Cost-aware Service Function Chaining With Reliability Guarantees in NFV-enabled Inter-DC Network

Xuxia Zhong, Ying Wang, Xuesong Qiu
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications, Beijing, China
Email: {zhongxuxia, wangy}@bupt.edu.cn

***Abstract*—Network function virtualization (NFV) moves hardware-based middleboxes to software-defined virtual network functions (VNFs) running on standard machines. By incorporating NFV in inter-DC network, subscribers can orchestrate service function chains (SFCs) intelligently and deploy VNFs faster. However, orchestrating SFCs in inter-DC network will incur high deployment cost, including expensive cross-DC bandwidth cost and virtual network function cost. Moreover, the NFV-enabled SFCs impose higher reliability requirements due to the functional characteristics. The entire SFCs will break down if any of the VNFs fails. To overcome these challenges, we propose a *C*ost-aware and *R*eliability-guaranteed *S*FC *O*rchestration (CRSO) scheme. CRSO firstly orchestrates SFCs in inter-DC network with less cost based on Hidden Markov Model (HMM). Then it backups VNFs to satisfy the reliability requirements. Experimental results show CRSO performs well for using about 20.4% less cost compared with the existing algorithm for accommodating the same SFCs.**

***Index Terms*—Network Function Virtualization, Service Function Chain Orchestration, Hidden Markov Model**

I. INTRODUCTION

In order to accomplish network services, traffic flows are usually processed by a list of network functions (NFs) according to a particular order. The sequence of network functions is service function chain (SFC) [1]. Network operators implement SFCs by steering flows to the hardware middleboxes located in local networks. Traditional middleboxes exhibit limitations in terms of network flexibility, scalability, manageability and operational efficiency [2]. Fortunately, network function virtualization (NFV) emerges as a new network architecture which decouples the NFs from the dedicated hardware appliances to virtual network functions (VNFs) running on commercial off-the-shelf (COTS) equipment [3]. Such an approach reduces capital expenditure (CAPEX) and operational expenditure (OPEX) for service deployment [4].

Recently, with the rise of cloud computing and big data, inter-data center (inter-DC) networks that connect geographically distributed data centers have been deployed rapidly [5]. As the public clouds in inter-DC networks usually offer a pay-as-you-go charging system and support elastic service scaling, the operational cost and complexity of maintenance can be significantly reduced [4]. By implementing VNFs in inter-DC networks, SFC subscribers can use the network resources flexibly and deploy services cheaper and faster. There are already several architectures (e.g., APLOMB [6] and Jingling [7]) proposed to migrate the virtual network functions from local networks to inter-DC networks. Despite the promising advantages, there are challenges need to be tackled in SFC orchestration in inter-DC network.

Challenge 1: how to orchestrate SFCs in inter-DC network with less cost? Orchestrating SFCs in inter-DC network will produce cross-DC bandwidth cost and VNF cost. Firstly, it is common to orchestrate SFCs across geographically distributed DCs to satisfy the location constraints or performance requirements [4]. For example, proxies and caches should be deployed near to the enterprise network, packet filters should be placed close to traffic sources [8]. However, the unit price of the inter-DC bandwidth is expensive. Secondly, data centers offer VNFs with various types, processing capacities and prices [4]. In an off-line SFC orchestration environment where hundreds or thousands of NFs need to be orchestrated on demand, the VNF cost occupies an important part. Therefore, it is challenging to orchestrate SFCs in an off-line manner with less cost.

Challenge 2: how to provide reliability-guaranteed network services? Except for the benefits of NFV, virtualized network functions impose new reliability concern. VNFs are deployed in generalized hardware that lacks the robustness of the specialized telecommunications equipment, thus the fault probability of VNFs are higher than traditional middleboxes [9]. Moreover, as the service request is in the form of SFC consisting a sequence of VNFs, the reliability of an end-to-end service is not determined by a single component but by the entire composition of the SFC [10]. Redundancy is an effective method to improve the reliability of a system. However, how to choose sufficient backup VNFs to satisfy the reliability requirement is intractable.

Inspired by the problems mentioned above, we study to orchestrate SFCs in inter-DC network cost-efficiently, while providing economical redundancy method to guarantee the reliability requirements of the subscribers. We firstly propose a comprehensive cost model which contains the inter-DC bandwidth, VNF and backup cost, and formulate the problem as an integer linear programming (ILP). With the ILP model, we can prove the problem is NP-hard. Then We design a time-efficient heuristic algorithm **C**ost-aware and **R**eliability-guaranteed **S**FC **O**rchestration (**CRSO**) to solve it. CRSO contains two modules: “Module 1: SFC Orchestration” and “Module 2: VNF backup”. Module 1 orchestrates SFCs cost-efficiently and provides best effort in reliability requirement

978-3-903176-15-7

based on Hidden Markov Model (HMM). Module 2 backups the VNFs with higher cost-efficient measure (CEM) defined in section 4.2 to enhance the reliability of SFCs more efficiently.

The rest of the manuscript is organized as follows. Some related works are presented in Section 2. Section 3 illustrates the problem overview, explains the mathematical model and formulates the problem in detail. Section 4 presents our proposed algorithm CRSO. Extensive simulations are in Section 5. Lastly, we give a brief conclusion in Section 6.

II. RELATED WORK

Some researchers have investigated the problem of multi-domain SFC orchestration. Q. Zhang *et al.* in [11] presented a vertex-centric distributed orchestration framework for multi-domain networks. The physical infrastructure information was maintained locally within each domain, hence there was an orchestrator associated within each domain to manage all the resources and communicate with other domains. They were interested in multi-domain area, but the paper was not fully compliant with the scenario envisioned in our work, where there is one network domain and multiple DC domains. Q. Xu *et al.* in [12] minimized the end-to-end latency when deploying cross-domain service function chains for 5G applications. For large-scale network, they proposed a heuristic approach based on Viterbi. However, they mainly considered the delay and ignored the SFC reliability. H. Chen [4] tackled the cost-efficient SFC orchestration problem across multiple clouds. They formulated the problem into an ILP. However, they didn't consider the sharing of VNFs which decreases the problem complexity.

Some papers studied the problem of reliability-aware SFC orchestration. J. Fan *et al.* in [13] presented a heuristic algorithm to minimize the physical resources consumption while guaranteeing the high reliability requirement. They considered all the bandwidth usage after adding the duplicates. However, our orchestration environment is inter-DC network. Normally, the intra-DC bandwidth is ignored as it is more cheaper than the inter-DC bandwidth. Therefore, the orchestration in [13] cannot be applied in our situation. Q. Long *et. al* in [14] established a reliability-aware and delay-constrained routing optimization framework for NFV-enabled data center networks. Their backup nodes were placed further downstream along the service's path therefore increasing number of hops and leading to longer end-to-end delay. However, our backup scheme is a active/standby mode, the backup VNF is activated once the primary fails. As the VNF instance can be shared by multiple SFCs to increase the resource utilization, W. Ding *et al.* in [15] proposed a cost-efficient redundancy method based on their designed Cost-Importance Measure (CIM) to select the backup candidate and the PN hardware. However, their physical node represents a server with limited resource capacity which is different from the environment of inter-DC network.

III. PROBLEM STATEMENT

A. Problem Overview

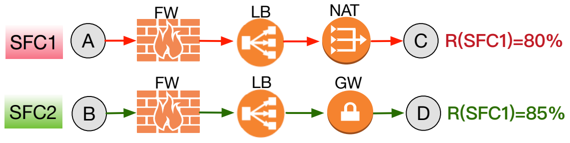
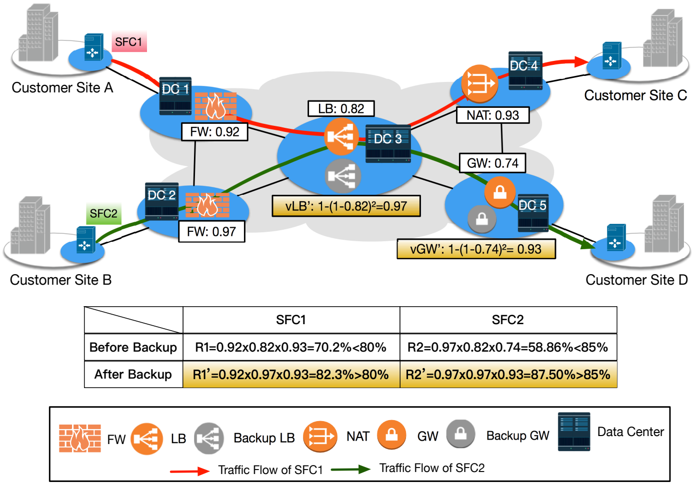


Fig. 1. Service function chains with reliability requirements



	SFC1	SFC2
Before Backup	R1=0.92x0.82x0.93=70.2%<80%	R2=0.97x0.82x0.74=58.86%<85%
After Backup	R1'=0.92x0.97x0.93=82.3%>80%	R2'=0.97x0.97x0.93=87.50%>85%

Fig. 2. SFC orchestration and redundancy scheme

To illustrate the problem, we will use an example here. Figure 1 depicts two SFCs to be orchestrated. SFC1 originates from customer site A and ends to customer site C. Its data needs to be processed by firewall (FW), load balancer (LB) and network address translation (NAT) in sequence. The requested reliability of SFC1 is 80%. For SFC2, its traffic traverses through FW, LB and GW in sequence, and the requested reliability is 85%. SFC1 and SFC2 are orchestrated in an inter-DC network depicted in Figure 2.

In the first step, the SFCs are orchestrated in a cost-efficient manner which offers best effort service in guaranteeing their reliability requirement. Before we add the backup VNF nodes, SFC1 passes through virtual firewall, vLB and vNAT that are respectively deployed in DC1, DC3 and DC4. The reliability of SFC1 reaches to 70.2% $(0.92 \times 0.82 \times 0, 93)$ which is lower than 80%. SFC2 passes through vFW, vLB and vGW that are respectively deployed in DC2, DC 3 and DC 5. The reliability for SFC2 is 58.86% $(0.97 \times 0.82 \times 0.74)$ which is lower than its expected value 85%. It should be noted that as virtualization can provide isolation between NFs running on the same VNF instance, VNFs can be shared by multiple SFCs to improve the resource utilization [16]. Therefore, the vLB deployed in DC3 are shared to by SFC1 and SFC2 in Figure 2.

In the second step, we backup VNFs to satisfy the reliability requirement with backup nodes. An advisable choice is adopted to enhance the reliability of the SFCs more efficiently. For instance, backing up vLB in DC3 will improve the reliability of both SFCs. After backup, the reliability of vLB changes to $1-(1-0.82)^2 = 0.97$. The reliability of SFC1 increases to 82.3% $(0.92 \times 0.97 \times 0.93)$ which is higher than 80%. Moreover, backing up the most unreliable node will increase the reliability significantly. The most unreliable vGW in DC5 increases to $1-(1-0.74)^2 = 0.93$ after adding a backup node. The reliability iof SFC2 increases to 87.50%

($0.97 \times 0.97 \times 0.93$) which is higher than 85%.

B. Network Model

Service Function Chain (SFCs). In an off-line manner, there are a total of P SFCs denoted as $\boldsymbol{SFC} = \{\boldsymbol{S^1}, \boldsymbol{S^2}, \boldsymbol{...S^P}\}$ to be orchestrated. One SFC is denoted as $\boldsymbol{S^p} = \{n_1^p, n_2^p, ...n_K^p\}$ ($1 \leq p \leq P$, $K = |S^p|$). σ^p and ξ^p represent the starting and ending location of S^p. n_i^p ($1 \leq i \leq K$) denotes the ith NF node of S^p. As every network function has a function type attribute (*i.e.*, firewall, IDS, IPS, proxy, *etc.*), we use t_i^p to describe the type of n_i^p. ρ_i^p denotes the requested IT resources of n_i^p. The NF node has a deployment location constraint denoted as set $\boldsymbol{L_i^p}$. In addition, $e_{(i-1)i}^p = (n_{(i-1)}^p, n_i^p)$ denotes the link between $n_{(i-1)}^p$ and n_i^p. $\beta_{(i-1)i}^p$ is the requested bandwidth of $e_{(i-1)i}^p$. The reliability requirement of $\boldsymbol{S^p}$ is represented as ϕ^p.

Inter-DC Network. The inter-DC network is denoted by graph $G^N = (D^N, E^N)$, where D^N and E^N denote the data centers and the links between them. We assume there are a total of $M = |D^N|$ data centers in G^N. $e_{mn}^N = (d_m, d_n) \in E^N$ is the link between d_m and d_n ($d_m, d_n \in D^N$). β_{mn}^N and η_{mn}^N represent the bandwidth capacity and peering link cost of e_{mn}^N respectively.

Data center. d_m represents a data center in D^N. We use v_s^m to denote a VNF instance deployed in d_m, the function type of v_s^m is denoted by t_s^m. All the VNF instances deployed in d_m are represented by V_m. Moreover, $\rho(t_s^m)$ and $r(t_s^m)$ which are related to the function type are used to represent the IT resources and reliability of v_s^m. α_m denotes the unit price of the IT resource in d_m. All the notations are summarized in Table 1.

C. Problem Formulation

We try to orchestrate the off-line SFCs in a cost-efficient way, while ensuring their reliability requirements. There are three types of costs generated during the orchestration procedure, inter-DC bandwidth cost, VNF cost and backup cost. The cost modelling are formulated as follow.

1) Inter-DC Bandwidth Cost:

The inter-DC bandwidth cost is represented by CL. $M(e_{(i-1)i}^p) \rightarrow e_{mn}^N$ denotes $e_{(i-1)i}^p$ passing through physical link e_{mn}^N. We use a binary variable $x_{mn}^{(i-1)i}$ to denote the mapping of link $e_{(i-1)i}^p$.

$$x_{mn}^{(i-1)i} = \begin{cases} 1 & M(e_{(i-1)i}^p) \rightarrow e_{mn}^N \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

If $e_{(i-1)i}^p$ traverses e_{mn}^N ($x_{mn}^{(i-1)i} = 1$), the inter-DC bandwidth cost of $e_{(i-1)i}^p$ is represented by $P_{mn}^{(i-1)i}$:

$$P_{mn}^{(i-1)i} = \beta_{(i-1)i}^p \times \eta_{mn}^N \tag{2}$$

Therefore, the total inter-DC bandwidth cost CL can be calculated as follow:

$$CL = \sum_{p \in [1,P]} \sum_{e_{mn}^N \in E^N} \sum_{i \in [2,K]} x_{mn}^{(i-1)i} \times P_{mn}^{(i-1)i} \tag{3}$$

TABLE I

NOTATIONS

Service Function Chain	
$\boldsymbol{S^p} = \{n_1^p, n_2^p, ...n_K^p\}$	service function chain
σ^p, ξ^p	starting and ending location of $\boldsymbol{S^p}$
n_i^p ($1 \leq i \leq K$)	the ith NF node in $\boldsymbol{S^p}$
$e_{(i-1)i}^p$	link between $n_{(i-1)}^p$ and n_i^p
t_i^p, ρ_i^p	function type and requested resource of n_i^p
$\beta_{(i-1)i}^p$	requested bandwidth of $e_{(i-1)i}^p$
$\boldsymbol{L_i^p}$	candidate deployment location of n_i^p
ϕ^p	reliability requirement of $\boldsymbol{S^p}$
Inter-DC Network	
$G^N = (D^N, E^N)$	inter-DC network
e_{mn}^N	link between d_m and d_n
β_{mn}^N	bandwidth capacity of e_{mn}^N
η_{mn}^N	peering link cost of e_{mn}^N
Data Center	
d_m	the m^{th} data center in D^N
v_s^m	the s^{th} VNF in d_m
t_s^m	function type of v_s^m
V_m	all the VNFs deployed in d_m, $v_s^m \in V_m$
$\rho(t_s^m)$, $r(t_s^m)$	IT resources and reliability of v_s^m
α_m	unit price of IT resource in d_m
Independent Variable	
$x_{mn}^{(i-1)i}$	equals to 1 if $e_{(i-1)i}^p$ passes e_{mn}
$y_{v_s^m}$	equals to 1 if v_s^m is in use
$z_{v_s^m}^i$	equals to 1 if n_i^p is deployed in v_s^m
$\psi_{v_s^m}$	equals to 1 if we backup v_s^m
Dependent Variable	
$P_{mn}^{(i-1)i}$	inter-DC bandwidth cost if $x_{mn}^{(i-1)i} = 1$
$\lambda_{v_s^m}^i$	VNF cost if $z_{v_s^m}^i = 1$
$\Delta_{v_m^s}$	incremental reliability of v_m^s if $\psi_{v_s^m} = 1$
$r_{v_s^m}^i$	the reliability of v_s^m if $z_{v_s^m}^i = 1$
R^p	the reliability of $\boldsymbol{S^p}$ after orchestration
Cost	
C	total cost
CL	inter-DC bandwidth cost
CN	VNF cost
CB	backup cost

2) VNF Cost:

All the NF nodes requested by SFCs are packed into VNF instances to accomplish the SFC chaining. If we consider the VNF instances as bins with a fixed capacity and the requested NFs as items with different sizes, the deployment of VNFs is transformed into a bin packing problem. We use a binary variable $y_{v_s^m}$ to denote the usage of the VNF v_s^m:

$$y_{v_s^m} = \begin{cases} 1 & v_s^m \text{ is in use} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

As a NF can only be packed into the VNF with the same function type, we use $\Lambda_{v_s^m}^i$ to check whether n_i^p and v_s^m have the same function type.

$$\Lambda_{v_s^m}^i = t_i \odot t_s^m = \begin{cases} 1 & t_i^p == t_s^m \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$\Lambda_{v_s^m}^i$ equals to 1 if n_i and v_s^m have the same type. $D(n_i^p) \rightarrow v_s^m$ denotes packing n_i^p into v_s^m. We use a binary variable $z_{v_s^m}^i$ to represent the packing of n_i^p:

$$z_{v_s^m}^i = \begin{cases} 1 & D(n_i^p) \rightarrow v_s^m,\ \Lambda_{v_s^m}^i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Moreover, we use $\lambda^i_{v^m_s}$ to denote the cost for packing n^p_i in v^m_s:

$$\lambda^i_{v^m_s} = \begin{cases} 0 & \text{exist } v^m_s \text{ to deploy } n^p_i \\ \rho(t^m_s) \times \alpha_m & \text{generate a new } v^m_s \end{cases} \tag{7}$$

If there exists a VNF v^m_s to pack n^p_i, then n^p_i is deployed in this instance with no cost ($\lambda^i_{v^m_s} = 0$) . Otherwise, a new VNF will be generated at the cost of $\rho(t^m_s) \times \alpha_m$. Therefore, the VNF cost denoted by CN can be calculated as follow:

$$CN = \sum_{d_m \in D^N} \sum_{v^m_s \in V^m} y_{v^m_s} \times \rho(t^m_s) \times \alpha_m \tag{8}$$

3) Backup Cost:

We enhance the reliability of SFCs with dedicated backup nodes. A binary variable $\psi_{v^m_s}$ is used to represent the redundancy of v^m_s.

$$\psi_{v^m_s} = \begin{cases} 1 & \text{backup } v^m_s \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

We use $\Delta_{v^s_m}$ to denote the incremental reliability of v^s_m after redundancy. It can be calculated as follow:

$$\Delta_{v^s_m} = 1 - (1 - \psi_{v^m_s} \cdot r(t^m_s))^t \tag{10}$$

t in Equation (10) represents the number of backups of v^s_m. Hence the reliability of $\boldsymbol{S^p}$ which is denoted as R^p can be calculated as follow:

$$R^p = \prod_{i \in [1,K]} r^i_{v^m_s} \tag{11}$$

$r^i_{v^m_s}$ in Equation (11) represents the reliability of v^m_s after being deployed in v^m_s ($z^i_{v^s_m} = 1$). $r^i_{v^m_s}$ can be calculated as follow:

$$r^i_{v^m_s} = r(t^m_s) + \psi_{v^m_s} \times \Delta_{v^s_m} \tag{12}$$

Therefore, the backup cost CB can be calculated as follow:

$$CB = \sum_{d_m \in D^N} \sum_{v^m_s \in V^m} \psi_{v^m_s} \times \rho(t^m_s) \times \alpha_m \tag{13}$$

4) Problem Statement:

With a goal to minimize the orchestration cost and satisfy the reliability requirements of the requested SFCs, we formulate the problem as an integer linear programming (ILP):

Objective:

$$C = Minimize(CL + CN + CB) \tag{14}$$

where C is the overall cost, which is the sum of the inter-DC bandwidth cost CL, VNF cost CN and backup cost CB.

Capacity Constraints:

$$\sum_{p \in P} \sum_{e^p_{(i-1)i}} x^{(i-1)i}_{mn} \times \beta^p_{(i-1)i} \leq \beta^N_{mn} \tag{15}$$

Equation (15) expresses the constraint of the bandwidth capacity. The summation of all the requested bandwidth of flows passing through substrate link e^N_{mn} must be less than its available bandwidth capacity β^N_{mn}.

$$\sum_{p \in P} \sum_{n^p_i} z^i_{v^m_s} \times \rho^p_i \leq \rho(t^m_s) \tag{16}$$

Algorithm 1 CRSO Overview

```
Require: substrate topology $G^N = \{D^N, E^N\}$, SFC set $\boldsymbol{SFC = \{S^1, S^2, ...S^p\}}$, $\boldsymbol{S^p} = \{n^p_1, n^p_2, ...n^p_K\}$
Ensure: Orchestration results $\boldsymbol{Q} = \{\boldsymbol{Q^1}, \boldsymbol{Q^2}...\boldsymbol{Q^P}\}$ and backup results $\psi_{v^m_s}$
 1: /*Cost-aware SFC Orchestration*/
 2: while ($\boldsymbol{SFC} \neq NULL$) do
 3:     Select the longest $S^p$ from $\boldsymbol{SFC}$
 4:     $\boldsymbol{Q^p} \leftarrow$ Module1: SFC orchestration($G^N$, $S^p$)
 5:     Insert $\boldsymbol{Q^p} = \{Q^p_1, Q^p_2...Q^p_K\}$ into $\boldsymbol{Q}$
 6:     Delete $S^p$ from set $\boldsymbol{SFC}$
 7: end while
 8: /*VNF Backup*/
 9: Calculate the reliability of $S^p$ after orchestration
10: Select the $S^p$ which do not satisfy the reliability requirement and put them in set $\boldsymbol{eSFC}$
11: $\psi_{v^m_s} \leftarrow$ Module 2: VNF Backup ($G^N$, $\boldsymbol{Q}$, $\boldsymbol{eSFC}$)
```

Equation (16) enforces the resource capacity bounds of VNF instance v^m_s. It means that the summation of all the requested resource of NFs deployed in v^m_s must be less than or equal to its available resource $\rho(t^m_s)$.

Location constraint:

$$\sum_{d_m \in D^N} \sum_{v^m_s \in V^m} z^i_{v^m_s} = 1, d_m \in \boldsymbol{L^p_i} \tag{17}$$

Equation (17) makes sure that only one VNF instance is selected for every network function node, and the deployment location of n^p_i must be within its data center set $\boldsymbol{L^p_i}$.

Reliability Constraint:

$$R^p \geq \phi^p, \forall \boldsymbol{S^p} \in \boldsymbol{SFC} \tag{18}$$

Equation (17) guarantees the reliability requirement of $\boldsymbol{S^p}$.

Connectivity Constraint:

$$\sum_{e^N_{mn} \in E^N} x^{(i-1)i}_{mn} - \sum_{e^N_{nm} \in E^N} x^{(i-1)i}_{nm} = \begin{cases} 1, & \text{if } \sum_{v^m_s \in V^m} z^{(i-1)}_{v^m_s} = 1 \\ -1, & \text{if } \sum_{v^m_s \in V^m} z^i_{v^m_s} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

Equation (19) refers to the flow conservation conditions, which denote that the net flow to a node is zero, except for the source node n^p_{i-1} and the sink node n^p_i.

We can prove the NP-hardness of this problem by restricting certain aspects of the original problems. We apply the restriction that $r(t^m_s) = 1$ ($\forall d_m \in D^N$), which means that SFCs will be reliable all the time. Then our problem is reduced to NFV location problem which is proved to be NP-hard in [17]. Therefore, a heuristic is proposed to solve the problem.

IV. ALGORITHM DESIGN

The **C**ost-aware and **R**eliability-guaranteed **SF**C **O**rchestration (CRSO) is proposed to solve the problem. Algorithm 1 is the overview of CRSO:

- Step 1: To make the problem tractable, CRSO deals with the off-line SFCs sequentially in the descending order of the SFC length (line 3).

Algorithm 2 SFC Orchestration

```
Require: $G^N = \{D^N, E^N\}$, $\boldsymbol{S^p} = \{n_1^p, n_2^p, ...n_K^p\}$, observed sequence $\boldsymbol{O}^p = \{t_1^p, t_2^p, ...t_K^p\}$, three-tuple $(\Pi, A, B)$
Ensure: Hidden state sequence $\boldsymbol{Q^p = \{Q_1^p, Q_2^p ... Q_K^p\}}$
1:  while $1 \leq i \leq K$ do
2:      if $i == 1$ then
3:          for $\forall d_m \in D^N$ do
4:              calculate $Pr(W_m^1) = Pr(t_1^p|W_m^1) \cdot \pi_m$
5:          end for
6:      end if
7:      if $1 < i \leq K$ then
8:          for $\forall d_m \in D^N$ do
9:              $Pr(W_m^i) = max_{d_x \in D^N} Pr(t_i^p|W_m^i) \cdot Pr(W_m^i|W_x^{i-1}) \cdot Pr(W_x^{i-1})$
10:             $Q_{i-1}^p \leftarrow d_x$, $d_x$ is the DC used to get $Pr(W_m^i)$
11:             Store $Q_{i-1}^p$ in $\boldsymbol{Q^p}$
12:         end for
13:     end if
14: end while
15: $Q_K^p \leftarrow d_m = argmax_{d_m \in D^N}(Pr(d_m|n_K^p))$
16: insert $Q_K^p$ in $\boldsymbol{Q^p}$
```

- Step 2: CRSO finds a cost-efficient path of the selected $\boldsymbol{S^p}$ using **Module 1: SFC Orchestration** and put the orchestration result in $\boldsymbol{Q^p}$ (line 4).
- Step 3: Deploy all the SFCs in $\boldsymbol{SFC}$ and puts the orchestration results in $\boldsymbol{Q}$ (5-7).
- Step 4: Select the SFCs that do not satisfy the reliability requirements and put them in $\boldsymbol{eSFC}$ (line 9-10).
- Step 5: Backup VNF instances using **Module 2: VNF Backup** to meet the reliability requirements (line 11).

A. Module 1: SFC Orchestration

We model the problem of cost-aware SFC orchestration into a Hidden Markov Model (HMM) [18]. Choosing data center d_m to serve n_i^p can be regarded as hidden state W_m^i. There are several factors which can be directly observed and dependent on the hidden states, e.g., the inter-DC bandwidth cost, the VNF cost and the reliability of VNFs. These factors can be used to construct the three-tuple (Π, A, B) of HMM, where Π is the set of initial probabilities, A is the transition probability matrix, and B is the emission probability matrix.

1) Initial probabilities:

$\Pi = \{\pi_1, \pi_2, ...\pi_M\}$ is the set of initial probabilities. π_m is the probability of choosing d_m to serve the first NF n_1^p, it can be calculated according to Equation (20).

$$\pi_m = \frac{\sum_{m\in[1,M]}(\lambda_m^1 + P_{\sigma m}^{01}) - (\lambda_m^1 + P_{\sigma m}^{01})}{(M-1)\sum_{m\in[1,M]}(\lambda_m^1 + P_{\sigma m}^{01})} \tag{20}$$

To determine π_m, we consider the cost for deploying n_1^p in d_m denoted as λ_m^1 and the inter-DC bandwidth cost referred as $P_{\sigma m}^{01}$. λ_m^1 can be calculated according to Equation (7). $P_{\sigma m}^{01}$ denotes the inter-DC bandwidth cost from starting location σ^p to data center d_m. $P_{\sigma m}^{01}$ can be calculated based on Equation (2). It should be noted that $\pi_m = 0$ if $d_m \notin \boldsymbol{L_1^p}$.

2) Transition probability matrix:

A is the transition probability and it can be calculated as follow:

$$A = \begin{bmatrix} Pr(W_1^i|W_1^{i-1}) & \cdots & Pr(W_1^i|W_M^{i-1}) \\ Pr(W_2^i|W_1^{i-1}) & \cdots & Pr(W_2^i|W_M^{i-1}) \\ \vdots & \vdots & \vdots \\ Pr(W_M^i|W_1^{i-1}) & \cdots & Pr(W_M^i|W_M^{i-1}) \end{bmatrix} \tag{21}$$

$Pr(W_m^i|W_n^i - 1)$ denotes the transition probability from state W_n^{i-1} to state W_m^i. It denotes the probability that the former n_{i-1}^p is deployed in d_m and the latter n_i^p is deployed in d_n. For the transition probability $Pr(W_m^i|W_n^{i-1})$, we mainly consider the inter-DC bandwidth cost $P_{nm}^{(i-1)i}$ calculated by Equation (2). Accordingly, the transition probability $Pr(W_m^i|W_n^{i-1})$ can be calculated as follow:

$$\begin{aligned} Pr(W_m^i|W_n^{i-1}) &= \frac{\sum_{n\in[1,M]} P_{nm}^{(i-1)i} - P_{nm}^{(i-1)i}}{(M-1)\sum_{n\in[1,M]} P_{nm}^{(i-1)i}} \\ &= \frac{\sum_{n\in[1,M]} \beta_{(i-1)i}^p \cdot \eta_{nm}^N - \beta_{(i-1)i}^p \cdot \eta_{nm}^N}{(M-1)\sum_{n\in[1,M]} \beta_{(i-1)i}^p \cdot \eta_{nm}^N} \\ &= \frac{\sum_{n\in[1,M]} \eta_{nm}^N - \eta_{nm}^N}{(M-1)\sum_{n\in[1,M]} \eta_{nm}^N} \end{aligned} \tag{22}$$

Therefore, the transition probability is related to the peering link cost η_{nm}^N between data centers.

3) Emission probability matrix:

B denotes the emission probability matrix. As S^p must go through the NFs in a predefined order, the observed sequence of S^p can be denoted as $O^p = \{t_1^p, t_2^p, ...t_K^p\}$. We use $Pr(t_i^p|W_m^i)$ to represent the emission probability of observing t_i^p under the condition W_m^i. In this model, there are two observed states, one is data center d_m is suitable for node n_i^p (referred as O_i^{yes}), the other is not suitable (represented by O_i^{no}).

$$B = \begin{bmatrix} Pr(O_i^{yes}|W_1^i) & \cdots & Pr(O_i^{yes}|W_M^i) \\ Pr(O_i^{no}|W_1^i) & \cdots & Pr(O_i^{no}|W_M^i) \end{bmatrix} \tag{23}$$

$$Pr(O_i^{yes}|W_m^i) = Pr(t_i^p|W_m^i) = e^{-\lambda_{v_s^m}^i} \cdot r(t_s^m) \tag{24}$$

$$Pr(O_i^{no}|W_m^i) = 1 - Pr(O_i^{yes}|W_m^i) \tag{25}$$

In Equation (24), $\lambda_{v_s^m}^i$ denotes the cost for deploying n_i^p in v_s^m, and $r(t_s^m)$ denotes the reliability of v_s^m. We consider the VNF cost and reliability while calculating the emission probability $Pr(O_i^{yes}|W_m^i)$. Therefore, the data center with less VNF cost and higher VNF reliability is more likely to be chosen to place node n_i^p. Moreover, as n_i^p has a candidate deployment location $\boldsymbol{L_i^p}$, n_i^p cannot be deployed in data centers out of set $\boldsymbol{L_i^p}$. Therefore, $Pr(O_i^{yes}|W_m^i) = 0$ if $d_m \notin \boldsymbol{L_i^p}$.

After determining the three-tuple (Π, A, B) of HMM, the most-likely sequence of hidden states can be predicted by Algorithm 2 which is based on Viterbi algorithm [19]. From line 2-6, we calculate the initial probability $Pr(W_m^1)$ of the hidden state W_m^1. Line 7-10 calculates the most probable hidden state that is the state with maximum $Pr(W_m^i)$. In line 11, we store d_x which is used to get the maximum $Pr(W_m^i)$

into $\boldsymbol{Q^p}$. The last node n_K^p is also deployed into d_m with the maximum hidden state probability according to line 15. Therefore, we orchestrate SFCs in inter-DC network cost-efficiently.

B. Module 2: VNF Backup

After section 4.1, all the SFCs $\boldsymbol{SFC = \{S^1, S^2, ...S^P\}}$ are orchestrated in G^N. Although VNFs with higher reliability is expected in HMM, it is still hard to satisfy the reliability requirement of every SFC. We put all the SFCs that do not satisfy the reliability requirement in set $\boldsymbol{eSFC}$ ($q = |\boldsymbol{eSFC}|$). Module 2 tries to backup VNFs with less cost to satisfy the reliability requirements requested by subscribers. A Cost-Efficient Measure (CEM) is firstly defined to evaluate the importance of VNFs.

$$CEM_{v_s^m} = \frac{e^{\prod_{p\in[1,q]} RI_{v_s^m}^p - 1}}{-e^{-\rho(t_s^m)\times\alpha_m} + 1} \tag{26}$$

$$RI_{v_s^m}^p = \begin{cases} \frac{R^p}{\phi^p} = \frac{\prod_{i\in[1,K]} r_{v_s^m}^i}{\phi^p} & R^p < \phi^p \\ 1 & R^p \geq \phi^p \end{cases} \tag{27}$$

$CEM_{v_s^m}$ in Eq.(26) evaluates the importance of v_s^m in the forwarding graph. Thereinto, $RI_{v_s^m}^p$ in the numerator of $CEM_{v_s^m}$ denotes the Reliability Increment for S^p after backing up v_s^m. $RI_{v_s^m}^p$ can be calculated by Eq.(27). $RI_{v_s^m}^p$ reaches to 1 if the reliability of S^p is greater than or equal to its reliability requirement ϕ^p. It means that $RI_{v_s^m}^p$ will not increase if the reliability requirement of $\boldsymbol{S^p}$ has been satisfied. The definition of $RI_{v_s^m}^p$ helps to decrease redundant backups. The total reliability increment is evaluated by $e^{\prod_{p\in[1,P]} RI_{v_s^m}^p - 1}$. It reaches to the maximum value 1 when all the reliability requirements are satisfied. $\rho(t_s^m) \times \alpha_m$ describes the cost for backing up v_s^m. CEM value is designed to gain higher cost-efficiency while selecting the backup VNFs. Bigger CEM means higher reliability increment (numerator) and lower backup cost (denominator).

Our backup strategy is based on greedy-algorithm. Algorithm 3 is the proposed VNF backup plan. From line 1-7, we calculate the reliabilities of SFCs after being deployed, and put the SFCs that do not meet the reliability requirements into $\boldsymbol{eSFC}$. From line 9-14, backup the VNF with maximum CEM iteratively until all the reliability requirements are satisfied.

V. SIMULATION RESULTS AND ANALYSIS

A. Simulation Setup

Inter-DC network: The substrate network is selected from the public topology-zoo which is the most accurate large-scale collection of network topologies available [20]. The bandwidth capacity β_{mn}^N is set between [100, 200] Gbps, which is used in Juniper packet optical DCI solution [21]. The peering link cost η_{mn}^N is uniformly distributed between [0.01, 0.02] \$/Mbps. The unit price α_m of IT resource is between [0.05, 0.10] \$/unit. η_{mn}^N and α_m are set according to the discussion in [22] [8]. The IT resource $\rho(t_s^m)$ is uniformly between [1, 3]. The reliability of each mapped VNF $r(t_s^m)$ is randomly distributed within [0.9, 0.99] [15].

Algorithm 3 VNF Backup

```
Require: Orchestration results $\boldsymbol{Q^p = \{Q_1^p, Q_2^p ... Q_K^p\}}$
Ensure: Redundancy plan $\psi_{v_s^m}$
 1: for $1 \leq p \leq P$ do
 2:     Calculate $R^p = \prod_{i\in[1,K]} r(n_i^p)$
 3:     if $R^p \leq \phi^p$ then
 4:         Put $v_s^m$ along with $\boldsymbol{S^p}$ in set $\boldsymbol{eVNF}$
 5:         Put $\boldsymbol{S^p}$ in set $\boldsymbol{eSFC}$
 6:     end if
 7: end for
 8: while ($\boldsymbol{eSFC} \neq NULL$) do
 9:     $\forall v_s^m$ in $\boldsymbol{eVNF}$, backup $v_s^m$ with maximum $CEM_{v_s^m}$
10:     $\psi_{v_s^m} \leftarrow 1$
11:     calculate $\forall R^p \in \boldsymbol{eSFC}$
12:     if $R^p > \phi^p$ then
13:         Delete $\boldsymbol{S^P}$ from $\boldsymbol{eSFC}$
14:     end if
15: end while
```

TABLE II
EVALUATION PARAMETERS

Inter-DC Network	
bandwidth capacityβ_{mn}^N	[100, 200] Gbps
peering link cost η_{mn}^N	[0.01, 0.02] \$/Mbps
number of function type	[4, 8]
IT resource capacity $\rho(t_s^m)$	[1, 3] units
VNF reliability $r(t_s^m)$	[0.9, 0.99]
unit price of IT resource α_m	[0.05, 0.1] \$/unit
Service Function Chain	
length of a SFC	[2, 6]
requested bandwidth $\beta_{(i-1)i}^p$	[10, 100] Mbps
requested resource ρ_i^p	[0.4, 1] units
requested reliability ϕ^p	[0.95, 0.98, 0.99, 0.995, 0.999]

SFC requests: We set the length of a SFC uniformly distributed between 2 to 6 [15]. The bandwidth of a SFC $\beta_{(i-1)i}^p$ is uniformly distributed between [10, 100] Mbps [23]. In the light of the SLA requirement of Google Apps, the reliability requirement ϕ^p is selected among [0.95, 0.98, 0.99, 0.995, 0.999] [24]. Each type of NF requests for IT resource within [0.4,1] units (based on the results in [22]).

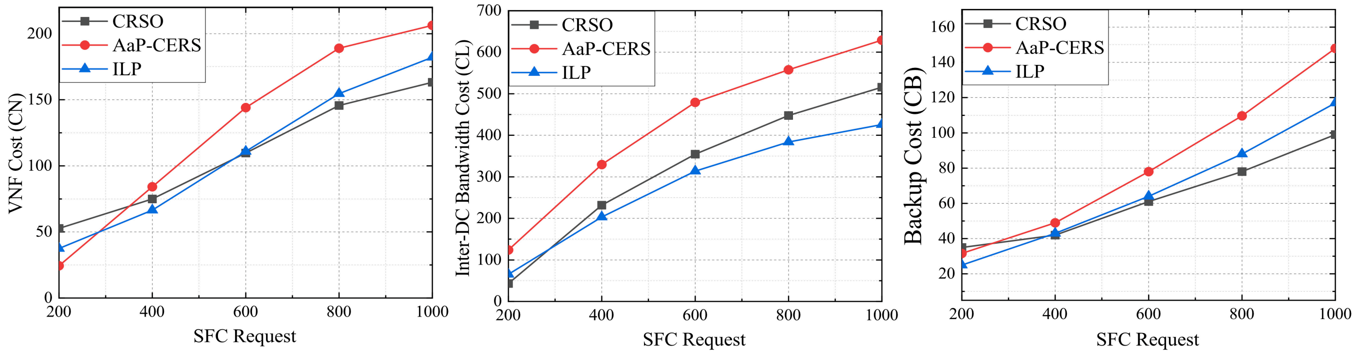
Comparison Algorithms: We implement our algorithm in Python 2.7. All the experiments were conducted on a machine with an Intel E3-1220 processor and DDR4 2133MHz memory. We compare our algorithm with AaP-CERS and ILP. AaP-CERS orchestrates the incoming SFCs with affiliation-aware vNF placement (AaP) proposed in [25] and backup the VNFs with cost-efficient redundancy scheme (CERS) proposed in [15]. ILP is solved by CPLEX.

B. Simulation and Results

1) Observing on Cost:

We firstly evaluate the orchestration cost produced by mapping SFC requests ranging from 200 to 1000. As the total cost contains three parts (inter-DC bandwidth, VNF and backup cost), we observe them one by one.

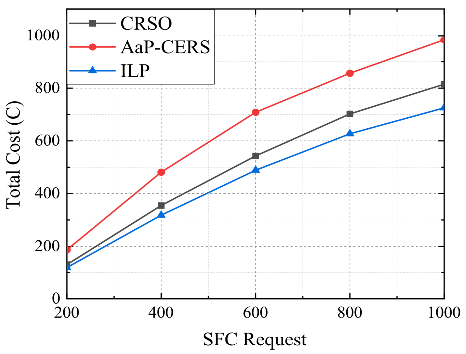
VNF cost: Figure 3(a) depicts the VNF cost with various SFC requests. When the number of SFCs is smaller than 400, AaP-CERS employs smaller VNF instances to deal with



(a) VNF Cost

(b) Inter-DC Bandwidth Cost

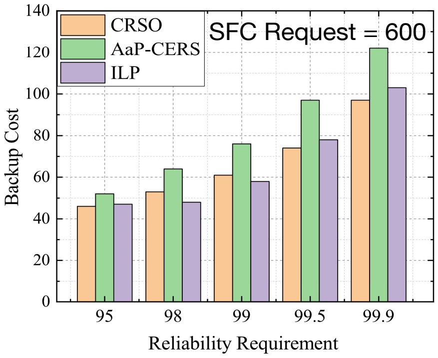
(c) Backup Cost



(d) Total Cost

Number of SFC Requests	Average Running Time (Seconds)		
	ILP	CRSO	AaP_CERS
200	1.04e+4	3.14e-2	2.89e-2
400	2.26e+4	4.97e-2	6.22e-2
600	3.52e+4	6.32e-2	7.45e-2
800	4.36e+4	8.41e-2	9.33e-2
1000	6.12e+4	9.73e-2	10.81e-2

(e) Running Time



(f) Backup Cost with Various Reliability Requirements

Figure 3. (a) VNF cost (b) inter-DC bandwidth cost (c) backup cost (d) total cost (e) running time comparison (f) backup cost with different reliability requirements

the incoming SFC requests. However, AaP-CERS increases significantly when the number is bigger than 600. This is because AaP considers the affiliation of the requested SFCs and merges the SFCs into graph(s) to increase the sharing of VNFs. However, when there are more SFC requests, it is difficult for AaP-CERS to merge SFCs into optimal service function graph(s) which satisfies the location constraints of the NF request. CRSO uses more VNFs in small-scale situation. CRSO deals with the SFCs with HMM in descending order of their length. We ignore other the affinity of SFCs to decrease the problem complexity. This will lead to more VNF cost especially in the small scale situation. When there are more SFCs, our proposed CRSO employs comparable VNFs compared with ILP.

Inter-DC bandwidth cost: Figure 3(b) shows the inter-DC bandwidth cost. It is conspicuous that AaP-CERS orchestrates SFCs in inter-DC network with more bandwidth. AaP firstly merges the SFC into graph(s) to decrease the usage of VNF instances, however this will lead to higher bandwidth consumption. CRSO considers both bandwidth and VNF cost, it uses about 26.2% less bandwidth cost compared with AaP-CERS. The bandwidth cost of CRSO is comparable to ILP when the number of SFCs is smaller than 400. When the number is bigger than 600, CRSO uses about 11.3% more inter-DC bandwidth cost compared with ILP. The definition of emission probability of CRSO considers both the VNF cost and the reliability. Therefore, CRSO sacrifices part of the cost for higher SFC reliability.

Backup cost: Figure 3(c) depicts the backup cost with SFCs ranging from 200 to 1000. AaP-CERS uses more backup nodes especially when the SFC request is bigger than 600. The cost importance measure defined in CERS only considers the reliability increment and ignores the reliability requirement requested by users. This will produce redundant backups leading to more backup costs. CRSO uses comparable backup nodes compared with ILP when the SFC request is smaller than 600. When the number is bigger than 800, CRSO uses 13.2% less backup cost. This is because CRSO considers the reliability requirements while orchestrating SFCs using HMM.

Total cost: The total cost is depicted in Figure 3(d). It is the sum of VNF cost depicted in Figure 3(a), the inter-DC bandwidth cost illustrated in Figure 3(b) and the backup cost shown in Figure 3(c). Obviously, AaP-CERS consumes more cost compared with CRSO and ILP. AaP-CERS uses about 20.4% and 15.6% more cost compared with ILP and CRSO respectively.

2) Observing on Execution Time:

Figure 3(e) depicts the execution time of different algorithms with SFC requests ranging from 200 to 1000. Obviously ILP orchestrates SFCs using much more running time. The average execution time of ILP is at least six times longer than CRSO and AaP-CERS. Heuristic CRSO and AaP-CERS use comparable execution time to accomplish the service function chaining. Therefore, it is difficult to calculate the optimal solution for large-scale SFC requests with ILP in a tolerable time.

3) Observing on Backup Cost:

Figure 3(f) fixes the number of SFCs and evaluates the backup cost produced by mapping SFC requests with different reliability requirements. Obviously, backup cost increases significantly with higher reliability requirement. AaP-CERS uses more backup nodes to satisfy the reliability requirements. When the reliability of SFCs is higher than 99%, AaP-CERS uses about 28.5% and 24.2% more cost compared CRSO and ILP respectively. CRSO uses comparable backup cost compared with ILP. However, when the reliability requirement is higher than 99.5%, the backup cost of CRSO is a little bit lower than ILP. CRSO sacrifices part of the inter-DC bandwidth and VNF cost to achieve higher reliability during the SFC orchestration. Therefore, the backup cost for CRSO is a bit lower than ILP especially in large-scale situation or higher reliability requirements.

VI. CONCLUSION

In this paper, we orchestrate SFCs in an inter-DC network cost-efficiently, while providing economical redundancy method to guarantee the reliability requirements of the subscribers. The problem is formulated into an integer linear programming program. Then we design a heuristic algorithm named CRSO to solve it. CRSO contains two modules. "Module 1: SFC Orchestration" realizes cost-aware SFC chaining based on HMM. "Module 2:VNF Backup" accomplishes reliability-guaranteed VNF backup based on a designed cost-efficient measure (CEM). CRSO backups the VNF with the highest CEM until all the reliability requirements are satisfied. Experimental results show that GRSO uses about 20.4% less cost compared with the existing algorithm for accommodating the same service function chain requests.

This work was supported by the National Natural Science Foundation of China (61501044).

REFERENCES

[1] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.

[2] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, Feb 2015.

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

[4] H. Chen, S. Xu, X. Wang, Y. Zhao, K. Li, Y. Wang, W. Wang, and L. M. Li, "Towards optimal outsourcing of service function chain across multiple clouds," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[5] W. Fang, M. Zeng, X. Liu, W. Lu, and Z. Zhu, "Joint spectrum and it resource allocation for efficient vnf service chaining in inter-datacenter elastic optical networks," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1539–1542, 2016.

[6] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem:network processing as a cloud service," *Acm Sigcomm Computer Communication Review*, vol. 42, no. 4, pp. 13–24, 2012.

[7] G. Gibb, H. Zeng, and N. Mckeown, "Outsourcing network functionality," in *The Workshop on Hot Topics in Software Defined Networks*, 2012, pp. 73–78.

[8] D. Dietrich, A. Abujoda, and P. Papadimitriou, "Network service embedding across multiple providers with nestor," in *2015 IFIP Networking Conference (IFIP Networking)*, May 2015, pp. 1–9.

[9] J. Kong, I. Kim, X. Wang, Q. Zhang, H. C. Cankaya, W. Xie, T. Ikeuchi, and J. P. Jue, "Guaranteed-availability network function virtualization with network protection and vnf replication," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[10] Q. Long, C. Assi, K. Shaban, and M. Khabbaz, "Reliability-aware service provisioning in nfv-enabled enterprise datacenter networks," in *International Conference on Network and Service Management*, 2017.

[11] Q. Zhang, X. Wang, I. Kim, P. Palacharla, and T. Ikeuchi, "Vertex-centric computation of service function chains in multi-domain networks," in *Netsoft Conference and Workshops*, 2016, pp. 211–218.

[12] Q. Xu, D. Gao, and T. Li, "Low latency security function chain embedding across multiple domains," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2018.

[13] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "Grep:guaranteeing reliability with enhanced protection in nfv," in *ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, 2015, pp. 13–18.

[14] L. Qu, C. Assi, K. B. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in nfv-enabled enterprise datacenter networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 554–568, 2017.

[15] W. Ding, H. Yu, and S. Luo, "Enhancing the reliability of services in nfv with the cost-efficient redundancy scheme," in *IEEE International Conference on Communications*, 2017, pp. 1–6.

[16] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 98–106.

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

[18] L. R. Rabiner, "A tutorial on hidden markov model and selected applications in speech recognition," *Readings in Speech Recognition*, vol. 77, no. 2, p. 267296, 1990.

[19] M. S. Ryan and G. R. Nudd, "The viterbi algorithm," *Proc IEEE*, vol. 61, no. 5, pp. 268–278, 1973.

[20] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[21] I. Juniper Networks, "Juniper networks packet optical data center interconnect solution," https://www.juniper.net/assets/jp/jp/local/pdf/solutionbriefs/3510590-en.pdf, august, 2016.

[22] "Amazon web services," https://aws.amazon.com/cn/ec2/pricing/.

[23] H. Chen, X. Wang, Y. Zhao, T. Song, Y. Wang, S. Xu, and L. Li, "Mosc: a method to assign the outsourcing of service function chain across multiple clouds ," *Computer Networks*, vol. 133, pp. 166–182, 2018.

[24] "Google apps service level agreement." https://gsuite.google.com/intl/en/terms/sla.html.

[25] Q. Sun, P. Lu, W. Lu, and Z. Zhu, "Forecast-assisted nfv service chain deployment based on affiliation-aware vnf placement," in *Global Communications Conference*, 2017.