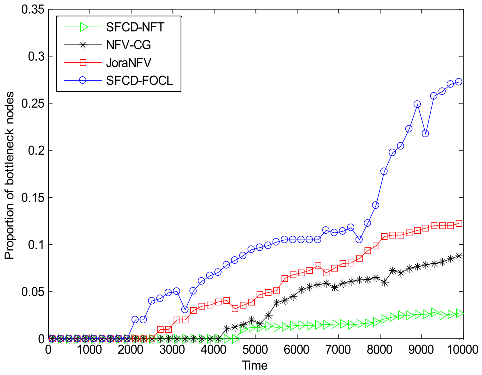


FIGURE 12. The proportion of bottleneck nodes.

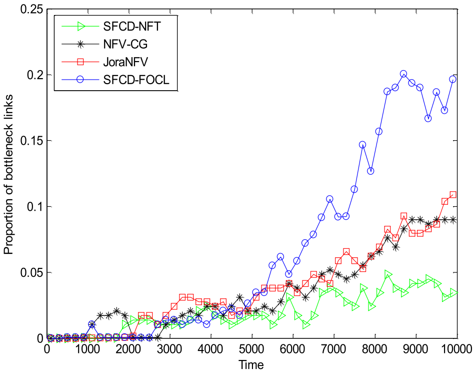


FIGURE 14. The proportion of bottleneck links.

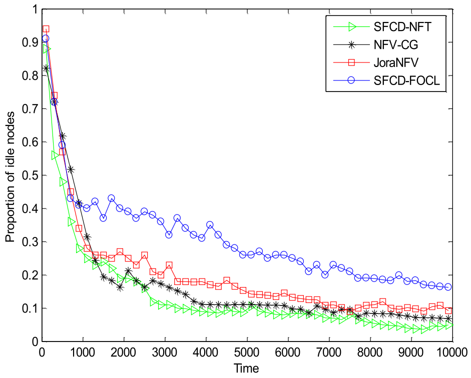


FIGURE 13. The proportion of idle nodes.

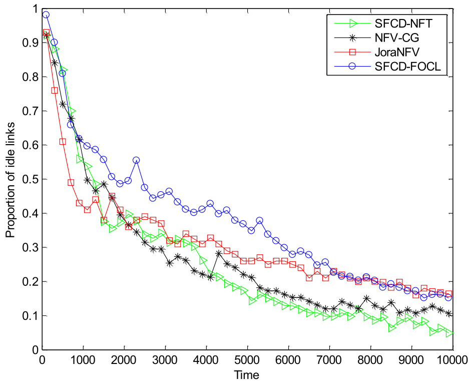


FIGURE 15. The proportion of idle links.

and grow rapidly over time, leading to the unreasonable utilization of the substrate network resources. The NFV-CG algorithm uses a decomposition method to establish an exact mathematical model of the SFC deployment and solves it by the column generation algorithm. Due to the superior performance of the column generation algorithm in solving large-scale linear programming problems, the proportions of bottleneck nodes and idle nodes are significantly lower than that of the JoraNFV and the SFCM-FOCL. The SFCD-NFT considers the resource utilization in the node selection stage and chooses the nodes with lower load as the candidate nodes. Therefore, the proportions of bottleneck nodes and idle nodes are minimal, and the node load of the substrate network is relatively balanced.

As shown in Figures 14 and 15, because the JoraNFV and the SFCM-FOCL fail to give full consideration to the link real-time load in the SFC deployment, the proportions of bottleneck links and idle links are higher, and the proportions of bottleneck links grow quickly with the consuming of substrate link resources. The utilization of substrate link resources is not reasonable. Due to the advantages of an exact mathematical model and the performance of the column generation algorithm in solving large-scale linear programming problems, the bottleneck links proportion and idle links proportion of the NFV-CG are both lower than that of the JoraNFV and the SFCM-FOCL. The SFCD-NFT takes full account of the resource utilization and eliminates the bottleneck nodes/links in the link selection. Therefore, the proportions of bottleneck links and idle links are minimal and the link load of the substrate network is more balanced. This experiment verifies the performance of the SFCD-NFT.

According to the above two sets of experiments, the SFCD-NFT proposed in this paper maintains better performance compared with the three other methods. This verifies the superiority of the SFCD-NFT.

VIII. CONCLUSION

In this paper, the SFCD-NFT is proposed to meet the demand of load balance, low delay and efficient utilization of substrate resources in operator networks. First, a candidate node set and a directed network are determined, and the SFC deployment problem is transformed into an optimal path selection problem. Then, a node disassembling method is used to transform the obtained directed network into a capacity-flow-cost network. Finally, a minimum-cost maximum-flow algorithm is used to solve the optimal path selection problem and complete the SFC deployment. Experiments show that the SFCD-NFT can guarantee the load balance of the substrate network and the reasonable using of the substrate network resources, reduce the average transmission delay of service flow, and thus improve the revenue of operators. We will study the SFC deployment in a more realistic scenario and try to use more practical simulation platforms to improve the credibility of experimental results in the future work.

REFERENCES

[1] H.-C. Hsieh, J.-L. Chen, and A. Benslimane, "5G virtualized multi-access edge computing platform for IoT applications," *J. Netw. Comput. Appl.*, vol. 115, pp. 94–102, Aug. 2018.

[2] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN—Key technology enablers for 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017.

[3] J. Tian, B. Wang, T. Li, F. Shang, and K. Cao, "Coordinated cyber-physical attacks considering DoS attacks in power systems," *Int. J. Robust Nonlinear Control*, to be published.

[4] M. Otokura, K. Leibnitz, Y. Koizumi, D. Kominami, T. Shimokawa, and M. Murata, "Evolvable virtual network function placement method: Mechanism and performance evaluation," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 27–40, Mar. 2019.

[5] S. H. He, K. Xie, X. Zhou, T. Semong, and J. Wang, "Multi-source reliable multicast routing with QoS constraints of NFV in edge computing," *Electronics*, vol. 8, no. 10, pp. 1–20, 2019.

[6] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1038–1051, Apr. 2016.

[7] J. de Jesus Gil Herrera and J. F. B. Vega, "Network functions virtualization: A survey," *IEEE Latin Amer. Trans.*, vol. 14, no. 2, pp. 983–997, Feb. 2016.

[8] G. Mirjalily and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," *Chin. J. Electron.*, vol. 27, no. 4, pp. 704–717, Jul. 2018.

[9] Y. Su, X. Meng, Q. Kang, and X. Han, "Survivable virtual network link protection method based on network coding and protection circuit," *IEEE Access*, vol. 6, pp. 67477–67493, Dec. 2018.

[10] X. Han, X. Meng, Q. Kang, and Y. Su, "Survivable virtual network link shared protection method based on maximum spanning tree," *IEEE Access*, vol. 7, pp. 92137–92150, Jul. 2019.

[11] X. Zhu and Q. Zhang, "Resource optimization algorithm of combination of NFV and SDN for application of multiple services," *J. Commun.*, vol. 39, no. 11, pp. 54–62, Nov. 2018.

[12] D. Ma, L. Zhuang, and J. Lan, "Discrete particle swarm optimization based multi-objective service path constructing algorithm," *J. Commun.*, vol. 38, no. 2, pp. 94–105, Jan. 2017.

[13] R. Xu, W. Wang, X. Gong, and X. Que, "Delay-aware resource scheduling optimization in network function virtualization," *J. Comput. Res. Develop.*, vol. 55, no. 4, pp. 738–747, 2018.

[14] R. Vilalta, A. Mayoral, R. Casellas, R. Martinez, and R. Munoz, "Experimental demonstration of distributed multi-tenant cloud/fog and heterogeneous SDN/NFV orchestration for 5G services," in *Proc. Eur. Conf. Netw. Commun.*, Athens, Greece, Jun. 2016, pp. 52–56.

[15] J. Sun, G. Zhu, G. Sun, D. Liao, Y. Li, A. K. Sangaiah, M. Ramachandran, and V. Chang, "A reliability-aware approach for resource efficient virtual network function deployment," *IEEE Access*, vol. 6, pp. 18238–18250, Apr. 2018.

[16] R. Vilalta, A. Mayoral, R. Casellas, R. Martinez, and R. Muñoz, "SDN/NFV orchestration of multi-technology and multi-domain networks in cloud/fog architectures for 5G services," in *Proc. 21st OptoElectron. Commun. Conf. (OECC)*, Niigata, Japan, Jul. 2016, pp. 1–3.

[17] G. Sun, G. Zhu, D. Liao, H. Yu, X. Du, and M. Guizani, "Cost-efficient service function chain orchestration for low-latency applications in NFV networks," *IEEE Syst. J.*, vol. 13, no. 4, pp. 3877–3888, Dec. 2019.

[18] J. Wang, H. Qi, K. Li, and X. Zhou, "PRSFC-IoT: A performance and resource aware orchestration system of service function chaining for Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1400–1410, Jun. 2018.

[19] Z. Chen, G. Feng, B. Liu, and Y. Zhou, "Construction policy of network service chain oriented to resource fragmentation optimization in operator network," *J. Electron. Inf. Technol.*, vol. 40, no. 2, pp. 763–769, Apr. 2018.

[20] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, Sep. 2016.

[21] Y. Liu, H. Zhang, H. Guan, and Y. Wang, "A method for adaptive resource adjustment of dynamic service function chain," *IEEE Access*, vol. 6, pp. 69988–70004, Nov. 2018.

[22] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[23] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE Conf. Comput. Commun.*, Hong Kong, Apr. 2015, pp. 1346–1354.

[24] D. Zhao, J. Ren, R. Lin, S. Xu, and V. Chang, "On orchestrating service function chains in 5G mobile network," *IEEE Access*, vol. 7, pp. 39402–39416, Apr. 2019.

[25] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, Nov. 2016.

[26] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Niagara Falls, Canada, Oct. 2015, pp. 255–260.

[27] T. Kim, S. Kim, K. Lee, and S. Park, "A QoS assured network service chaining algorithm in network function virtualization architecture," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, Shenzhen, China, May 2015, pp. 1221–1224.

[28] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.

[29] N. Huin, B. Jaumard, and F. Giroire, "Optimal network service chain provisioning," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1320–1333, Jun. 2018.

[30] G. Sun, Y. Li, D. Liao, and V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 1175–1191, Sep. 2018.

[31] O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "Joint VNF placement and multicast traffic routing in 5G core networks," in *Proc. IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[32] W. He, S. Guo, Y. Liang, and X. Qiu, "Markov approximation method for optimal service orchestration in IoT network," *IEEE Access*, vol. 7, pp. 49538–49548, Apr. 2019.

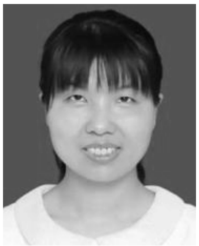
[33] G. Sun, Z. Chen, H. Yu, X. Du, and M. Guizani, "Online parallelized service function chain orchestration in data center networks," *IEEE Access*, vol. 7, pp. 100147–100161, Aug. 2019.

[34] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.

XIAOYANG HAN was born in Zhumadian, China, in 1986. He received the B.S. and M.S. degrees from Air Force Engineering University, China, in 2009 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include network function virtualization and network security.

XIANGRU MENG was born in Lantian, China, in 1963. He received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, China, in 1985, 1988, and 1994, respectively. From 1995 to 1997, he was a Visiting Scholar at the University of Electronic Science and Technology, Chengdu, China. He is currently a Professor with Air Force Engineering University, Xi'an, China. His research interests include next generation Internet, network virtualization, and survivable networks.

ZHENHUA YU received the B.S. and M.S. degrees from Xidian University, Xi'an, China, in 1999 and 2003, respectively, and the Ph.D. degree from Xi'an Jiao Tong University, Xi'an, in 2006. He is currently a Professor with the College of Computer Science and Technology, Institute of Systems Security and Control, Xi'an University of Science and Technology, Xi'an. He has authored more than 20 technical papers in conferences and journals, and holds two invention patents. His research interests mainly focus on cyber-physical systems, system security, and social networks.



QIAOYAN KANG was born in Yongchun, China, in 1980. She received the B.S., M.S., and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 2001, 2004, and 2008, respectively. She is currently an Assistant Professor with Air Force Engineering University. Her research interests include network management and survivable networks.

YU ZHAO received the B.S., M.S., and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 2009, 2011, and 2015, respectively. He is currently a Visiting Scholar with Air Force Engineering University. His research interests include cyber-physical systems and system security.