Computer Networks 133 (2018) 166–182

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

MOSC: a method to assign the outsourcing of service function chain across multiple clouds☆

Huan Chen, Xiong Wang, Yangming Zhao, Tongyu Song, Yang Wang, Shizhong Xu*, Lemin Li

Key Laboratory of Optical Fiber Sensing and Communication, Education Ministry of China, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

A R T I C L E I N F O

Article history:
Received 3 May 2017
Revised 15 October 2017
Accepted 17 January 2018

Keywords:
Network Function Virtualization
Cloud computing
Service function chain
Hidden Markov Model

A B S T R A C T

As Network Function Virtualization (NFV) becomes reality and cloud computing offers a scalable pay-as-you-go charging model, more network operators would like to outsource their Service Function Chains (SFC) to the public clouds in order to reduce the operational cost. Unfortunately, challenges of Quality of Service guarantee still exist while minimizing the operational cost with outsourcing SFC to public clouds. In this paper, we investigate this problem when there are a large number of candidate cloud providers with the consideration of diverse pricing schemes of network functions, additional latency caused by public network, and the relationship between the Virtual Network Function (VNF) performance and its cost. Compared with our previous conference version, we design D-MOSC, an improved deviation based heuristic algorithm to assign the Outsourcing of SFC across multiple clouds based on Hidden Markov Model (HMM). The extensive simulations show that MOSC saves up to 79.2% cost compared with that of deploying network functions in the local network. MOSC also achieves up to 50.7% cost savings compared with the result of the first-fit based optimization algorithm. Compared with the greedy version, D-MOSC achieves up to 26.7% cost savings with the guarantee of latency requirements.

1. Introduction

In today's operator networks, the communication traffic should go through a series of middleboxes, such as Network Address Translation (NAT), Firewall (FW), and Intrusion Detection/Prevention System (IDS/IPS). Usually, these middleboxes should be traversed in a specific order. For example, for a secure server whose accesses need to be highly restricted, its incoming traffic may first traverse an FW and then an IDS/IPS. Such ordered middleboxes is referred as Service Function Chains (SFC) [1].

Traditionally, the SFCs is implemented by steering flows to the hardware middleboxes located in local networks that incur high deployment and maintenance cost [2]. To solve this problem, Network Function Virtualization (NFV) implements middleboxes in software running on VMs [3–5], and some technologies are proposed to face the dynamic requirements of services [6,7]. In addition, several architectures (e.g., APLOMB [2], Jingling [8]) are proposed to migrate VNFs from local networks to clouds. As public clouds usually offer a pay-as-you-go charging model and elastic service delivery, the operational cost and complexity of maintenance can be reduced. More and more operators would like to outsource the SFC to the public cloud by renting computation resources.

When the network operators make the cost-efficient outsourcing plans, there are several issues that should be addressed. First of all, there are a large number of cloud providers (e.g., Amazon, Azure, Linode, Aliyun) which have diverse pricing schemes (as shown in Table 1 [9]) and different technical specifications (e.g., cloud latency). Moreover, redirecting flows to the clouds may introduce extra delays, thus how to guarantee the Quality of Service (QoS) of flows should be considered. At last, some network functions cannot be outsourced due to the security or privacy reasons (e.g., sensitive data storage).

More broadly, RFC7498 [1] proposed the topological dependencies of Service Function Chaining. The Network Functions need to be deployed on the network path or be traversed by corresponding

☆ The preliminary version of this paper titled "Towards Optimal Outsourcing of Service Function Chain Across Multiple Clouds" was published in the proceedings of IEEE ICC 2016. In this longer version, we present the following extended work. (1) We present more background on Service Function Chaining and how the SFC-enabled domains work. (2) We discuss the motivation example with extensive simulations to show the impacts with a larger dataset. (3) We propose an improved deviation based algorithm to find tradeoff between cost and latency, and make the algorithm in the conference version as the baseline. (4) We present the evaluation of our improved algorithm with that of the greedy based algorithm in the conference version. (5) We provide more concentrate discussion on modeling our problem as a Hidden Markov Model, and discuss some new related works.

* Corresponding author.
E-mail addresses: wangxiong@uestc.edu.cn (X. Wang), xsz@uestc.edu.cn (S. Xu).

https://doi.org/10.1016/j.comnet.2018.01.020
1389-1286/

Table 1
Pricing schemes of different cloud storage providers [9].

	S3 USA, EU ($)	Rackspace ($)	Nirvanix ($)
Data transfer in (GB)	0.10	0.08	0.18
Data transfer out (GB)	0.15	0.22	0.18
Storage (GB/month)	0.15	0.15	0.25

flows with traffic steering. Therefore, determining the outsourcing plan of Service Functions Chain is not an easy task, which sometimes forces the network administrators to create complicated routing schemes or provide an alternate virtual topology to satisfy the requirements [1], these mechanisms introduce extra network delays. Moreover, the performance of VNFs on different cloud environment is different. For example, DPI may only have 1 Gbps throughput with legacy NICs, but can be 10 times faster while using DPDK (Data Plane Development Kit) accelerated version of VNF [10,11]. A cloud server equipped with SATA-3 SSD (Solid State Drive) may have extreme high speeds close to 3.2 Gbps for data operations, but the price per gigabyte is 10 times greater than legacy drives [12]. Furthermore, other metrics of the cloud performance (e.g., bandwidth, CPU share) is also tightly related to the price [13]. Despite the performance, the pricing also matters the data reliability, using full-replication roughly doubles the hosting costs [9].

In view of this, we are to investigate how to migrate the SFCs to the public clouds in order to minimize the cost. Meanwhile, we should also take the QoS into consideration, i.e. the latency requirement of each SFC should be satisfied. In order to solve the problem, we first formulate it as an Integer Linear Programming (ILP) model. As this problem is *NP-hard*, the ILP model is intractable in large size networks. Accordingly, we propose a heuristic to schedule the outsourcing of SFC across multiple clouds with QoS guarantee (MOSC). The key idea of MOSC is to model the cloud selection as hidden stats transition in Hidden Markov Model (HMM), and then leverage Viterbi algorithm [14] to derive the SFC outsourcing plan by predicting the most-likely state sequences.

The technical contributions of this paper are summarized as follows:

- We formulate the service function chain outsourcing problem as an ILP model and propose deviation algorithm based MOSC, an efficient heuristic algorithm to solve this problem.
- We compare the results of MOSC with that of ILP model in small-scale networks. The results show that MOSC achieves near-optimal results in small-scale networks.
- We conduct extensive simulations to evaluate the performance of MOSC in large-scale networks. The results show that 79.7% cost is saved compared with that of deploying VNFs in the local network. Moreover, MOSC achieves up to 50.7% cost savings compared with that of the first-fit based optimization algorithm.
- Comparing with the conference version [15], we conduct extensive simulations to evaluate the performance improvements of Deviation algorithm based MOSC (D-MOSC) to that of the previous Greedy algorithm based MOSC (G-MOSC).

This paper is organized as follows. Firstly we discuss the outsourcing problem by a motivation example in Section 2. After that, the service function chain outsourcing problem is formulated as an ILP model in Section 3. In Section 4, we design an efficient heuristic algorithm for the problem, followed by simulation results in Section 5. The related works are presented in Section 6. At last, the conclusion is given in Section 7.

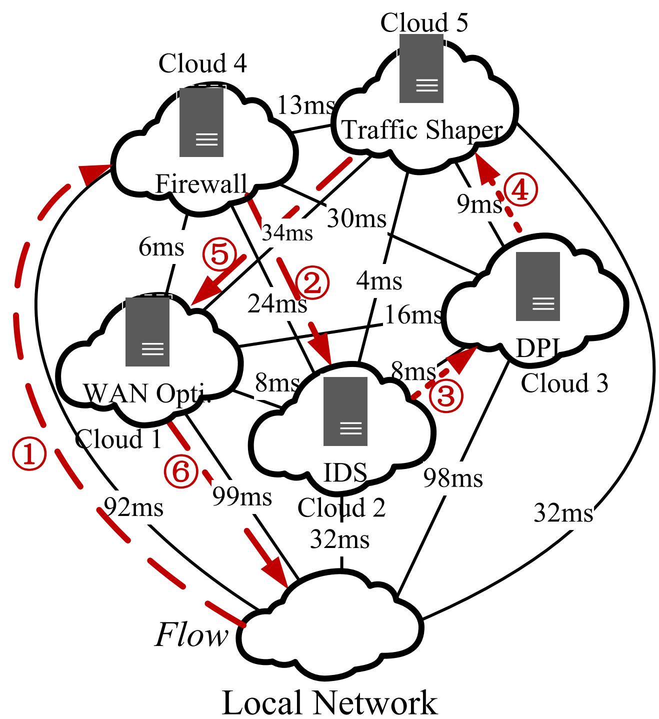

Fig. 1. Motivation example (cost efficient deployment).

Table 2
Pricing schemes of virtual network functions for flow of unit size.

	Firewall ($)	IDS ($)	DPI ($)	Traffic shaper ($)	WAN opti. ($)
Cloud 1	6	8	12	11	3
Cloud 2	13	5	28	19	24
Cloud 3	5	9	8	12	18
Cloud 4	2	9	16	10	13
Cloud 5	8	6	31	5	6
Local Network	13	18	32	22	28

2. Motivation

2.1. Motivation example

There are several works focusing on the VNF placement problem [16–20]. In this section, we explain why existing solutions are not suitable for the service function chain outsourcing problem through an example shown in Fig. 1.

We suppose that there is one flow of unit size in the local network, and the SFC of the flow is {Firewall → Intrusion Detection System (IDS) → DPI → Traffic Shaper → Wide Area Network (WAN) Optimizer}. Both the local network and public clouds can provide the five types of VNFs. The pricing schemes of the five VNFs in public clouds, as well as that in the local network, are given in Table 2.

RFC7665 [21] defined the components of core SFC architecture. The components include Service Function Forwarder (SFF), Service Functions, proxies, etc. All of these components compose a specific SFC-enabled domain. And different domains are interconnected with SFC encapsulation. The high-level architecture of these components is shown in Fig. 2.

In each SFC-enabled domain, the Service Function Forwarder (SFF) is responsible for injecting the SFC encapsulated traffic into public network and receiving the traffic. However, the encapsulation is not used for routing or path selection, it relies on the outer network transport [21]. The underlying devices (e.g., router, switch)