WILEY | Hindawi

Research Article

Towards Supporting Security and Privacy for Social IoT Applications: A Network Virtualization Perspective

Jian Sun [iD],[1] Guanhua Huang,[1] Arun Kumar Sangaiah,[2] Guangyang Zhu,[1] and Xiaojiang Du[3]

[1]*Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), UESTC, Chengdu, China*
[2]*Vellore Institute of Technology, Tamil Nadu, India*
[3]*Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA*

Correspondence should be addressed to Jian Sun; sj@uestc.edu.cn

Network function virtualization (NFV) is a new way to provide services to users in a network. Different from dedicated hardware that realizes the network functions for an IoT application, the network function of an NFV network is executed on general servers, and in order to achieve complete network functions, service function chaining (SFC) chains virtual network functions to work together to support an IoT application. In this paper, we focus on a main challenge in this domain, i.e., resource efficient provisioning for social IoT application oriented SFC requests. We propose an online SFC deployment algorithm based on the layered strategies of physical networks and an evaluation of physical network nodes, which can efficiently reduce bandwidth resource consumption (OSFCD-LSEM) and support the security and privacy of social IoT applications. The results of our simulation show that our proposed algorithm improves the bandwidth carrying rate, time efficiency, and acceptance rate by 50%, 60%, and 15%, respectively.

1. Introduction

In traditional service provider networks, network functions (NFs) (e.g., intrusion detection systems (IDSs), gateways, load balancers, network address translators (NATs), and firewalls [1, 2]) for social IoT applications [3–6] are implemented by specialized hardware devices, and it is expensive to incorporate some new devices into an existing service network. With an increasing number of social IoT applications, the demand of corresponding NFs for each user necessitates a significant amount of hardware resources. Furthermore, security and privacy preservation in social IoT applications are also important issues that need to be solved [7–10]. To address these problems, network function virtualization (NFV) technology has been put forward. In NFV networks, NFs are implemented in the form of software instead of dedicated hardware and separately run on different virtual machines (VMs) [11, 12] and thus guarantee the security and privacy requirements. As shown in Figure 1, NFs that run in NFV networks are called virtual network functions (VNFs).

Different VNFs are arranged in a prescribed order to form a service function chain (SFC) to fulfill the communication requirements [13–15].

NFV is convenient for social IoT service providers to deploy VNFs on commercial servers and manage the underlying network [16–18]. Using NFV technology, the social IoT service provider can flexibly deploy virtual NFs on commercial servers [19–21], which are traditionally fulfilled on dedicated hardware. Furthermore, with an increasing social IoT service demand [22–25], common commercial servers can provide support to multiple VNFs, which significantly reduces the vacancy rate of physical resources [26–28] and saves the cost of purchasing new dedicated hardware. The network operators can flexibly deploy and chain the VNFs according to the social IoT users' requests and practical network topology. This approach decreases the use of inapplicable functions and the corresponding deployment costs. Moreover, this approach can customize the social IoT services for the clients, introduce a broad business outlook, and promote its commercialization. Regarding
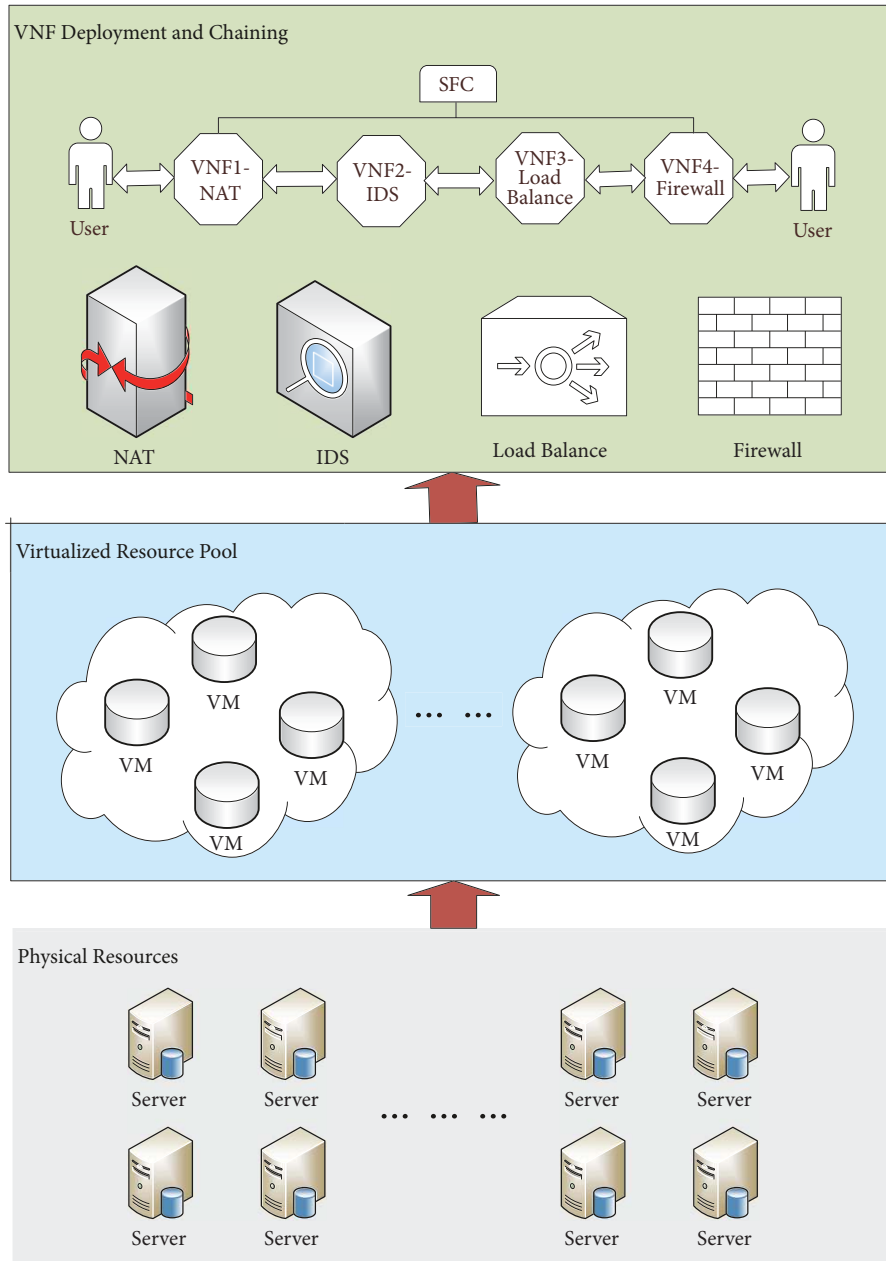
FIGURE 1: SFC organization by virtualizing commercial servers.

traditional NFs, VNFs increase policy compliance capabilities [29] and promote the emergence of more novel techniques and the development of NFV for social IoT application.

NFV can bring additional benefits to social IoT service providers, i.e., optimize their operational expenditure (OPEX) and capital expenditure (CAPEX) [11, 30–32] and improve the quality of service (QoS) by decreasing latency, reducing bandwidth consumption and increasing adaptation. NFV introduces many benefits to both users and social IoT service providers, but there are still many research limitations that need to be dealt with. For example, the bandwidth consumption of the SFC is an important research topic.

Bandwidth consumption represents the resource cost of communication among users/clients [33, 34]. The network operator seeks to achieve low bandwidth consumption to complete the deployment of the current SFC request and save resources for the next SFC request to improve the accommodating capacity of the network [35]. With an increasing diversity of service requirements and additional pressure from massive information transmission, bandwidth resources have become progressively scarce [36]. Furthermore, low bandwidth consumption can provide the client/user a good experience due to the excellent performance of the network operator. In previous research, several algorithms have aimed to reduce bandwidth consumption. In [17], the authors

focused on bandwidth consumption optimization and provided a framework for their studied problem. Thus, instead of the existing hardware environment, targeted research of NFV can produce increased benefits and reduce the energy and space consumption of various middle boxes [19, 37].

In this paper, we study the problem of how to provision social IoT oriented SFC requests while minimizing the bandwidth resource consumption as well supporting the security and privacy requirements. The studied problem has been proven to be an NP-hard problem [35, 37, 38].

Thus, we propose an efficient algorithm, called OSFCD-LSEM, with layered strategies for physical networks. It not only efficiently solves the SFC deployment problem but also makes the physical network more robust. Our OSFCD-LSEM algorithm evaluates the nodes in the physical network to optimize the bandwidth consumption and thus achieves a higher acceptance ratio and shorter response time (i.e., latency) for user demands than the exiting work. In addition, we can make the physical network more robust using our OSFCD-LSEM algorithm. The main contributions of this paper are as follows:

(i) We develop a model to evaluate the physical network and extend it precisely to its "*weak points*" and make the physical network more robust.

(ii) We propose an efficient algorithm (i.e., OSFCD-LSEM) for social IoT oriented SFC deployment that is based on the layered strategies of a physical network and evaluation of a physical network node to optimize the consumption of bandwidth resources and guarantee the security and privacy.

(iii) We conduct extensive simulations to verify and evaluate our algorithm in this paper. The results show that our proposed algorithm can efficiently address the online SFC deployment problem.

The remainder of this paper is organized as follows. Section 2 provides an overview of related works. Section 3 provides the problem descriptions and formulation. In Section 4, we describe our SFC deployment algorithm based on the layered strategies of physical networks and evaluation of physical network nodes. Section 5 shows the simulation results and analysis. Section 6 summarizes this paper.

2. Related Work

NFV aims to satisfy client demands with minimal resource consumptions (e.g., bandwidth consumption and computing core consumption) and high performance (e.g., low latency and high throughput) [20]. An increasing number of studies have focused on the more efficient deployment of social IoT oriented SFC requests in various scenarios.

The research in [39] focused on the suitability of different architectures of data center for a resilient SFC and the placement of VNFs with high availability constraints. In [40], the joint VNF placement and path selection problem were studied. The authors proposed a chain deployment

algorithm to balance the chain length and reuse factor; its target was to serve as many demands as possible under limited resources. The research in [41] designed a heuristic algorithm to address the NFV-RA problem in a coordinated manner by splitting the SFC request. In [42], the author made a BCMP mixed queuing network to replace the SFC model and proposed the convex optimization problem to shorten the acceptance interval of the service chain. The simulation results shown the algorithm in [42] can effectively adapt the resource usage to the network dynamics and serve more demands than other existing algorithms. The authors of [43] studied SFC deployment in a multi-domain network and proposed a vertex-centric distributed computing algorithm to find all feasible SFC deployment methods of user requests in the distributed orchestration framework; then, the algorithm selects the most appropriate scheme to deploy the SFC to achieve efficient performance. In [44], the researcher discussed the phenomenon that the VNFs may sprawl across the network due to inefficient mapping during the SFC orchestration process and designed an efficient greedy heuristic algorithm to solve the problem. The author in [45] studied the service availability constraints in a data center and designed a heuristic algorithm to deploy the SFC. In the situation of distributed NFVs, the authors in [46] studied the security level of a network security framework. The authors proposed a new optimization algorithm to avoid a single point of failure in a bottleneck problem and obtained reasonable performance relative to that of a common device. Li et al. [47] presented a real-time resource-distributing system for NFV, which integrates timing analysis with other algorithms, such as timing abstraction, SFC consolidation, and linear programming algorithm, to efficiently distribute network resources while considering latency constraints.

Many researchers have studied the deployment of the social IoT oriented VNF or SFC, and a few of them have attached importance to the total bandwidth consumption. Ye et al. [17] focused on the joint topology design and SFC deployment problem to minimize the total consumption of bandwidth resources. Their main idea was to reduce bandwidth usage and cost by prioritizing VNF deployment. In [48], the authors proposed and analyzed two deployment strategies, HSD and VSD, whose performance was analyzed in terms of cost. They claimed that HSD had better performance in terms of an even distribution of the load over all servers, access links, and ToR switches. The authors in [49] combined NFV with cloud computing and designed a bandwidth-guaranteed SFC-placing algorithm; then, the authors tested the performance of their algorithm in a data center. The simulation results showed that the heuristic algorithm had excellent performance. In [38], the authors designed a forecast-assisted online SFC deployment algorithm that included the prediction of future VNF requirements. The simulation results showed that the algorithm in [38] could reduce the blocking probability of SFC requests and effectively improve the profit of the service provider from SFC deployment.

Overall, bandwidth is the most prominently expensive resource that directly affects the number of demands that the
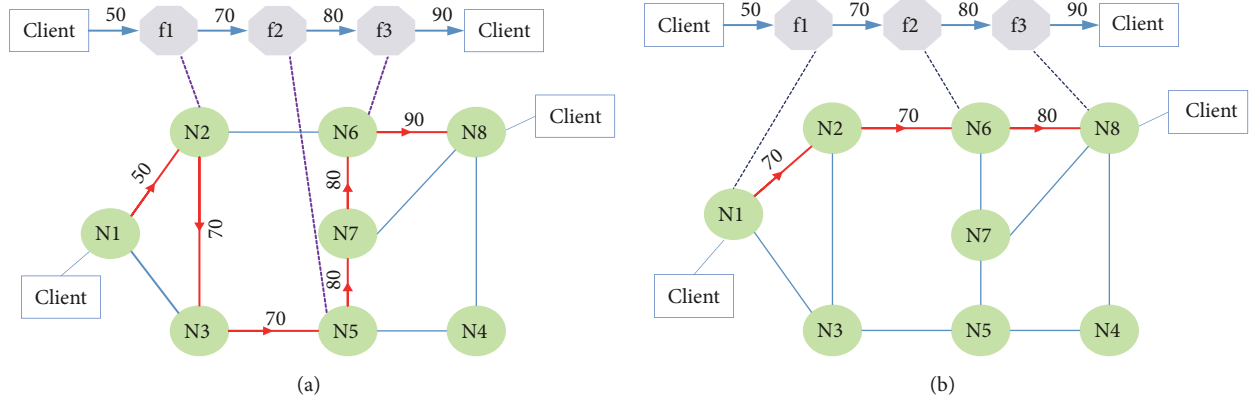
FIGURE 2: Two examples of SFC deployment.

network can satisfy online and the capacity of the physical network. Thus, in this research, we focused on the bandwidth consumption and designed an efficient algorithm to provision social IoT oriented SFC requests.

3. Problem Formulation

In this paper, we studied the online social IoT oriented SFC deployment problem. We considered a situation in which each SFC request has a pair of start and terminal nodes attached to the given physical network nodes and a specific sequence of VNFs that make up continuous functions. We have to deploy these VNFs onto the corresponding physical nodes and then organize the VNFs to form an SFC. To reduce the bandwidth resource consumption, we need to use fewer nodes to shorten the length of the SFC as much as possible.

A user request can be denoted as $S = (F_S, E_S)$, where $F_S = \{f_1, f_2, \ldots, f_m\}$ is the set of VNFs and $E_S = \{e_1, e_2, \ldots, e_q\}$ represents the virtual link after SFC deployment. The real physical underlying network can be modeled as $G = (N, L)$, where G is an undirected weighted graph, $N = \{N_1, N_2, \ldots, N_y\}$ is the set of physical nodes, and $L = \{L_1, L_2, \ldots, L_k\}$ is the set of real links in the physical underlying network. We use P to denote the physical path that holds the SFC request. We define C_B^T to denote the total bandwidth consumption, which is defined in

$$C_B^T = \sum_{L_{i\_e_j} \in P\_\{S\}} C_B^{L_{i\_e_j}} \tag{1}$$

where $L_{i\_e_j}$ denotes that virtual link, e_j is deployed on the link L_i, and $C_B^{L_{i\_e_j}}$ is the bandwidth consumption of virtual link e_j, which is deployed on the link L_i. We define $R_C^{N_i}$ as the available computing resources of the physical node N_i and $C_N^{N_i\_fj}$ as the computing resource requirements of VNF f_j, which would be deployed on the node N_i. $R_B^{L_i}$ is the available bandwidth resources of the physical link L_i.

To deploy an SFC, we must map the VNFs to some node and the virtual links E_S to some real links in the physical network; the path $P\_\{S\}$, which would hold the SFCs, must have sufficient nodes and computing resources to deploy the corresponding VNFs, and the physical links must have sufficient available bandwidth for communication among those VNFs. In addition, the available bandwidth resources must satisfy the requirements of the E_S in the corresponding user request. Similar to [10, 20], this paper assumes that each VNF from the same SFC needs to use different physical network nodes. Then, the number of VNFs (denoted as ‖Fs‖) should be less than the number of nodes of physical path $P\_\{S\}$ (denoted by ‖$P\_\{S\}$‖). Then, the problem of SFC deployment can be formulated as follows:

$$\min \sum_{L_{i\_e_j} \in P\_\{S\}} C_B^{L_{i\_e_j}}$$

$$s.t. \quad \forall L_{i\_e_j} \in P\_\{S\}$$

$$R_B^{L_i} - C_B^{L_{i\_e_j}} \geq 0 \tag{2}$$

$$\forall N_{i\_f_j} \in P\_\{S\}$$

$$R_C^{N_i} - C_N^{N_i\_f_j} \geq 0$$

$$\|P\_\{S\}\| - \|Fs\| \geq 0$$

Formula (2) is used to model the SFC deployment problem, which can minimize the bandwidth consumptions while finishing SFC deployment. There must be sufficient available computing resources to deploy corresponding SFCs, and the bandwidth must be sufficient to satisfy the communication demands among the corresponding VNFs. In addition, there must be sufficient nodes in the physical path P to deploy the VNFs of the SFC request.

Figure 2 shows two different examples of provisioning an SFC request, both of which successfully deploy the SFC request and satisfy the clients' demands. Nodes N1 and N8 are the requested client node and destination client node, respectively. There are three VNFs (i.e., $f_1, f_2,$ and f_3) in the SFC request. Client node N1 needs to transmit 50 units

of traffic to VNF f_1. Between VNFs f_1 and f_2, 70 units of information need to be transmitted. Between VNFs f_2 and f_3, 80 units of traffic need to be transmitted. There is a 90-unit transmission demand from VNF f_3 to the destination client node $N8$. As shown in Figure 2(a), we deploy the VNFs f_1, f_2, and f_3 onto the physical nodes $N2$, $N5$ and $N6$, respectively. Then, we find the path $P = \{N1\text{-}N2, N2\text{-}N3, N3\text{-}N5, N5\text{-}N7, N7\text{-}N6, N6\text{-}N8\}$ to host the SFC. In this deployment scheme, the total bandwidth consumption is 440 units. Although this approach successfully deploys the SFC request, it wastes bandwidth resources. In Figure 2(b), the VNFs f_1, f_2, and f_3 are deployed onto the physical nodes $N1$, $N6$, and $N8$, respectively. Then, we find a short path $P = \{N1\text{-}N2, N2\text{-}N6, N6\text{-}N8\}$ (shown as the red line in Figure 2(b)) which can also deploy all VNFs from the user requests. However, the total consumption of bandwidth resources of this scheme is only 220 units, which is approximately half of the consumption of the deployment in Figure 2(a). In Figure 2(b), path P is the shortest path to deploy this SFC request, and this scheme can reduce more bandwidth consumption. Mapping the SFC to the path in Figure 2(b), the service network can deploy more SFC with other links, which is impossible in Figure 2(a).

Therefore, different deployment schemes significantly affect the bandwidth consumptions and thus affect the available network capacity. An efficient algorithm is important and urgently needed to better deploy SFC requests and reduce bandwidth consumption.

4. Algorithm Design

Since the optimal SFC deployment problem is NP-hard [35, 37], we design an efficient algorithm for solving our research problem in this section. Our proposed algorithm employs the layered strategies of physical networks and evaluation of physical network nodes to optimize the consumption of bandwidth resources, which is denoted by OSFCD-LSEM. The basic idea of OSFCD-LSEM is to find the shortest paths to deploy SFC requests while saving as many bandwidth resources as possible to satisfy more user requests. When a user demand arrives, the OSFCD-LSEM algorithm begins to handle it. First, calling Algorithm 2, the OSFCD-LSEM algorithm layers the physical underlying network and obtains the information from the nodes and links in each layer of the physical network. Then, it calls Algorithm 3 to perform an evaluation of nodes in the network and select some nodes that are most suitable to host the VNFs of this SFC request. Finally, it connects the deployed VNFs by using a shortest path to fulfill an SFC. By layered strategies and selection of the nodes for VNF deployment, the OSFCD-LSEM algorithm can deploy VNFs in the most suitable nodes and deploy the SFC requests in appropriate simple paths to save more bandwidth resources. The physical path must contain the start node N_r and the terminal node N_a. Furthermore, the nodes in the physical path must have sufficient resources to host the VNFs of the user request.

In the proposed OSFCD-LSEM algorithm, we provide some definitions of variables. G_L is the physical network after layering, and V_X is used to denote the set of nodes in the X-th layer. The X-th layer is denoted as $L.X$. G_L^X is the inner layered network in the X-th layer ($L.X$), and we used $L.Y$ to denote the Y-th layer in G_L^X. $V_{(X,Y)}^i$ is the set of nodes in the $L.Y$ of G_L^X, which includes the node N_i. E_X is the set of links that connect the nodes from $L.X$ and $L.X$-1. $E_{(X,Y)}^i$ denotes the corresponding links that connect the nodes in the $L.Y$-1 about node N_i, and L_{MAX}^i is the corresponding maximal layer of node N_i. L_{MAX} is the layer number of G_L. N_T is the node number of the physical network, and L_T is the link number of the physical network G. The OSFCD-LSEM algorithm is shown in Algorithm 1.

Next, Algorithm 2 is responsible for handling hierarchical physical networks in our algorithm. Algorithm 2 layers the entire network to obtain the layering information of the nodes and links in the network and outputs the results to other algorithms. Therefore, Algorithm 2 is the basis of our SFC request deployment scheme and physical network node evaluation. The physical network layering can be formulated as follows:

$$G_L = \sum_{X=1}^{L_{MAX}} (V_X, E_X) + \sum_{X=2}^{L_{MAX}} G_L^X \tag{3}$$

$$G_L^X = \sum_{V_i \in V_X} \sum_{Y=1}^{L_{MAX}^i} \left(V_{(X,Y)}^i, E_{(X,Y)}^i \right) \tag{4}$$

$$\sum_{X=1}^{L_{MAX}} V_X - N_T \geq 0 \tag{5}$$

$$\sum_{X=2}^{L_{MAX}} E_X + \sum_{X=2}^{L_{MAX}} \sum_{V_i \in V_X} \sum_{Y=2}^{L_{MAX}^i} E_{(X,Y)}^i - L_T = 0 \tag{6}$$

In (3), two parts make up G_L: one part is the inner layer network G_L^X about the X-th layer ($L.X$) and the other part is the overall layered network. The layering process begins from the request node N_r; thus, $V_1 = N_r$, $E_1 = \emptyset$, and $G_L^1 = \emptyset$. Equation (4) shows that, to make G_L closer to the physical network G, each layer except layer $L.1$ should obtain its inner layer node-related information. The OSFCD-LSEM algorithm can obtain a more precise evaluation of the physical network and more efficiently deploy the corresponding SFC requests. In (5), the equation describes that all nodes in the layered physical network must be in the corresponding layer. In (6), the equation shows that each link should be in the corresponding layer or inner layer.

In Figure 3, we give an example of layering a network. Figure 3(a) shows the topology of a physical network, and it has a total of 10 nodes. Figure 3(b) provides the detailed processing procedure for layering the network topology. We assume that the start node N_r in the request is node $N1$ and that the terminal node N_a in the request is node $N9$. First, our algorithm places the start node $N1$ into $L.1$ ($N1$ is the only node in V_1 of $L.1$) and places nodes $N2$, $N3$, and $N4$ into L.2 because they directly have links to node $N1$. Then, our algorithm places nodes $N5$, $N6$, and $N9$, all of which have links to some nodes in the L.2 (i.e., nodes $N2$, $N3$, and $N4$),

```
Input: (1) SFC request. (2) physical network G;
Output: SFC deployment scheme.
(1) Receive a user request;
(2) Initialize Path = [];
(3)  N_a ⟶ Path; N_L = N_a;
(4) Layer the topology: Algorithm 2(N_r; N_L; G);
(5) Get L_A: the layers that destination client belongs to;
(6) while L_S > max(L_A) + ∑_{X=1}^{L_MAX} max(L^i_MAX) ∀N_i ∈ V_X do
(7)    if max{L_A} = L_MAX
(8)          N_TEMP = Algorithm 3(; ; true; max{L_A});
(9)          N_TEMP ⟶ Path;
(10)         N_L = N_TEMP;
(11)   else
(12)         N_TEMP = Algorithm 3(; ; false; max{L_A});
(13)         N_TEMP ⟶ Path;
(14)         N_L= N_TEMP;
(15)   end if
(16)   L_S = L_S - 1;
(17)   VNF ⟶ N_TEMP;
(18)   Algorithm 2(N_r; N_L; G);
(19)   Update L_A;
(20) end while
(21) if L_S ≤ max(L_A)
(22)       Select min L.X ∈ L_A && L.X > L_S;
(23)       while N_r ∉ Path do
(24)             N_TEMP = Algorithm 3 (; ; true; L.X);
(25)             N_TEMP ⟶ Path;
(26)             N_L = N_TEMP;
(27)             L.X = L.X - 1;
(28)             VNF⟶ N_TEMP;
(29)       end while
(30) end if
(31) SFC deployment scheme is Path.
```

ALGORITHM 1: OSFCD-LSEM algorithm.

into $L.3$ (we specify that all nodes can only belong to one layer except the terminal node $N9$; thus, node $N4$ cannot be part of $L.3$, even though it connects with node $N3$, which is in $L.2$).

In our layered network, except for the destination client node $N9$, the nodes in one layer must have links to the nodes in the previous layer. Thus, nodes $N7$, $N8$, and $N10$ directly have links to the nodes in $L.3$ (i.e., nodes $N5$, $N6$, and $N9$), whereas $N10$ connects only with destination client node $N9$ among the three nodes in $L.3$. Node $N10$ should not be placed in layer $L.4$. Thus, we place only nodes $N7$ and $N8$ into $L.4$. Nodes $N9$ and $N10$ connect with nodes $N7$ and $N8$ in $L.4$; thus, we place nodes $N9$ and $N10$ in layer $L.5$, and because $N9$ connects with node $N10$, our algorithm places node $N9$ in $L.6$. When ten nodes in G belong to the corresponding layers, the overall network-layered processing finishes. Thus, the OSFCD-LSEM algorithm can guarantee finding a path without loops. For each $L.X$, we also need to layer and obtain the inner layer G_L^X. In the example shown by Figure 3(b), the layer $L.2$ has an inner layer G_L^2 that includes two layers. For G_L^2 in Figure 3(b), each layer $X ≤ L_{MAX}$ and each node $N_i ∈ V.X$ should be set as the start node N_r; let $N_a = ∅$. Then, we

obtain their inner layer information. In G_L^2, both L_{MAX}^{N3} and L_{MAX}^{N4} are equal to 2, whereas L_{MAX}^{N2} is equal to 1.

Overall, the physical network is layered into six layers with two inner layer networks associated with nodes $N3$ and $N4$. The start node N_r is the only one in the first layer, and the terminal node N_a is in three layers, including $L.3$, $L.5$, and $L.6$. Thus, we obtain three paths that can be used between N_r and N_a. L_P is used to denote the length of path, which is equal to the number of VNFs that the path can hold. L_S is the length of an SFC, which is equal to the VNF number of the SFC request.

Algorithm 3 evaluates the nodes in the physical network and selects a node which is most suitable to deploy the required VNF. After obtaining the layering information, Algorithm 3 makes the decision regarding whether the multiple links can satisfy the user request. When the maximal layer number in L_A of the terminal nodes N_a, which is obtained from the sum of all inner layers, is smaller than the SFC length L_S in the user request, the physical network cannot satisfy the user request. For example, if we need to deploy an SFC into the physical network in Figure 3(a), the start and terminal nodes are $N1$ and $N9$. The maximum layer number in L_A is

Input: (1) N_r; (2) N_a; (3) physical network G.
Output: G_L;
(1) $N_r \longrightarrow V_1$; $L_{MAX} = L.1$;
(2) **for** $V_{L_{MAX}} \neq \emptyset$; $N_m \neq N_a$; **do**
(3)   **for** each $N_n \in G$; **do**
(4)   **if** $N_m \longleftrightarrow N_n$ && $N_n \notin \sum_1^{L_{MAX}} V_X$
(5) $N_n \longrightarrow V_{L_{MAX+1}}$;
(6)   **else if** $N_m \longleftrightarrow N_n$ && $N_n \in \sum_1^{L_{MAX}} V_X$ && $N_n = N_a$
(7) $N_n \longrightarrow V_{L_{MAX+1}}$;
(8)   **end if**
(9)   **end for**
(10) L_{MAX} ++;
(11) **end for**
(12) **for** $L.X \leq L_{MAX}$; **do**
(13)   **for** $N_m \in V_X$; **do**
(14) $N_m \longrightarrow V_{(X,1)}$;
(15) $L^m_{MAX} = L^m.1$;
(16)   **for** $V_{(X,L^m_{MAX})} \neq \emptyset$; **do**
(17)   **if** $N_n \in V_X$ && $N_m \longleftrightarrow N_n$ && $N_n \notin \sum_1^{L^m_{MAX}} G_L^X$
(18) $N_n \longrightarrow V_{(X,L^m_{MAX})}$;
(19)   **end if**
(20) L^m_{MAX} ++;
(21)   **end for**
(22)   **end for**
(23) **end for**
(24) **return** G_L

ALGORITHM 2: Network layered processing.

Input: (1) SFC request;
        (2) G_L;
        (3) **bool** *direction*;
        (4) **X:** $N_L \in V_X$;
Output: the node N_C which has the minimum value of δ;
(1) Temp = $+\infty$
(2) int i= 0;
(3) **if** *direction* is true
(4)   i= X-1;
(**5**) **end if**
(6) **if** *direction* is false
(7)   i= X+1;
(8) **end if**
(9) **for** $N_m \in V_i$ **do**
(10)   **if** $N_m \longleftrightarrow N_L$
(11)   **if** $B_{si\_m} > B_{ri\_m}$ && $B_{se\_m} > B_{re\_m}$ && $C_{s\_m} > C_r$;
(12)   Compute δ according to Equation (7);
(13)   **if** $\delta <$ Temp
(14)   Temp = δ;
(15) $N_C = N_m$;
(16)   **end if**
(17)   **end if**
(18)   **end if**
(19) **end for**
(20) **return** N_C.

ALGORITHM 3: Evaluation of related nodes.

(a) Physical network

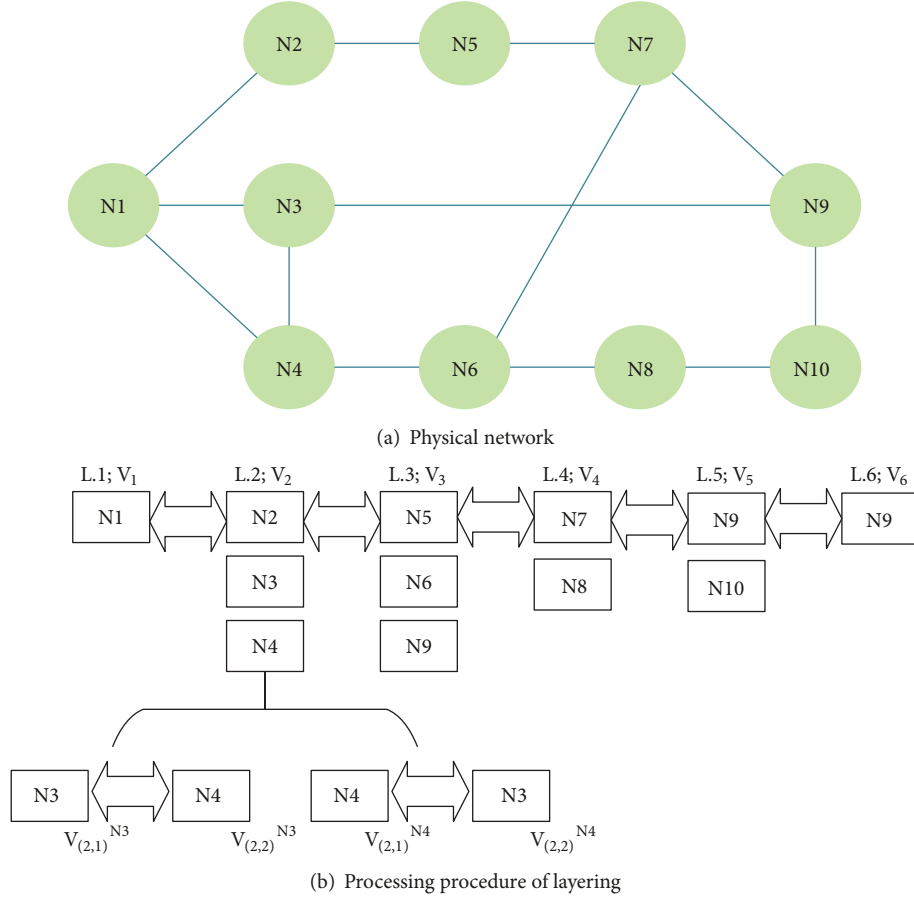

(b) Processing procedure of layering

FIGURE 3: An example of a layered physical network.

6, and there is an inner layer in layer $L.2$. The total sum is 7, so this network topology can satisfy only an SFC request whose length is no more than 7. An SFC request whose number of VNFs is more than 7 is too large for this network. However, as long as the number of VNFs is not greater than the number of nodes in the network, our OSFCD-LSEM algorithm can try to find more paths to deploy the longer SFC. It may need additional time and capacity of bandwidth resources because there are too many VNFs that need to be deployed. The proposed Algorithm 3 can search nodes in both positive and negative directions. When addressing an SFC that is longer than the abovementioned sum, Algorithm 3 commonly finds a suitable node in the layer V_{L_A+1} instead of in the upper layer V_{L_A-1}; then, the algorithm can increase the length of the path (L_P). However, an extreme situation should be considered, such as when the node is in the last layer $L.L_{MAX}$. For this situation, our algorithm will find a node in the upper layer $V_{L_{MAX}-1}$ and run Algorithm 2 again.

Finally, we must evaluate the nodes from each layer of the layered network and select some nodes to map the VNFs. Algorithm 3 uses the abovementioned strategy to find a path from N_a to N_r to host an SFC request and satisfy the user demand. Algorithm 3 selects the nodes from each layer using (7) and (8). The selected node must directly link to the node in the next layer V_N and must have sufficient resources

to deploy the VNFs and satisfy the corresponding function requirements.

$$\delta = \min\left(\frac{B_{si\_m} - B_{ri\_vnf_{i,j}}}{B_{si\_m}}, \frac{B_{se\_m} - B_{re\_vnf_{i,j}}}{B_{se\_m}}\right)$$
$$\times \frac{C_{s\_m} - C_{r\_vnf_{i,j}}}{C_{s\_m}} \tag{7}$$

$$C_{s\_m} = C_{total\_m} - \sum_{S_i \in SFC\_online} \sum_{vnf_{i,j} \in S_i} C_{r\_vnf_{i,j}} \times \alpha_{vnf_{i,j}\_m} \tag{8}$$

In (7), we use δ to denote the node's appropriateness for the VNF in the user request. B_{si_m} is the idle bandwidth of all links that connect node N_m with the nodes in the next layer V_N, and $B_{ri\_vnf_{i,j}}$ is the requested bandwidth resource from $vnf_{i,j}$ to the $vnf_{i,j+1}$, where $vnf_{i,j}$ is the j-th VNF in the S_i (i.e., the i-th SFC request). B_{se_m} is the idle bandwidth of the path that connects node N_m with the nodes in the upper layer V_U, and $B_{re\_vnf_{i,j}}$ is the requested bandwidth resource for making $vnf_{i,j}$ connect with the last VNF. C_{s_m} represents the idle computing resources of node N_m, and $C_{r\_vnf_{i,j}}$ represents the requested computing resources of the $vnf_{i,j}$. In (8), C_{total_m} represents the total computing resources of node N_m, SFC_{online} is the set of online SFC requests, and
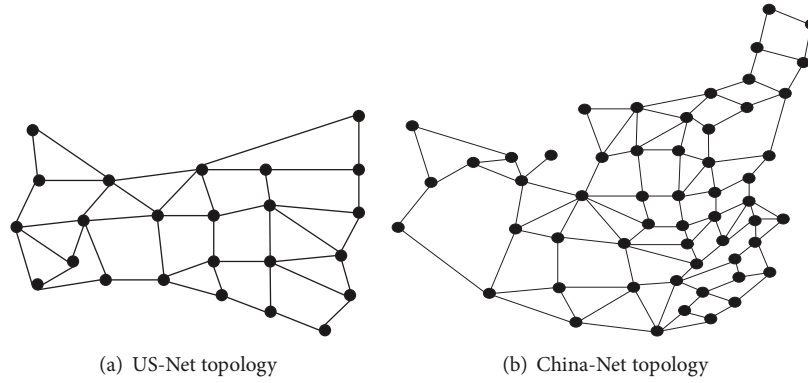
(a) US-Net topology   (b) China-Net topology

FIGURE 4: Two real network topologies used in our simulations.

$\alpha_{vnf_{i,j}\_m}$ denotes whether $vnf_{i,j}$ is deployed on node N_m. We can evaluate whether a physical node is suitable to deploy the corresponding VNF according to the value of δ. After physical node evaluation, we select the node with the minimal value of δ to hold the corresponding VNF.

5. Simulation Results and Analysis

To evaluate the performance of our algorithm, we compare our algorithm with two existing algorithms, i.e., closed-loop with critical mapping feedback (CCMF) [17] and key-VNF deploy first (KVDF) [38]. CCMF deploys the VNFs that have more resource requirement priorities to optimize the total consumption of bandwidth resources. KVDF focuses on the relation between different VNFs in the same SFC and the relation among different SFCs. According to the relation and influence, KVDF prioritizes the deployment of the key-VNF and key-SFC.

To evaluate the performance of algorithms in different network scenarios, we evaluate the performance of compared algorithms in moderate-scale and large-scale networks, the US-Network [50] (shown in Figure 4(a)), and the China-Network [51] (shown in Figure 4(b)). We use GT-ITM [52] to generate the moderate-scale and large-scale network topologies. In moderate-scale network topologies, there are approximately 100 nodes and 400 links. In large-scale network topologies, there are approximately 300 nodes and 1000 links. The US-Net topology has 24 nodes and 43 links. Additionally, the China-Net topology has 55 nodes and 103 links. In these network topologies, the bandwidth resource of all links is uniformly distributed at 100~200 units and the computing resource of all nodes is set as 10 units.

In our simulations, we set the following parameters according to the existing work [49]. The computer randomly generates user requests with lengths (i.e., Ls) from 5 to 14, and these online SFC requests arrive as a Poisson process. For each physical network and for a given L_s, we randomly generate 10,000 SFC requests with the request client and destination client nodes, which are randomly assigned to the physical nodes. The computing resource demand of each VNF follows a uniform distribution U (1, 3), and the bandwidth resource demand of each virtual link follows a uniform distribution U (20, 50).

With the increasing number of users and SFC requests, the deployment of SFC in a static network becomes increasingly challenging. Thus, the network scalability must be improved. For a network-aware scaling strategy, it is better to extend the network instead of changing the network. In network G, we define the evaluation of information G^S in

$$G^S = \sum_{X=1}^{L_{MAX}} \sum_{N_i \in V_X} (C_s + B_{si} + B_{se}) \tag{9}$$

The OSFCD-LSEM algorithm layers the physical network, obtains the layer which has minimum resources, analyzes its inner layer information, and obtains the "*weak*" *nodes* or *links* that influence the capacity of the network. Then, the OSFCD-LSEM algorithm extends their resources to secure a more robust network.

We obtained the simulation results by averaging the results of multiple simulations. We used an Ubuntu virtual machine running on a computer with a 3.7 GHz Intel Core i3-4170 and 4 GB of RAM to run the simulations. And the algorithms were coded in Java programming language.

From Figure 5, we can see the simulation results of our OSFCD-LSEM algorithm in a moderate-scale network. Figure 5(a) shows the information of the physical network. We can see that $L.8$ limits the network capacity and will affect the deployment of SFC. Figure 5(b) shows the information of the physical nodes in $L.8$. In $L.8$, node-67 has the minimal amount of bandwidth, and node-72 has the minimal amount of computing resources. Both node-67 and node-72 are the "*weak points*" in the physical network. If we can increase their corresponding resources, the capacity of the physical network will be enhanced. For network operators, increasing the corresponding resources to the corresponding nodes and links is necessary and highly beneficial; moreover, they can also obtain a more robust network.

Figure 6 shows the simulation results on the US-Net network. Figure 6(a) shows the information about each layer in the overall network, and we find that $L.4$ may be the "*weak point*" of the physical network because of the lack of computing and bandwidth resources. Figure 6(b) shows
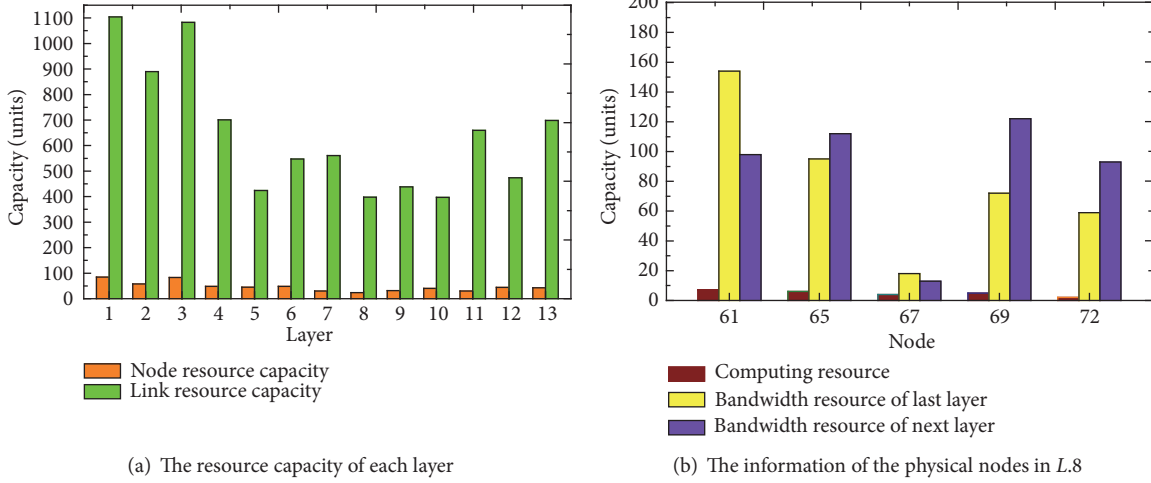
(a) The resource capacity of each layer

(b) The information of the physical nodes in $L.8$

FIGURE 5: Simulation results for the moderate-scale network.

(a) The resource capacity of each layer

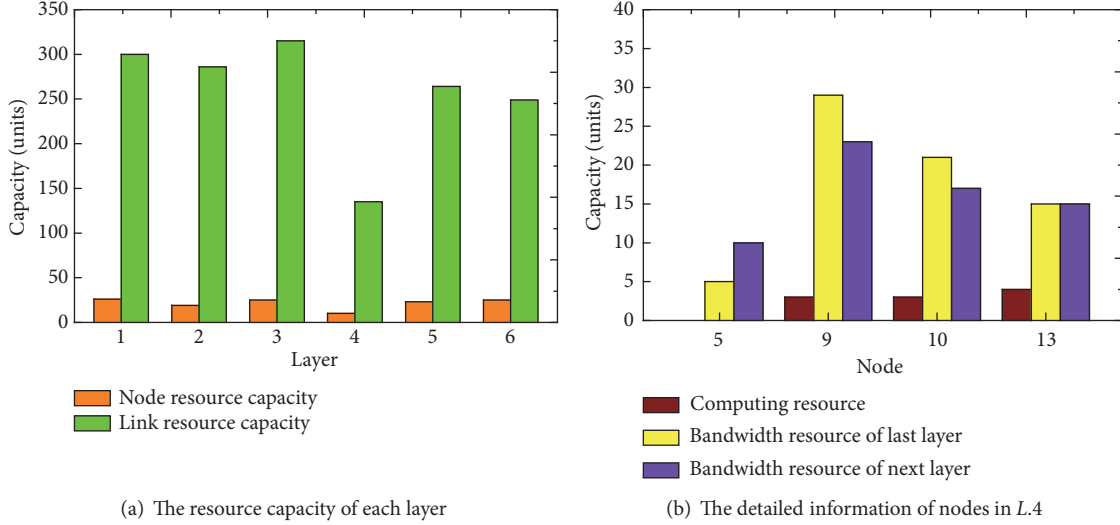(b) The detailed information of nodes in $L.4$

FIGURE 6: Simulation results for the US-Net network.

the nodes' detailed information about $L.4$. In the evaluation result, node 5 has no computing resources to carry any VNF. This means that node 5 is a "*dead point*" in the network, and it significantly compromises network performance. Moreover, there are a few bandwidth resources in node 5 for connecting the nodes in the last layer and the next layer. It is necessary and urgent to supplement the corresponding bandwidth resources to make the network more robust. With respect to node 13, though there are enough computing resources, there are insufficient bandwidth resources. Thus, increasing bandwidth resources in node 5 to connect the nodes in $L.3$ and $L.4$ will significantly improve the ability of the physical network to provision more SFC requests and make the network more robust.

For network operators and our OSFCD-LSEM algorithm, those "*weak points*" serve as the focal points for the extension of the physical network. Relative to an approach that blindly

extends the physical network to all nodes and links without focus and direction considerations, the extension implemented by the OSFCD-LSEM algorithm is more accurate and efficient. Our OSFCD-LSEM algorithm can be used to supply resources to the nodes that require the most resources, and it avoids wasting resources while improving the total physical network capacity.

Figure 7 shows the simulation results of the acceptance ratios of the OSFCD-LSEM algorithm and the other two algorithms. Figures 7(a), 7(b), 7(c), and 7(d) show the comparable results in the moderate-scale, large-scale, US-Net, and China-Net networks, respectively. The OSFCD-LSEM algorithm has a higher SFC request acceptance ratio than the CCMF and KVDF algorithms in all networks. As a result, the OSFCD-LSEM can find the appropriate nodes for VNF deployment based on the client's direction with the help of the network layering algorithm. Compared with the other

(a) Results for moderate-scale network

(b) Results for large-scale network

(c) Results for US-Net network

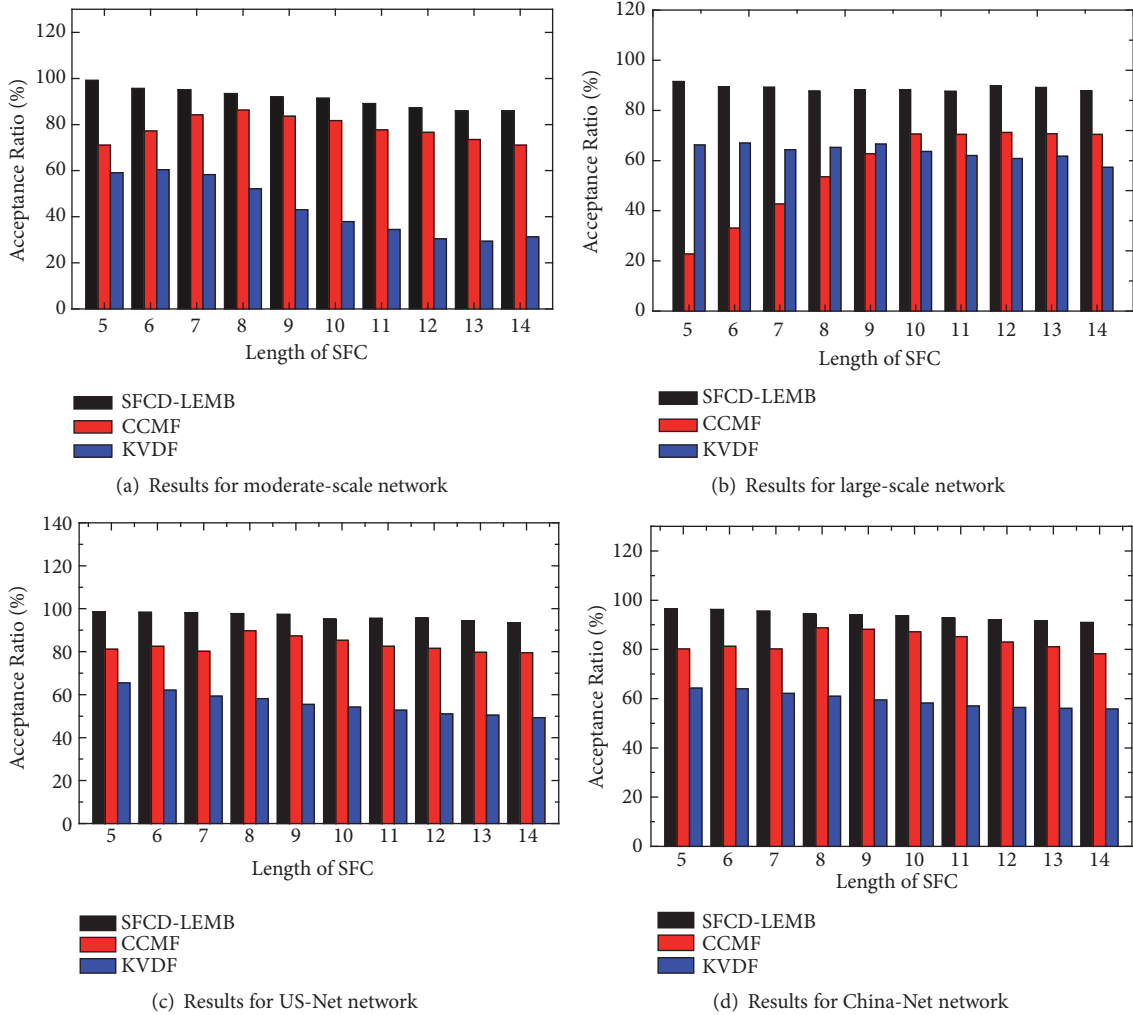(d) Results for China-Net network

Figure 7: Acceptance ratios in different physical networks.

two algorithms, OSFCD-LSEM always finds the shortest and most appropriate path to deploy SFCs; thus, it has the highest acceptance ratio among the algorithms. Furthermore, the OSFCD-LSEM algorithm has a relatively stable acceptance ratio in arbitrary scale networks and different L_s because our OSFCD-LSEM algorithm evaluates the network after layering part of the network and can appropriately finish the SFC deployments. In addition, the OSFCD-LSEM algorithm performs better in both networks that are generated by GT-ITM and in the real networks. The excellent performance is achieved because it is based on the evaluation of the layered network.

Figure 8 shows the simulation results of the running time of the OSFCD-LSEM algorithm and the other two algorithms. Figures 8(a), 8(b), 8(c), and 8(d) reveal the comparable results in the moderate-scale, large-scale, US-Net, and China-Net networks, respectively. Among the three algorithms, the running time of the OSFCD-LSEM algorithm for performing the deployment is shortest because it can find nodes for VNF deployment based on the appropriate search direction rather than on random searching. Therefore, the

OSFCD-LSEM can find the path towards the client node within fewer searching steps and shorter searching time compared with the other algorithms. Moreover, the optimization of the OSFCD-LSEM algorithm makes its running time slowly increase with the increasing length of SFC (L_s). As L_s increases, OSFCD-LSEM requires only a few searching steps, and the running time slowly increases due to the directional search advantage.

Figures 9(a), 9(b), 9(c), and 9(d) show the simulation results of the consumptions of bandwidth resources in the moderate-scale, large-scale, US-Net, and China-Net networks, respectively. Figure 9 shows that the OSFCD-LSEM algorithm can provision SFC requests with less bandwidth consumption than the other two algorithms for the same SFC requests. Since it employs the network layering strategy, the shortest paths can be found by the OSFCD-LSEM algorithm to deploy the SFC based on an efficient search direction. Then, compared with the other two algorithms, SFC communication via the shortest paths consumes less bandwidth resources. With the increase in L_s and network size, the OSFCD-LSEM algorithm shows outstanding

(a) Results for moderate-scale network

(b) Results for large-scale network

(c) Results for US-Net network

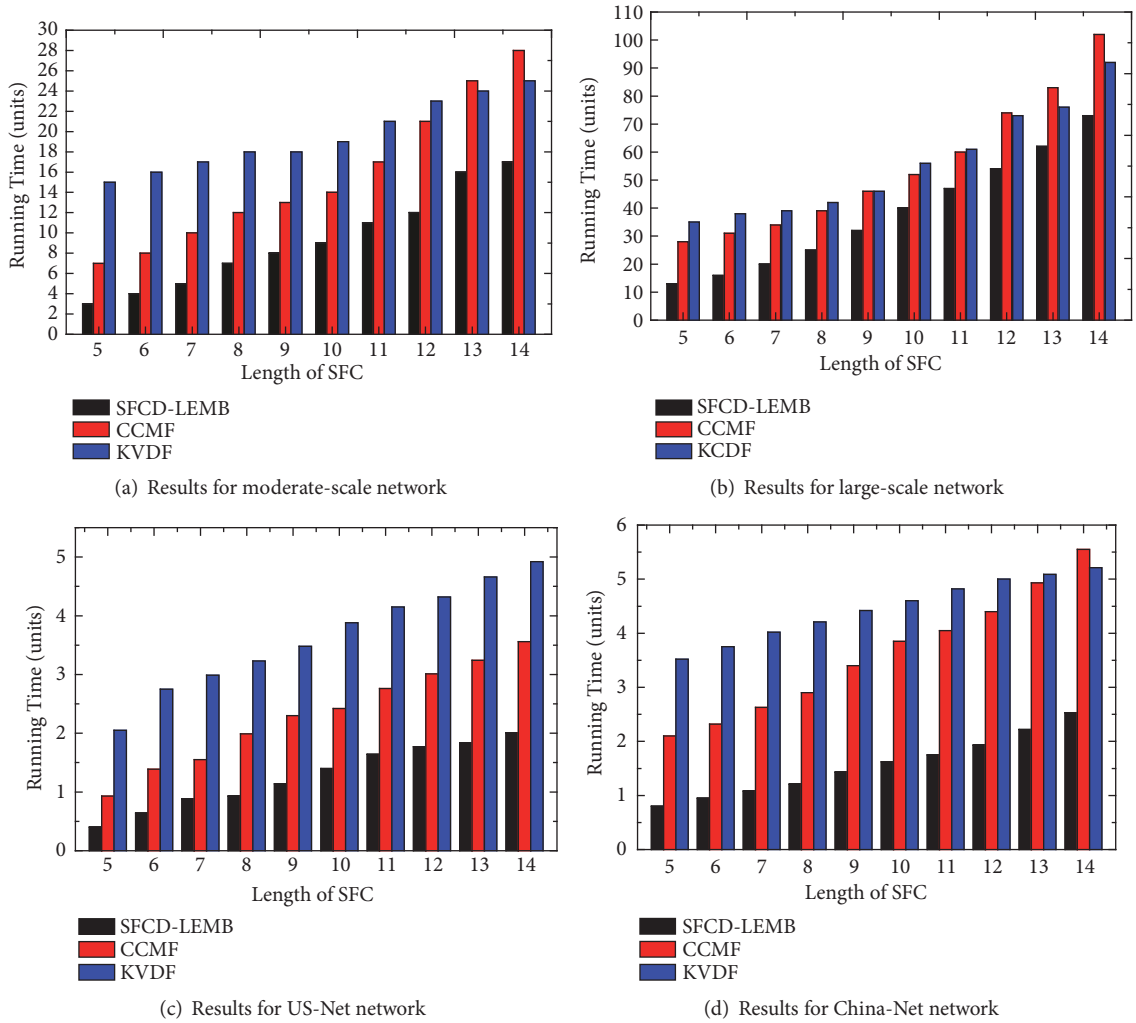(d) Results for China-Net network

FIGURE 8: Running time in different physical networks.

performance in saving bandwidth resources because it obtains the layering information of the nodes and links in the network by layered strategies, which is one of the main contributions of OSFCD-LSEM. Importantly, since saving bandwidth resources is the goal of this study, the simulations of different network scenarios and SFC lengths (L_s) show that the OSFCD-LSEM algorithm obtains the expected results.

In summary, the OSFCD-LSEM algorithm layers the physical network and is superior to the other two compared algorithms in terms of the acceptance ratio, running time, and bandwidth consumption. These achievements can be attributed to the manner in which the OSFCD-LSEM algorithm obtains overall evaluation information of the physical network.

6. Conclusion

Under NFV technology, the network functions can be migrated from dedicated hardware and be deployed onto commercial servers in any necessary location. NFV can

remarkably enhance the flexibility and reduce the resource consumption in the physical network. With the benefit of NFV, the clients' experience and network performance are both improved.

In this paper, we study the efficient online social IoT application oriented SFC provision problem. We propose an SFC deployment algorithm, OSFCD-LSEM, which layers the physical network to obtain the layering information of the nodes and links in the network. Moreover, it also selects the most suitable node to host the VNFs by evaluating some nodes in the physical network. The simulation results show that the OSFCD-LSEM algorithm has better performance in terms of time efficiency, acceptance ratio, and bandwidth consumption for provisioning social IoT application oriented SFC requests. Furthermore, to satisfy the increasing social IoT demands, we can use the layering information to appropriately extend the physical network.

In the future work, we can introduce artificial intelligence algorithm to the existing framework to improve the accuracy rate, or to study the algorithm to run efficiently in a more complex physical network environment.
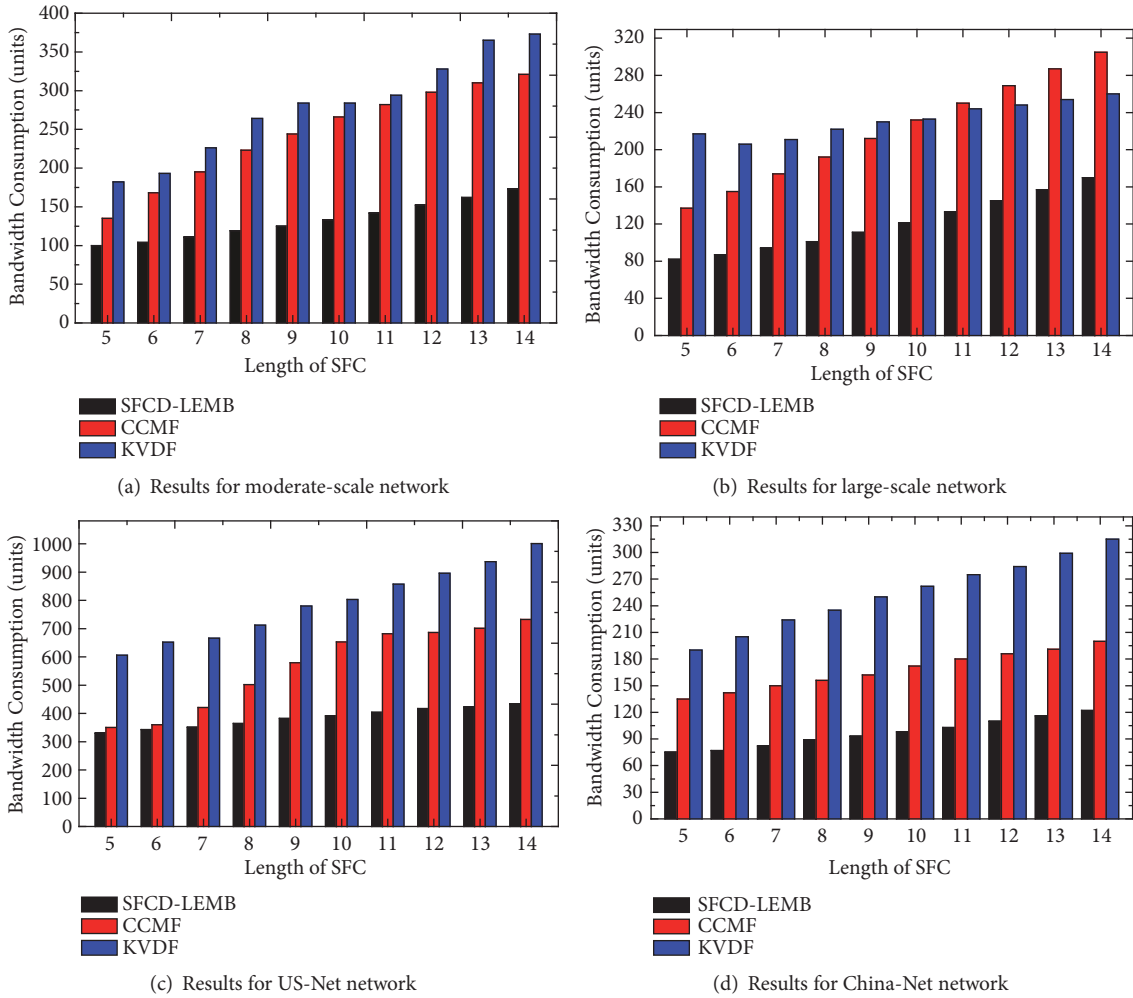
(a) Results for moderate-scale network

(b) Results for large-scale network

(c) Results for US-Net network

(d) Results for China-Net network

Figure 9: Bandwidth consumptions in different physical networks.

Data Availability

The data supporting the results reported in this work is generated in our simulation experiments in our study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] M. S. Yoon and A. E. Kamal, "NFV resource allocation using mixed queuing network model," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–7, 2016.

[2] G. Sun, S. Sun, J. Sun, H. Yu, X. Du, and M. Guizani, "Security and privacy preservation in fog-based crowd sensing on the internet of vehicles," *Journal of Network and Computer Applications*, vol. 134, pp. 89–99, 2019.

[3] G. Sun, V. Chang, M. Ramachandran et al., "Efficient location privacy algorithm for Internet of Things (IoT) services and applications," *Journal of Network and Computer Applications*, vol. 89, pp. 3–13, 2017.

[4] Q. Du, H. Song, and X. Zhu, "Social-feature enabled communications among devices toward the smart iot community," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 130–137, 2019.

[5] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12313–12325, 2018.

[6] G. Sun, D. Liao, H. Li, H. Yu, and V. Chang, "L2P2: A location-label based approach for privacy preserving in LBS," *Future Generation Computer Systems*, vol. 74, pp. 375–384, 2017.

[7] G. Sun, L. Song, D. Liao, H. Yu, and V. Chang, "Towards privacy preservation for "check-in" services in location-based social networks," *Information Sciences*, vol. 481, pp. 616–634, 2019.

[8] H. Song, G. A. Fink, and S. Jeschke, *Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications*, Wiley-IEEE Press, 2017.

[9] X. Huang, Y. Lu, D. Li, and M. Ma, "A novel mechanism for fast detection of transformed data leakage," *IEEE Access*, vol. 6, pp. 35926–35936, 2018.

[10] G. Sun, Y. Xie, D. Liao, H. Yu, and V. Chang, "User-defined privacy location-sharing system in mobile online social networks," *Journal of Network and Computer Applications*, vol. 86, pp. 34–45, 2017.

[11] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 1346–1354, 2015.

[12] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *Proceedings of the IEEE 4th International Conference on Cloud Networking*, pp. 171–177, 2015.

[13] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On Dynamic service function chain deployment and readjustment," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 543–553, 2017.

[14] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proceedings of the IEEE 3rd International Conference on Cloud Networking*, pp. 7–13, 2014.

[15] X. Wang, C. Wu, F. Le et al., "Online VNF scaling in datacenters," pp. 1-9, 2016.

[16] 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 Systems Journal*, pp. 1–13.

[17] Z. Ye, X. Cao, J. Wang, H. Yu, and C. Qiao, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Network*, vol. 30, no. 3, pp. 81–87, 2016.

[18] S. Clayman, E. Maini, A. Galis et al., "The dynamic placement of virtual network functions," *IEEE Network Operations and Management Symposiu*, pp. 1–9, 2014.

[19] G. Sun, V. Chang, G. Yang, and D. Liao, "The cost-efficient deployment of replica servers in virtual content distribution networks for data fusion," *Information Sciences*, vol. 432, pp. 495–515, 2018.

[20] P. S. Khodashenas, B. Blanco, M.-A. Kourtis et al., "Service mapping and orchestration over multi-tenant cloud-enabled RAN," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 904–919, 2017.

[21] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[22] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Communications Magazine*, vol. 15, no. 4, pp. 60–66, 2008.

[23] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[24] H. Song, R. Srinivasan, T. Sookoor et al., *Smart Cities: Foundations, Principles and Applications*, Wiley, 2017.

[25] X. Du, M. Guizani, Y. Xiao et al., "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1223–1229, 2009.

[26] L. Liu, C. Chen, T. Qiu, M. Zhang, S. Li, and B. Zhou, "A data dissemination scheme based on clustering and probabilistic broadcasting in VANETs," *Vehicular Communications*, vol. 13, pp. 78–88, 2018.

[27] X. Li and C. Qian, "Low-complexity multi-resource packet scheduling for network function virtualization," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 1400–1408, 2015.

[28] G. Sun, Y. Li, D. Liao, and V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1175–1191, 2018.

[29] G. Xiong, Y.-X. Hu, L. Tian, J.-L. Lan, J.-F. Li, and Q. Zhou, "A virtual service placement approach based on improved quantum genetic algorithm," *Frontiers of Information Technology and Electronic Engineering*, vol. 17, no. 7, pp. 661–671, 2016.

[30] G. Sun, Y. Li, Y. Li, D. Liao, and V. Chang, "Low-latency orchestration for workflow-oriented service function chain in edge computing," *Future Generation Computer Systems*, vol. 85, pp. 116–128, 2018.

[31] G. Sun, D. Liao, D. Zhao, Z. Xu, and H. Yu, "Live migration for multiple correlated virtual machines in cloud-based data centers," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 279–291, 2018.

[32] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2008–2025, 2017.

[33] R. Mijumbi, J. Serrat, J.-L. Gorricho et al., "Network function virtualization: state-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[34] W. Cerroni and F. Callegati, "Live migration of virtual network functions in cloud-based edge networks," in *Proceedings of the IEEE International Conference on Communications*, pp. 2963–2968, 2014.

[35] Y. Sang, B. Ji, G. R. Gupta, X. Du, and L. Ye, "Provably efficient algorithms for joint placement and allocation of virtual network functions," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 829–837, 2017.

[36] J. Batallé, J. F. Riera, E. Escalona, and J. A. García-Espín, "On the implementation of NFV over an OpenFlow infrastructure: Routing function virtualization," *Software Defined Networks for Future Networks and Services*, pp. 1–6, 2013.

[37] W. Ma, C. Medina, and D. Pan, "Traffic-aware placement of NFV middleboxes," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–6, 2015.

[38] Q. Sun, P. Lu, W. Lu, and Z. Zhu, "Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–6, 2017.

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

[40] S. Herker, X. An, W. Kiess, S. Beker, and A. Kirstaedter, "Data-center architecture impacts on virtualized network functions service chain embedding with high availability requirements," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–7, 2015.

[41] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 1–9, 2016.

[42] H. Jin, D. Pan, J. Xu et al., "Efficient VM placement with multiple deterministic and stochastic resources in data centers," in *Proceedings of the IEEE Global Communications Conference*, pp. 2505–2510, 2012.

[43] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–6, 2015.

[44] Q. Zhang, X. Wang, I. Kim, P. Palacharla, and T. Ikeuchi, "Vertex-centric computation of service function chains in multi-domain networks," in *Proceedings of the IEEE Conference on Network Softwarization*, pp. 211–218, 2016.

[45] T. Wen, H. Yu, G. Sun et al., "Network Function Consolidation in Service Function Chaining Orchestration," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, 2016.

[46] T. Park, Y. Kim, J. Park, H. Suh, B. Hong, and S. Shin, "QoSE: Quality of security a network security framework with distributed NFV," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, 2016.

[47] Y. Li, L. T. X. Phan, and B. T. Loo, "Network functions virtualization with soft real-time guarantees," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 1–9, 2016.

[48] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 60–66, 2015.

[49] G. Sun, Y. Li, H. Yu, A. V. Vasilakos, X. Du, and M. Guizani, "Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks," *Future Generation Computer Systems*, vol. 91, pp. 347–360, 2019.

[50] A. Haque and P.-H. Ho, "A study on the design of survivable optical virtual private networks (O-VPN)," *IEEE Transactions on Reliability*, vol. 55, no. 3, pp. 516–524, 2006.

[51] X. Xie, N. Hua, and X. Zheng, "Dynamic routing, wavelength and core allocation in multi-core fiber based optical networks considering inter-core crosstalk," in *Proceedings of the Signal Processing in Photonic Communications (pp.JM3A-4)*, Optical Society of America, 2015.

[52] K. Calvert, J. Eagan, S. Merugu, A. Namjoshi, J. Stasko, and E. Zegura, "Extending and enhancing GT-ITM," in *Proceedings of the ACM Sigcomm Workshop on Models*, pp. 23–27, 2003.