# Service Function Chain Mapping with Resource Fragmentation Avoidance

Zhikai Zhu, Hancheng Lu, Jian Li, Xiaoda Jiang
Department of Electrical Engineering and Information Science,
University of Science and Technology of China, Hefei, Anhui 230027 China
zkzhu@mail.ustc.edu.cn, hclu@ustc.edu.cn, lijian9@mail.ustc.edu.cn, jxd95123@mail.ustc.edu.cn

***Abstract*—In the context of Service Function Chain (SFC), SFC Mapping (SFCM) problem has a decisive impact on the resource utilization efficiency of physical networks, where node resources and link resources are concerned. However, most existing work on the SFCM problem has not taken into account difference in the amount of node resources and link resources. This kind of difference might result in some nodes that cannot be mapped because of insufficient link resources around these nodes. This phenomenon is referred to as resource fragmentation. In this paper, we attempt to improve the resource utilization efficiency by reducing resource fragmentation in physical networks, where SFCM is performed. First and most importantly, we propose a metric called Resource Fragmentation Degree (RFD) to quantify resource fragmentation. The basic idea behind RFD is that the resource availability of a node is determined by the residual link resources around the node. Based on RFD, we formulate the SFCM problem with goal of minimizing resource fragmentation. Furthermore, we also propose an efficient online heuristic to find the optimal mapping strategy. Simulation results show that much more SFC requests can be accepted by reducing resource fragmentation in physical networks and the proposed algorithm achieves more than 25% higher acceptance ratio compared with existing algorithms.**

***Index Terms*—Virtual Network Function, Service Function Chain, Service Function Chain Mapping, Resource Fragmentation**

## I. INTRODUCTION

Traditionally, network operators utilize dedicated hardware to host different network functions, such as firewalls, load balancers, network address translators and so on. Due to the expensive cost and weak extensibility of dedicated hardware, network operators have to pay high capital and operating expenses. To solve this problem, Network Function Virtualization (NFV), which allows different network functions to be implemented in Virtual Machines (VMs) hosted on high volume servers, is proposed by European Telecommunications Standards Institute [1]. These network functions are also called Virtual Network Functions (VNFs).

When network operators deal with service requests from customers, network flows are often required to pass through a sequence of VNFs in a particular order, which is known as Service Function Chaining (SFC) [2]. For each SFC request, the sequence of VNFs is required to be deployed in various physical network locations with certain constraints, which is referred to as Service Function Chain Mapping (SFCM) problem. The SFCM problem is of great importance in the context of SFC. It has a decisive impact on the resource utilization efficiency of physical networks. However, the SFCM problem is non-trivial as the complex resource distribution of physical networks and the chaining relationship among the VNFs should be jointly considered.

There exist much work on the SFCM problem. Some of them assume that all SFC requests are known in advance. Mehraghdam et al. [3] perform a Pareto set analysis to observe the relations between different optimization objectives. Ye et al. [4] formulate the problem as an Integer Linear Programming (ILP) problem with the goal of minimizing the bandwidth consumption and propose a heuristic algorithm to solve it. In practice, SFC requests need to be served and mapped one after another, and they arrive and depart dynamically, which is considered as the online version of SFCM problem. Lukovszki et al. [5] study the optimal online SFC embedding method to maximize the number of admitted requests. Cao et al. [6] use time-dependent duals to deal with flow arrivals and departures. Bari et al. [7] introduce Viterbi algorithm to minimize the capital expenditures and operational expenditures.

When SFCM is performed in physical networks, two kinds of resources, i.e., node resources and link resources, are concerned. However, most existing work has not taken into account difference in the amount of node resources and link resources. Instead, they mainly focus on optimizing the utilization efficiency for node resources or link resources individually. Although load balancing is applied in some of them, it is either among nodes or among links. That is to say, existing work might lead to imbalance in the usage of node resources and link resources. Consequently, the amount of node resources and link resources will show difference during SFCM. This kind of difference might result in reduced resource utilization efficiency in physical networks as the availability of node resources depends on the usage of link resources. When an SFC request arrives, there might be some nodes that cannot be mapped because of insufficient link resources around these nodes. This phenomenon is referred to as resource fragmentation. Resource fragmentation leads to some node resources are isolated and cannot be used during SFCM. In extreme cases, physical networks are partitioned when bottleneck links exhaust their resources. In the context of SFC, VNFs from different SFC requests are encouraged to be mapped on the same node so that the number of active VMs

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

can be reduced. By doing so, more resource fragmentation will exist.

Resource fragmentation has significant effect on the resource utilization efficiency of physical networks. This motivates us to improve the resource utilization efficiency by reducing resource fragmentation during SFCM. The most challenging problem is how to measure resource fragmentation in a quantitive manner. Moreover, considering SFC requests arrive and depart randomly in practice, online SFCM is applied in this paper. The main contributions of this paper are summarized as follows:

- We propose a new metric called Resource Fragmentation Degree (RFD) to quantify resource fragmentation in physical networks. During SFCM, a virtual link in SFC requests is mapped to paths in physical networks. Therefore, the derivation of RFD at a node is based on the residual link resources around the node instead of that directly connected to the node.
- Based on RFD, the SFCM problem is formulated as a Mixed Integer Program (MIP) problem with the goal of minimizing resource fragmentation in physical networks. Furthermore, an efficient online heuristic algorithm is proposed to solve the MIP problem.
- Extension simulations are implemented and performed to validate our work. The results show that resource fragmentation has a significant impact on the resource utilization efficiency of physical networks and thus the proposed algorithm can accept much more SFC requests compared with existing algorithms.

The rest of the paper is organized as follows. Section II describes the system model. In Section III, the definition of the proposed RFD is described. In Section IV, the MIP problem with the goal of minimizing resource fragmentation is formulated and solved by an efficient online heuristic algorithm. Section V gives the performance evaluation. Finally, the paper is concluded in Section VI.

## II. SYSTEM MODEL

In this section, we give an overview of the system model in the context of SFC. Particularly, resource fragmentation is illustrated when SFCM is performed. Node resources and link resources are concerned, which refer to node computational capacity and link bandwidth respectively. The definitions of important symbols used in this paper are summarized in Table I.

### A. Network Model

The physical network is modeled as an undirected weighted graph $G^P = (S^P, N^P, L^P)$, where $S^P$, $N^P$, $L^P$ denote the set of switch nodes, function nodes and physical links respectively. VNFs are assumed to be deployed on the VMs in function nodes while switches nodes are only used for mapping the ingress and egress of each SFC request. Each VM in function nodes has computational capacity (i.e., node resource) and each link has limited bandwidth resource (i.e., link resource). Different VMs on the same physical function node must host different VNFs, while VMs on different physical function nodes can support the same VNF.

TABLE I
NOTATIONS

| **Physical Network and VNF** | |
|---|---|
| $S^P$ | Physical switches nodes set. |
| $N^P$ | Physical function nodes set. |
| $n$ | The size of $N^P$. |
| $L^P$ | Physical links set. |
| $F$ | Set of all supported VNFs. |
| $M$ | Set of all VMs. |
| $C_i$ | The number of VMs $i \in N^P$ can host. |
| $U_i$ | Set of VMs hosted on node $i \in N^P$. |
| $C_M^{cap}(m)$ | The total capacity of $m \in M$. |
| $R_M^{cap}(m)$ | The residual capacity of $m \in M$. |
| $loc(m)$ | The physical function node where VM $m \in M$ deploys. |
| $func(m)$ | Function of $m \in M$. |
| $C_P^b(i,j)$ | The total bandwidth on $(i,j) \in L^P$. |
| $R_P^b(i,j)$ | The residual bandwidth on $(i,j) \in L^P$. |
| **SFC requests** | |
| $N^s$ | VNF nodes set of request s. |
| $L^s$ | virtual links set of request s. |
| $u^s, v^s$ | The ingress and egress switches of request s. |
| $func(k)$ | Function of $k \in N^s$. |
| $C_s^{cap}(k)$ | Required capacity of $k \in N^s$. |
| $C_s^b(k,l)$ | Required bandwidth of $(k,l) \in L^s$. |

We assume that the arrive time of SFC requests follows the poisson distribution and the lifetime obeys the exponential distribution. Each SFC request $s$ consists of one ingress node, one egress node, several VNF nodes and virtual links which connect the VNF nodes. It can be represented by a directed weighted graph $G^s = (N^s, L^s)$, where $N^s$ denotes the set of VNF nodes and $L^s$ represents the set of virtual links. VNF nodes and virtual links are associated with different computational capacity and bandwidth requirements, respectively.

### B. Resource Fragmentation

Fig. 1 shows the example of a physical network and two SFC requests. The physical network consists of 4 switch nodes, 4 function nodes and several links. There are several VMs installed in a function node and the number in rectangle texts nearby denotes the node's residual computational capacity. The residual bandwidth resource of a link is denoted by the number near that link. Initially, all VMs have the same residual capacity (i.e., 50 units) and all links have the same bandwidth resource (i.e., 35 units). Each SFC request has one ingress node, one egress node, several VNF nodes and virtual links. Numbers on top of VNF nodes and virtual links denote the requested capacities and bandwidth resources, respectively.

Fig. 2 gives two mapping strategies for two SFC requests. It is assumed that ingress and egress of SFC request 1 are located at switch $a$ and switch $d$ while ingress and egress of SFC request 2 are located at switch $b$ and switch $d$. With the first strategy, when SFC request 1 comes, its mapping procedure is: VNF1 $->A$, VNF2 $->C$, VNF3 $->D$. Then, SFC request 2 comes. In order to better utilize resources, different service requests are encouraged to share VMs. In this case,

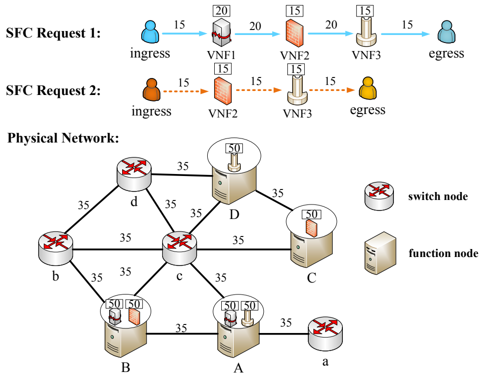
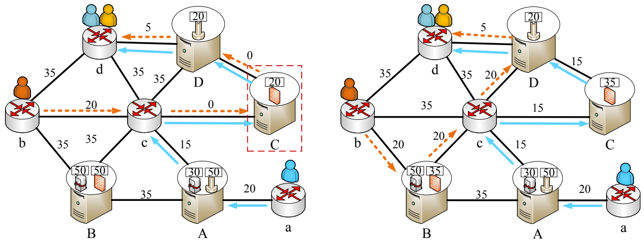


Fig. 1. Physical network and two SFC requests.

VNF2 of SFC request 2 should be mapped on node $C$ and VNF3 should be mapped on node $D$ as Fig. 2(a) shows. As we can see, there are 20 unit resources remaining in node $C$, but its adjacent links have exhausted their bandwidth resources. Therefore, before these two SFC requests departs, the 20 unit resources cannot be used by the incoming SFC requests. That is to say, the resources of node $C$ become fragmented resources. With the second strategy, on the contrary, VNF2 of request 2 is mapped on node $B$ as shown in Fig. 2(b), resource fragmentation can be avoided.



(a) Mapping solution one

(b) Mapping solution two

Fig. 2. Two mapping strategies for SFCM problem in Fig.1.

## III. DEFINITION OF RESOURCE FRAGMENTATION DEGREE

In this paper, RFD is proposed to quantify resource fragmentation in the physical network. We present the definition of RFD in this section. The basic idea behind RFD is also described in detail.

### A. Residual Resource Ratio

The definition of RFD is derived from the conception of connectivity in graph theory [8], which is defined as the number of vertexes or edges making the graph disconnected when they are removed. But it only defines the connectivity of the entire graph, and it does not define the connectivity of a single vertex. In this paper, we extend this definition to describe the connectivity of physical function nodes in the physical network.

Residual resource ratio is used to measure the connectivity of physical function nodes, which is defined as the ratio of residual resources to total resources at each element, where element can be referred to node, link or path in the physical network. This ratio represents the ability for the current element to connect its adjacent elements. When the ratio is 0, the resource of the element is used up, which might break up the connectivity of elements around it. In this case, some elements might be isolated and their resources are fragmented.

Since each physical function node runs several VMs, the residual resource ratio of it is defined as the ratio of the sum of residual capacities of VMs to the sum of total capacities of VMs, as shown in Equation 1.

$$\rho_i = \frac{\sum_{m \in M : loc(m)=i} R_M^{cap}(m)}{\sum_{m \in M : loc(m)=i} C_M^{cap}(m)}. \tag{1}$$

Each path between two nodes in the physical network consists of a sequence of physical links. In order to limit the mapping cost, it is assumed that the length of candidate mapping paths should not be greater than a constant $\epsilon$. When a virtual link of an SFC request is mapped on a path consists of a sequence of several physical links, the physical link referred to as bottleneck link which has the least bandwidth determines the path resource. Based on this observation, the residual resource ratio $\eta^p$ of one path $p$ is defined as the residual resource ratio of the bottleneck link. Then, the candidate path set between node $i$ and node $j$ is denoted as $path(i,j)$, where the number of this set is $|path(i,j)|$. Finally, the residual resource ratio of paths between node $i$ and node $j$ can be expressed as

$$\eta_{i,j} = \frac{\sum_{p \in path(i,j)} \eta^p}{|path(i,j)|}. \tag{2}$$

### B. Definition of Resource Fragmentation Degree

In the physical network, for function node $i$, there is a set of function nodes whose distances from node $i$ are less than $\epsilon$. The size of this set is denoted by $d_i$. In the physical network, the weighted adjacency matrix $T$ is exploited to represent connectivity of function nodes, which is expressed as follows.

$$T = \begin{bmatrix} 0 & \eta_{12} & \cdots & \eta_{1n} \\ \eta_{21} & 0 & \cdots & \eta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{n1} & \eta_{n2} & \cdots & 0 \end{bmatrix} \times \begin{bmatrix} \frac{1}{d_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{d_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{d_n} \end{bmatrix} \tag{3}$$

Where $\eta_{i,j}$ is defined in Equation (2) and $d_i$ can be obtained as follows.

$$d_i = \sum_j |path(i,j)|. \tag{4}$$

Then, the connectivity vector $\boldsymbol{\kappa}$ for physical function nodes can be defined as Equation (5), where $\boldsymbol{\lambda} = (\rho_1, \rho_2, \cdots, \rho_n)$ and $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \cdots, \kappa_n)$.

$$\boldsymbol{\kappa} = \boldsymbol{\lambda} * T \tag{5}$$

Based on $\boldsymbol{\kappa}$, RFD vector of physical function nodes can be derived as Equation (6), where $\boldsymbol{r} = (r_1, r_2, \cdots, r_n)$.

$$\boldsymbol{r} = \mathbf{1} - \boldsymbol{\kappa}. \tag{6}$$

From Equation (1)(2)(3)(5)(6), we can see that RFD of a physical function node increases as the residual resource ratio of its adjacent paths decreases. When all of its adjacent paths exhaust their resources, maximum RFD of the function node is reached which equals 1.

In order to further quantify the fragmented resources, we define the RFQ $\Delta$ as the product of the sum of residual capacities of VMs and RFD. RFQ of function node $i$ is express as

$$\Delta_i = r_i \sum_{m \in M: loc(m)=i} R_M^{cap}(m). \tag{7}$$

## IV. PROBLEM FORMULATION AND PROPOSED ALGORITHM

Based on the definition of RFD and RFG, we formulate the SFC problem with the goal of minimizing resource fragmentation. Furthermore, an effective online heuristic algorithm is proposed to solve the problem.

### A. Problem Formulation

With consideration of resource fragmentation, we formulate the SFCM problem as a MIP problem as follows.

*1) Variables:*

- $X_m^k$ : The value is 1 when VNF node $k \in N^s$ is allocated on VM $m \in M$ and 0 in other cases. When $X_m^k = 1$, the function of VNF node $k$ and VM $m$ must be consistent, i.e. $func(k) = func(m)$.
- $Y_i^k$ : The value is 1 when VNF node $k \in N^s$ is mapped on the physical function node $i \in N^P$ and 0 in other cases. When $Y_i^k = 1$, there must have $X_m^k = 1$ and $loc(m) = i$.
- $Z_{i,j}^{k,l}$ : The value is 1 when virtual link $(k,l) \in L^s$ is mapped on physical link $(i,j) \in L^P$ and 0 in other cases.

*2) Objective function:* We aim at minimizing the total RFQ.

$$minimize \sum_{i \in N^P} r_i \sum_{m \in M: loc(m)=i} R_M^{cap}(m) \tag{8}$$

*3) Constraints:* There are three kinds of constraints involved, including capacity constraints, flow constraints and deployment constraints.

**Capacity constraints:**

$$X_m^k C_s^{cap}(k) \le R_M^{cap}(m),\ \forall k \in N^s, m \in M \tag{9a}$$

$$Z_{i,j}^{k,l} C_s^b(k,l) \le R_P^b(i,j),\ \forall (i,j) \in L^P, (k,l) \in L^s \tag{9b}$$

Equation (9a) ensures that the capacity required by the VNF node of current request do not exceed the required capacity of VM. Equation (9b) makes sure that the physical link has enough bandwidth to accommodate the virtual links.

**Flow Constraints:**

$$Z_{i,j}^{k,l} + Z_{j,i}^{k,l} \le 1,\ \forall (i,j) \in L^P, (k,l) \in L^s \tag{10a}$$

$$\sum_{(i,j) \in L^P} (Z_{i,j}^{k,l} - Z_{j,i}^{k,l}) = Y_i^k - Y_i^l,\ \forall k,l \in N^s, (k,l) \in L^s \tag{10b}$$

Equation (10a) ensures that each virtual link of SFC request can only be mapped on one direction of physical link. Equation (10b) indicates that in-flow and out-flow of each middle physical node is equal.

**Deployment Constraint:**

$$\sum_{m \in M} X_m^k = 1,\ \forall k \in N^s \tag{11}$$

Equation (11) dedicates that each VNF node can only be mapped on one VM.

Given a solution of this MIP problem, we can test whether it satisfies Equations (9)-(11), but it is hard to solve the optimization problem within polynomial time [7], with the consideration of network size. Hence, this MIP problem is an NP-hard problem.

### B. Proposed SFCM-RFD Algorithm

In order to solve the SFCM problem, a novel online heuristic algorithm is proposed, which is called Service Function Chain Mapping based on Resource Fragmentation Degree (SFCM-RFD). This algorithm involves RFD and RFQ to choose suitable mapping location. The inputs of the algorithm are the physical network and dynamic SFC requests while the output is a solution that maps SFC requests to the physical network. During the mapping process, the proposed algorithm chooses proper locations in the physical network for SFC requests in a heuristic manner. Some definitions used in the proposed algorithm are given as follows.

- $\Omega(k)$ : candidate physical function node set of VNF node $k$ of SFC request
- $\Psi(k)$ : physical function node which virtual node $k$ is mapped on

Details of SFCM-RFD are described in $Algorithm$ 1. When an SFC request arrives, SFCM-RFD first executes the initial procedure at line 3 to set all nodes and links in unmapped states. The mapping procedure begins with the mapping of ingress and egress of the SFC request. For each VNF node $k \in N^s$, SFCM-RFD first finds the candidate physical function nodes set $\Omega(k)$, where each physical node runs the VM that instantiates the VNF of node $k$ and the capacity of the VM satisfies the resource requirement of node $k$. If $\Omega(k)$ is empty, the SFC request will be rejected. SFCM-RFD finishes node mapping and link mapping with $Algorithm$ 2. If the mapping fails, the algorithm will be backtracked to the previous node and repeat the above process. The total backtrack time is limited by $L$. If current backtrack time $t$ is larger than $L$, the SFC requests will be rejected. When all nodes and links of the SFC request are mapped in the physical network, SFCM-RFD updates the physical network status and prepares for processing the next SFC request.

**Algorithm 1** SFCM-RFD Algorithm

```
Input physical network $G^P$, SFC requests
Output mapping result of SFC requests
set all nodes and links unmapped states
map ingress and egress of SFC requests
select a positive integer $L$
while all nodes $k \in N^S$ have been mapped do
    if $\Omega(n^v)$ is empty then
        return SFC request rejected
    end if
    Algorithm 2: Node and link mapping.
    if VNF node $k$ is mapped successfully then
        $k \leftarrow k+1$
    else
        if $k > 0, t \leq L$ then
            $t \leftarrow t+1$
            $k \leftarrow k-1$
            remove node $k$ from physical network
        else
            SFC request rejected
        end if
    end if
end while
update all states of physical network.
```

SFCM-RFD uses $Algorithm$ 2 to choose the best mapping location. The algorithm calculates $C$ for each candidate physical function node $i \in \Omega(k)$. When calculating $C$, the node $k$ and its adjacent virtual links are mapped temporarily. If one of the virtual links cannot be mapped, $C$ is considered as $+\infty$. Otherwise, $C$ is set to increased RFQ. If the minimum $C$ is $+\infty$, it indicates that there are no physical paths between two physical nodes to be mapped. In this case, mapping failure will be produced. At last, SFCM-RFD maps node $k$ and its adjacent links which connect the mapped neighbor VNF nodes.

SFCM-RFD involves two major procedures: K-shortest-paths search and $C$ calculation between any two physical nodes. The time complexity of K-shortest-paths algorithm is a polynomial type [9]. When calculating $C$ for each physical function node, neighbor nodes of current selected VNF node and corresponding shortest paths are compared, so its calculation complexity is also considered as a polynomial type. Moreover, an upper bound is introduced to avoid the exponential time complexity when using backtracking method.

## V. PERFORMANCE EVALUATION

### A. Simulation Settings

Simulations are carried out by using a C++ simulator. The GT-ITM tool [10] is used to generate the physical network topology and SFC requests. In simulations, there are 10 switch nodes, 40 function nodes and 129 links in physical network. Also, 10 different kinds of VNFs are used and each SFC request is a sequence of several VNFs. All SFC requests are generated according to a poisson process with an average arrival rate of $\lambda$ requests per 1000 time units. Other simulation parameters are shown in Tab.II, which follow the uniform distributions. The settings of these parameters and distributions are inspired by the simulations of a well-studied virtual network embedding problem [11].

**Algorithm 2** Node and link mapping

```
for $i \in \Omega(k)$ do
    map node $k$ on physical node $i$
    for each mapped neighbor VNF node $k'$ of $k$ do
        map virtual link $k'k$ on physical path
    end for
    if one virtual link cannot be mapped then
        $C(i) \leftarrow +\infty$
    else
        $C(i) \leftarrow$ increased RFQ
    end if
    remove node $k$ and its adjacent virtual links
end for
find the physical node $i_{min}$ with lowest $C$
if $C(i_{min})$ is $+\infty$ then
    return FAILED
end if
map VNF node $k$ on physical node $i_{min}$
for each mapped neighbor VNF node $k'$ of $k$ do
    map virtual link $k'k$ on physical path $\Psi(k')i$
end for
return SUCCESS
```

TABLE II
SIMULATION PARAMETERS RANGES

| Parameter | Minimum | Maximum |
|---|---|---|
| Number of VMs in each function node | 2 | 4 |
| Bandwidth for each physical link | 50 | 100 |
| Capacity of each VM | 50 | 100 |
| Number of VNF nodes per SFC | 5 | 10 |
| Required capacity for each VNF node | 20 | 30 |
| Required bandwidth for each virtual link | 20 | 30 |

The proposed algorithm is compared with two typical existing algorithms: ProvisionTraffic [7] and COATS [6]. ProvisionTraffic solves the SFCM problem by the Viterbi algorithm. COATS constructs a layered graph and finds one shortest path from ingress node to egress node in this layered graph to minimize the link resource consumption.

### B. Simulation Results

Fig. 3 and Fig. 4 show the acceptance ratio of SFC requests and mean RFQ of physical function nodes respectively. The value of $\lambda$ is set to 100. The acceptance ratio equals to the number of mapped requests dividing the total ones. In the first 500 time units, RFQ rises sharply due to the resource consumption of physical resources. During this period, many SFC requests are rejected due to insufficient resources in the

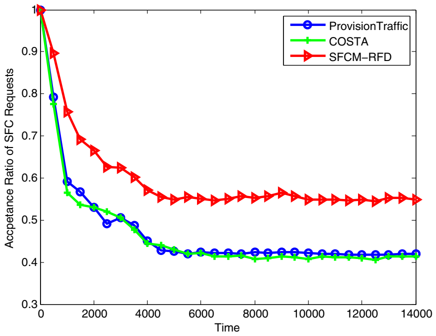


Fig. 3. Acceptance ratio of SFC requests over time ($\lambda$=100).

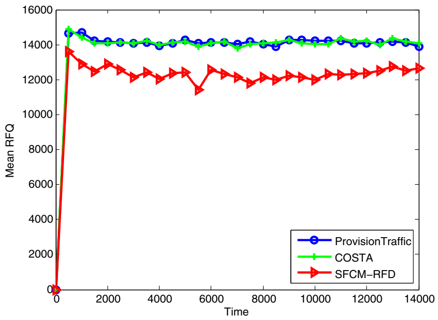


Fig. 4. Mean RFQ of physical function nodes over time ($\lambda$=100).

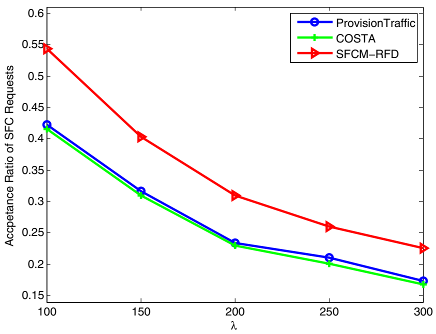


Fig. 5. Acceptance ratio of SFC requests over different $\lambda$.

physical network, so the acceptance ratio reduces quickly. Then, RFQ fluctuates within a short range and the acceptance ratio converges gradually to steady value. Besides, Fig. 3 and Fig. 4 show that the proposed SFCM-RFD outperforms other two heuristic algorithms with about 25% higher acceptance ratio and fewer RFQ. This is due to the fact that SFCM-RFD follows the principle of minimizing the RFQ, and it can effectively reduce fragmented resources and improve the resource utilization. In contrast, the other two algorithms ignore the effect of resource fragmentation, causing the rejection of more SFC requests.

Fig. 5 illustrates the acceptance ratio of SFC requests when $\lambda$ changes from 100 to 300. When the arrival rate is higher, there will be more resource consumption in the physical network. In this case, more SFC requests will be rejected due to insufficient physical network resources. Fig. 5 shows that SFCM-RFD achieves better performance with more than 25% higher acceptance ratio. The reason is that higher arrival rate results in more difference in the amount of node resources and link resources, which makes resource fragmentation more severe. In this case, minimizing fragmented resources is important when mapping SFC requests. These results also indicate that the proposed algorithm and definition of RFD are effective.

## VI. CONCLUSION

In this paper, we attempt to improve the resource utilization efficiency by reducing resource fragmentation in the context of SFC. First and most importantly, we propose RFD to measure resource fragmentation in physical networks in a quantitive manner. Based on RFD, we formulate the SFCM problem with the goal of minimizing resource fragmentation. Furthermore, an efficient online heuristic algorithm is proposed to solve the problem. Simulation results validate the proposed algorithm outperforms existing algorithms in terms of acceptance ratio of SFC requests.

As future work, we will extend our work to solve the SFCM problem with service-specific constraints.

## VII. ACKNOWLEDGMENT

This work was supported in part by the National Key R&D Program of China (2016YFB0800301), the National Science Foundation of China (No.61390513, 91538203), the Fundamental Research Funds for the Central Universities.

## REFERENCES

[1] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, pp. 90–97, Feb 2015.

[2] P. Quinn and T. Nadeau, "Problem statement for service function chaining," tech. rep., IETF, Fremont, CA, USA, RFC 7498, Apr. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7498.txt.

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

[4] Z. Ye, X. Cao, and C. Qiao, "Joint topology design and mapping of service function chains in network function virtualization," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2016.

[5] T. Lukovszki and S. Schmid, "Online admission control and embedding of service chains," in *International Colloquium on Structural Information and Communication Complexity*, pp. 104–118, Springer, July 2014.

[6] Z. Cao, M. Kodialam, and T. Lakshman, "Traffic steering in software defined networks: planning and online routing," *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 65–70, August 2014.

[7] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 725–739, Dec 2016.

[8] H. Nagamochi and T. Ibaraki, *Algorithmic aspects of graph connectivity*, vol. 123. Cambridge University Press New York, 2008.

[9] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[10] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, pp. 594–602, Mar 1996.

[11] R. Mijumbi, J. Serrat, and J.-L. Gorricho, "Self-managed resources in network virtualisation environments," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pp. 1099–1106, IEEE, May 2015.