ASCO: An Availability-aware Service Chain Orchestration

[1]WenChen He, [2]Xiong Chen, [1]XueSong Qiu, [1]ShaoYong Guo, [1]Peng Yu
[1]*State Key Laboratory of Networking and Switching Technology*
[1]*Beijing University of Posts and Telecommunications, Beijing, China*
[2]*Nanrui Group Corporation/State Power Nanrui Technology Co., Ltd, Nanjing, China*
Email: wche@bupt.edu.cn, chenxiong@sgepri.sgcc.com.cn, xsqiu@bupt.edu.cn, syguo@bupt.edu.cn, yupeng_bupt@126.com

***Abstract*—Network Functions (NFs) are decoupled from dedicated hardware equipment and present a better shareability of physical resources on basis of Network Function Virtualization (NFV) technology. The placement and routing of VNFs are orchestrated flexibly by operators for providing differentiated services to users. QoS parameters such as delay, service availability and resource utilization efficiency need to be taken into account. In this paper, an availability-aware service chain orchestration (ASCO) mechanism including pre-pruned depth-first search (PDFS) algorithm and M-to-N backup mechanism is proposed. PDFS is developed from depth-first search algorithm by taking delay as pruning factor. An end-to-end reachable and cost-optimized path set is obtained and extended to final orchestration path through adopting of PDFS. Moreover, M-to-N backup mode allows one of M primary VNFs to have N migration options. Load balancing factor is designed to achieve network traffic balance by selecting path with more remaining bandwidth. The simulation result shows that our algorithm can meet QoS requirements and service availability in a low-cost way.**

***Index Terms*—NFV, availability, QoS, service chain, orchestration**

I. INTRODUCTION

In traditional service provisioning architecture, network functions and hardware devices are coupled to form a vertical integrated network structure which is rigid and not conducive to sharing resources and flexibly deploying services. NFV utilizes IT virtualization technology to achieve various software-based network functions by industrial standard high-capacity servers, memories and switches. This technology improves unification and generalization of network devices while realizes flexible deployment and instantiation of network function software, making service management and orchestration available for network operators [1-3].

In the process of wide services offered by network operators, the Service Level Agreements (SLAs) formalize the service grade details promised to a particular user. The SLAs are various among operators, but they generally contain QoS parameters such as minimum guaranteed bit rate, maximum delay and packet loss [4-7]. Additionally, the availability of service affected by the failures of VNFs is another crucial parameter. The breakdown of physical machines (PM), virtual machines (VM) and software will cause failures of VNFs and further affect normal execution of entire service chain, resulting in serious data loss and resources waste. Redundancy backup is a common method to improve availability. However, this method may lead to extra overhead including resource consumption and end-to-end transmission delay of services, which is unacceptable for delay-sensitive services. Therefore, it is urgent to efficiently meet QoS requirements and service availability with limited resources. Authors in [8] propose two integer linear programming models for service chain orchestration to efficiently provision the service requests. High complexity makes those models not suitable for large-scale computing in practical application. Authors in [9] propose to select the most important VNF to back up based on the classical Birnbaum importance measure. But the QoS requirements of service after being backed up will change in terms of resource consumption and end-to-end delay, which needs to be taken into consideration.

According to above analyses, this paper proposes an availability-aware service chain orchestration mechanism. Within QoS we improve depth-first search algorithm by taking delay as pruning factor and then obtain the orchestration scheme including placement and routing of VNFs by a pre-pruned depth-first search (PDFS). As for service availability, M network functions are allowed to share N backups to improve availability and resource utilization. Meanwhile, the traffic balancing of links must be taken into consideration to avoid congestion caused by excessive load.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System model

As shown in Fig.1, service chain includes a set of ordered VNFs installed in virtual machines. When a single VNF fails, it can migrate to the backup VNF. For instance, video streaming services are implemented by traversing Network Address Translation (NAT), Firewall (FW), Video Optimization Controller (VOC) and Intrusion Detection System (IDS). When a virtual FW fails, service can be completed by migrating to backup FW.

The substrate physical network is represented by a weighted undirected graph $G = (N, L)$, where N and L are sets of physical nodes and links respectively. For any server node n_i, its capacity is $cap(n_i)$, which represents physical resources such as CPU, memory resource. For any physical link $l_{ij} \in L$ connecting n_i to n_j, its bandwidth is b_{ij}, and transmission

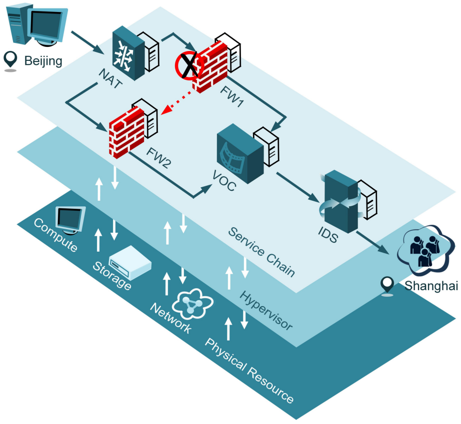

Fig. 1. System model.

delay is d_{ij}. $V = \{v_1, v_2, ..., v_K\}$ represents a set of VNFs, and K represents the number of VNFs. $S = \{s_1, s_2, ..., s_i\}$ represents a set of service chains, $K(s_i)$ represents the number of needed VNFs in s_i. VNF v_i consumes physical resources $cap(v_i)$. v_i has processing delay d_i and availability a_i. Similarly, virtual link l^v_{uv} between VNFs is mapped to physical link l_{ij}. A service has three attributes: minimum bandwidth requirement B_{req}, minimum end-to-end delay requirement D_{req}, and minimum availability requirement A_{req}.

B. Service chain optimal orchestration model

This paper defines three binary variables to describe service chain orchestration.

1) $x_{i,j}$: $x_{i,j}$=1 indicates that v_j is backed up by v_i. Otherwise, $x_{i,j}$=0;

2) $y_{ij,uv}$: $y_{ij,uv}$=1 indicates that l^v_{uv} is mapped to l_{ij}. Otherwise, $y_{ij,uv}$=0;

3) $z_{i,j}$: $z_{i,j}$=1 indicates that v_j is mapped to n_i. Otherwise, $z_{i,j}$=0.

VNFs consume CPU and memory resources. The unit price of server resources is denoted by c_1. Therefore, related cost of installing VNFs is given by

$$cost(VNF){=}c_1 \sum_{i\in N}\sum_{j\in V} z_{i,j}cap(v_j) \tag{1}$$

A load balancing factor Θ_{ij} of l_{ij} is defined to indicate load status of link. Its value is inversely proportional to the remaining bandwidth. This factor is given by

$$\begin{aligned}\Theta_{ij} &= \frac{\alpha}{b^{ij}_{remain} + \beta} + \gamma \\ &= \frac{\alpha}{b_{ij} - \sum\limits_{i,j\in N}\sum\limits_{u,v\in V} B^{ij}_{req} y_{ij,uv} + \beta} + \gamma, \forall u, v\end{aligned} \tag{2}$$

Where α, β, γ is a set of adjustment factors. c_2 indicates unit price of bandwidth. Therefore, the cost of bandwidth is given by

$$cost(bandwidth) = c_2 \sum_{i,j\in N}\sum_{u,v\in V} y_{ij,uv} B^{ij}_{req} \Theta_{ij} \tag{3}$$

It can be seen from (3) that links with larger remaining bandwidth resources have relatively lower cost.

Node and link constraints:

The server has limited resources and cannot continue to host VNFs when resources are occupied.

$$C_1 : \sum_{j\in V} z_{i,j}cap(v_j) \le cap(n_i), \forall i \tag{4}$$

Similarly, physical links must have enough remaining bandwidth resource.

$$C_2 : \sum_{u,v\in V} y_{ij,uv} B^{ij}_{req} \le b_{ij}, \forall i, j \tag{5}$$

Availability constraint: Original service chain without backups has minimum availability requirement A_{req}. The service availability will be improved after backing one certain primary VNF up. The backup mechanism is finally completed when availability A_s of s_i meets requirement A_{req}.

$$C_3 : A_s = \prod a_i \ge A_{req} \tag{6}$$

QoS constraint: This paper considers services with requirement of minimum delay D_{req}. The end-to-end delay mainly includes processing delay on VNF and the transmission delay on links. Therefore, we should ensure that the new chain still meets delay requirements of service.

$$\begin{aligned}C_4 : D = &\sum_{j\in V}\sum_{i\in N} z_{i,j} d_j + \sum_{u,v\in V}\sum_{i,j\in N} y_{ij,uv} d_{ij} \\ &\le D_{req}\end{aligned} \tag{7}$$

In summary, the orchestration model involves optimization objective function and above constraints. It is given by

$$\begin{aligned}&\min\{cost(VNF) + cost(bandwidth)\} \\ &s.t. \begin{cases} C_1, ..., C_4 \\ C_5 : x_{i,j}, y_{ij,uv}, z_{i,j} \in \{0, 1\} \quad \forall i, j, u, v \end{cases}\end{aligned} \tag{8}$$

III. ALGORITHM DESCRIPTION

ASCO completes service orchestration in a cost-optimal manner as well as guarantees QoS and availability requirements. Firstly, for current service chain s_i, a set of original end-to-end reachable paths *Original_set* is obtained by PDFS. Secondly, the above paths is classified into ($K(s_i)$+1) classes. Class i indicates that there are already existing available i VNFs. Then, ASCO selects the lowest cost paths in each class to form a optional path set *Optional_set*. In the following, paths in optional path set need to be expanded to $K(s_i)$ VNFs to complete service chain s_i. At last, ASCO selects a path with the lowest cost as final orchestration scheme. The procedure is illustrated in Alg.1.

After service orchestration, ASCO needs to back current service chain up to meet availability requirements according to $A_i < A_{req}$. It selects VNFs with the lowest availability a_{lowest} in this chain, which can maximize backup efficiency. In the process of backing up, algorithm firstly selects the existing and available backup for primary VNF under the

Algorithm 1 Availability-aware Service Chain Orchestration Mechanism (ASCO)

```
Require: Substrate physical network $G$; Virtual network function $V$; Service chain $S$; QoS;
Ensure: Orchestration scheme;
for service chain $s_i$ do
    Start from physical endpoints $n_s$, $n_t$;
    Get reachable paths by PDFS($\Omega$, $n_s$, $n_t$);
    Classify paths based on the number of existing VNFs;
    Select the lowest-cost paths from each class;
    Extend paths in Optional_set
    for $path \in Optional\_set$ do
        for $n_j \in path$ do
            Install required VNF $v_r$ in $n_j$;
            Calculate $cost(VNF)$;
        end for
        Select scheme with lowest cost;
        if No server nodes available on $path$ then
            for $n_j \notin path$ do
                Install required VNF $v_r$ in $n_j$;
                Extend $path$ to $n_j$ by PDFS;
                Calculate $cost(VNF)+cost(bandwidth)$;
            end for
            Select scheme with lowest cost;
        end if
    end for
    while $A_i < A_{req}$ do
        Bestbackups= M-to-N($s_i$);
    end while
end for
return Orchestration and backup scheme;
```

condition of meeting delay requirement. Through the above selection, one primary VNF may have multiple backups, and one backup also may support multiple primary VNFs. When no backup is available, ASCO will select a server node with sufficient resources to create a new backup. ASCO calculates $A_i < A_{req}$ of current service chain in real time, and ends when the availability meets requirements.

The delay is designed as a pruning factor to be added to the depth-first search algorithm. Therefore, we propose PDFS to solve routing problem among network nodes. PDFS uses Boolean value Ω to pre-prune the searching branches that have not met delay requirement. Definition of Ω is given by

$$\Omega = (\sum_{j \in V} \sum_{i \in N} z_{i,j} d_j + \sum_{i,j \in N} \sum_{u,v \in V} y_{ij,uv} d_{ij} \geqslant D_{req}) \tag{9}$$

$\Omega = 1$ indicates that delay has exceeded delay requirement of service when current path reaches this node. When PDFS is searching current node, it calculates delay of current routing scheme. If $\Omega = 1$, the current routing will be stopped, and return to the previous node to continue search until all paths are obtained. Finally, we calculate cost of all paths derived from PDFS and select the path with the lowest cost.

Algorithm 2 M-to-N Service Chain Backup Mechanism (M-to-N)

```
Require: Substrate physical network $G$; Virtual network function $V$; Service chain $S$; QoS;
Ensure: Backup scheme;
Back up VNF with the lowest $a_i$
Choose available backup if it already exists
for $v_b \in$ VNF candidates do
    Backuppath = PDFS($\Omega$, $n_s$, $n_b$) + PDFS($\Omega$, $n_b$, $n_t$);
    BackupCost=cost(Backuppath);
end for
if no backups available then
    for $n_b \in$ server nodes do
        Create a new backup $v_b$ in $n_b$;
        Backuppath = PDFS($\Omega$, $n_s$, $n_b$) + PDFS($\Omega$, $n_b$, $n_t$);
        BackupCost= cost(VNF) + cost(Backuppath);
    end for
end if
return Backup scheme;
```

IV. EVALUATION ANALYSIS

A. Simulation setting

A simulation network consisting of 20 nodes and 100 links is set up through Java language. The link bandwidth is randomly set between 20-30 Mbit/s. Each service chain contains 2-4 VNFs. According to requirements of SLA, availability requirements for service chains are chosen from [0.95, 0.98, 0.99, 0.995, 0.999]. In this experiment, 2500 service chains are initialized continuously and dynamically.

The following two orchestration algorithms are selected as comparisons. (1) MinCost algorithm: It uses shortest path algorithm to obtain one cost-optimal path and extends it. Algorithm selects VNF with the lowest cost to backup. (2) Single-path algorithm: Backups are installed downstream of current path to avoid additional link and delay overhead.

B. Result analysis

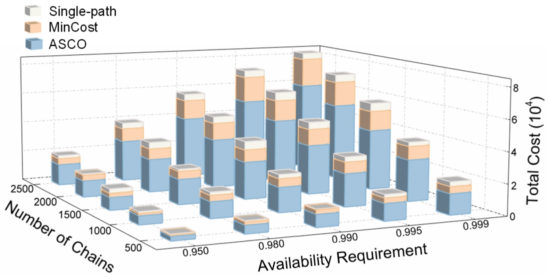

Fig. 2. Cost.

1). Cost: Fig. 2 shows cost of three algorithms. Taking availability requirement of 0.98 as an example, ASCO saves 17% and 32% cost compared to MinCost and Single-path respectively. This is because that, ASCO gets the final optimal path set by extending paths in original path set with minimum cost, instead of just expanding an original link with minimum cost. It allows M VNFs to use N backups in common and improves utilization of resources.

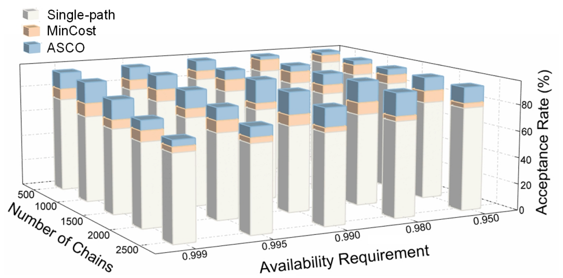

Fig. 3. Service acceptance rate.

2). Service acceptance rate: Taking 0.95 as an example, acceptance rates of ASCO, Mincost and Single-path are 93.4%, 81.2% and 77.7% respectively. ASCO uses additional delay as a constraint to orchestration scheme so that services always meet delay requirement. M-to-N backup method ensures that ASCO can use limited resources to maximize availability. Therefore, acceptance rate is always higher than the other two.

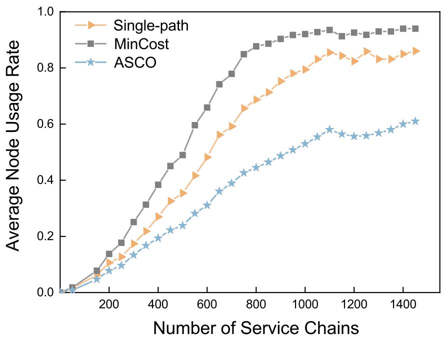

Fig. 4. Average node usage rate.

3). Average node usage rate: Fig. 4 shows average server nodes usage rate obtained by three algorithms. Compared with the other two algorithms, ASCO always deploys service chains based on the minimum cost and maintains a low resource occupancy rate, which indicates that algorithm can effectively improve resource utilization. M-to-N backup method minimizes the number of backup VNFs. These measures make average node usage rate minimum.

4). Variance of link usage rate: When the number of services reaches 1000, ASCO's variance is 17% and 65% lower than that of Mincost and Single-path, respectively. We design a load-balancing factor in path selection process and add it to calculation of link cost. Its value is inversely proportional to remaining capacity of link bandwidth. Therefore, ASCO can ensure that the network is in a balanced state.

V. CONCLUSION

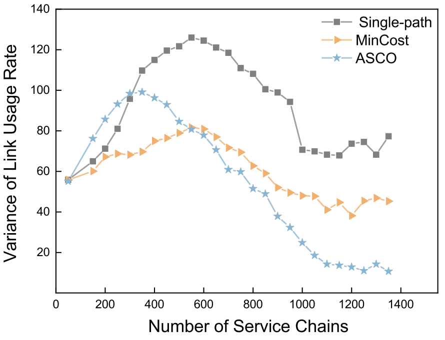

Fig. 5. Variance of link usage rate.

To complete service orchestration reasonably and efficiently, we propose an ASCO mechanism. It comprehensively considers delay and availability requirements, then completes service chain deployment with the minimum cost. We also propose an M-to-N backup scheme. It can not only maximize resource utilization, but also improve the service availability. The simulation result shows that ASCO mechanism can complete service chains orchestration with minimal cost as well as meet delay and availability requirements.

ACKNOWLEDGMENT

This paper is supported by ”Core Technology and Development of Key Equipment for Power System Stable Control based on Multi Source Real Time Data Transfer Scheduling Mode”(Project NO: SGTYHT/15-JS-191).

REFERENCES

[1] 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,” 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015, pp. 98-106.

[2] Yossi Kanizo, Ori Rottenstreich, Itai Segall and Jose Yallouz, Designing Optimal Middlebox Recovery Schemes with Performance Guarantees, IEEE Journal on Selected Areas in Communications (JSAC), 2018.

[3] 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,” 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015, pp. 98-106.

[4] Tung-Wei Kuo, Bang-Heng Liou, Kate Ching-Ju Lin, Ming-Jer Tsai, Deploying Chains of Virtual Network Functions: On the Relation Between Link and Server Usage, IEEE/ACM Transactions on Networking, vol. 26(4), pp. 1562-1576, 2018.

[5] Argawal Satyam, Malandrino Francesco, Carla Fabiana Chiasserini, De Swedes, Joint VNF Placement and CPU Allocation in 5G, IEEE INFOCOM 2018, 2018.

[6] A. Tawbeh, H. Safa and A. R. Dhaini, ”A hybrid SDN/NFV architecture for future LTE networks,” 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

[7] D. Cotroneo et al., ”Network Function Virtualization: Challenges and Directions for Reliability Assurance,” 2014 IEEE International Symposium on Software Reliability Engineering Workshops, Naples, 2014, pp. 37-42.

[8] N. Huin, B. Jaumard and F. Giroire, ”Optimization of network service chain provisioning,” 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-7.

[9] W. Ding, H. Yu and S. Luo, ”Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme,” 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.