Optimal Embedding of Hybrid Service Function Chain with DAG Abstraction

Xu Lin*, Deke Guo†*, Yulong Shen*, Guoming Tang†, Bangbang Ren†

*Schoool of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071, China

†Science and Technology on Information Systems Engineering Laboratory

National University of Defense Technology, Changsha, Hunan, 410073, China

***Abstract*—In Network Function Virtualization (NFV) enabled traffic engineering, it is an important and challenging problem to optimally embed a Service Function Chain (SFC) over a target network. Recently, to reduce the traffic delivery latency, SFC with parallel VNFs (named hybrid SFC) is proposed. In this paper, aiming at cost minimization we address the optimal hybrid SFC embedding problem. Specifically, we develop a novel abstraction model for hybrid SFCs with Directed Acyclic Graph (DAG), which can convert diverse hybrid SFCs to the standardized DAG-SFC form. Then, we propose a breadth-first search based greedy method to solve the problem. The simulation results validate the effectiveness of our approach for cost reduction in hybrid SFC embedding.**

I. Introduction

Network Function Virtualization (NFV), has been proposed to replace special-purpose hardware by hosting virtual network function (VNF) software on general-purpose CPUs or virtual machines. This could bring us rapid deployment, network scalability, low-cost update and encourage innovation on network. To capture many known benefits of cloud computing such as decreased costs and easy management, VNFs could be hosted in the public cloud or private clouds.

To get a specific and complete end-to-end service from the source to the destination, network flow needs to pass through multiple VNFs in a particular sequential order, which is known as the service function chain (SFC). How to find out such a routing path for a flow with a specific SFC is known as SFC embedding, which is a joint procedure of VNF assignment and path selection. Recently, many researches focus on this problem with different design goals, such as minimum cost, maximum network throughput, delay reduction or jointly considering of all the above. However, they could not achieve a breakthrough on the total delay reduction due to the sequential process between VNFs in traditional SFC.

Recently, some emerging researches show the possibility of VNF parallelism [1] and propose a framework to enable parallelism for those VNFs without ordering requirement [2]. As shown in Fig. 1(b), the SFC can be deployed with the combination of sequential and parallel VNFs, which we called **hybrid SFC** in this paper. With hybrid SFC, it has been validated that the traffic delivery delay can be significant reduced [2]. Nevertheless, current work only experiments hybrid SFC under the single server scenario. The general deployment of hybrid SFC over the network has not been considered yet.

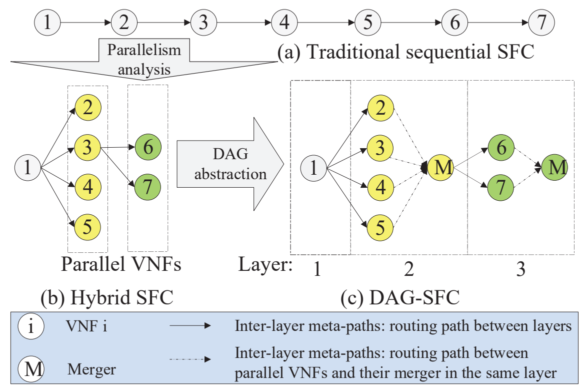


Fig. 1. From sequential SFC to hybrid SFC with DAG abstraction.

In this work, we aim to minimize the total cost when embedding a hybrid SFC under the constraint of node and link capacities over a priced cloud network. Specifically, we take the efforts as follows. Firstly, we propose a novel abstraction model, a kind of standardized Directed Acyclic Graph (DAG), to clearly model the orchestration of hybrid SFCs. Then, we propose the optimal hybrid SFC embedding problem and design a greedy method based on breadth-first search (named BBE) to solve it efficiently. Our simulation results demonstrate that BBE has a good performance at total cost reduction.

II. Hybrid SFC Embedding with DAG Abstraction

A. DAG Abstraction for Hybrid SFCs

Based on the analysis of [1], [2], we propose a novel hybrid SFC abstraction model using a standardized DAG. As demonstrated in Fig. 1, a sequential SFC could be transformed to a hybrid form by analyzing the parallelism in the chain, so that we could abstract the hybrid form with a DAG. As Fig. 1(b) shows, the SFC can be divided into several VNF sets, each of which contains VNFs that can be executed in parallel and is called a *parallel VNF set* (PVS). Then, as illustrated by Fig. 1(c), the SFC can converted to multiple layers, each with a PVS. The relation between layers (i.e., VPSes) is still sequential. Notice that each layer, which contains more than one VNFs, is usually followed by a merger that is responsible for integrating the middle results from the parallel VNFs.

Generally speaking, for any service function chain, we can divide it into one or multiple serial layers, each consisting of

978-1-5386-2542-2/18/$31.00 ©2018 IEEE

a PVS (followed by a merger) or a single VNF, as illustrated by Fig. 1(c). Therefore, the above DAG abstraction can be a standardized procedure for hybrid SFC transformation, and we name the transformed form as DAG service function chain (DAG-SFC). With the standardized form of DAG-SFC, we then turn to solve the DAG-SFC embedding problem.

B. Optimal DAG-SFC Embedding

We consider the cloud network as our target network, which is an overlay network built upon the underlying fundamental network (e.g., established by the telecom operators). In a cloud network, there are many geo-dispersed cloud nodes that are connected via network links. Upon each node, there can be multiple VNF instances, deployed by third party providers or the network operators. Additionally, each VNF instance may have a rental price (corresponding to its deployment cost and resource consuming) and a traffic processing capacity, while each network link also has a link price and the bandwidth capacity. From the perspective of a consumer, how to reduce the total cost when embedding a required hybrid SFC into the network is significantly important.

With the above overview, we formally define the optimal DAG-SFC embedding problem as follow:

*Definition 1 (**Optimal DAG-SFC Embedding Problem**):* Given the target network and a traffic flow with source-destination pair, the problem is to strategically embed a specified DAG-SFC into the target network, so that the total traffic delivery and processing cost (including link cost and VNF rental cost) can be minimized, under the constraints of network link capacity and VNF traffic processing capability.

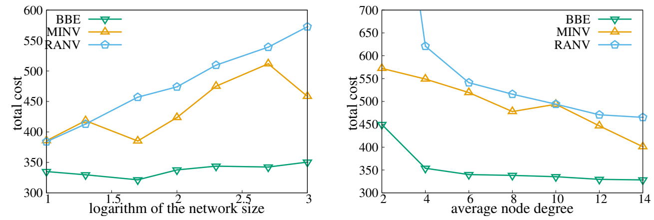
III. Solutions to DAG-SFC Embedding

The optimal DAG-SFC embedding problem is NP-hard, we omit the proof due to the space limitation. To tackle such difficult problem, we propose a Breadth-first Backtracking Embedding (BBE) method, which is inspired by the breadth-first search idea. The major steps of BBE are as follow:

1) Search for multiple feasible sub-solutions as the candidates of embedding solution for one single layer;
2) Repeat the above candidate searching process layer by layer, until all layers of the DAG-SFC are traversed;
3) Among the compete embedding solutions formed by series of sub-solution candidates, select the cheapest one as the final solution.

In detail, to generate the feasible sub-solution candidates of the l^{th} layer based on a specific sub-solution at the $(l-1)^{th}$ layer, three steps need to take, i) forward search, ii) backward search and iii) candidate sub-solution generation.

Forward search: Forward search is based on the solution of previous layers of DAG-SFC. The function of the l^{th} layer forward search is two-fold: i) to find an adjacent node set (called Forward Node Set, FNS) which includes all required VNFs of l^{th} layer (according to the DAG-SFC) from the end node of the $(l-1)^{th}$ layer, and ii) to route the inter-layer meta-paths (as shown by the solid connections in Fig. 1(c)) between the $(l-1)^{th}$ layer and the l^{th} layer.



(a) Impact of network size. (b) Impact of average connectivity.

Fig. 2. Simulation results.

Backward search: After the forward search, for each node in FNS containing a merger, we start a backward search process. The function of backward search is two-fold: i) to further narrow the FNS obtained in previous forward search process, and ii) to route the l^{th} layer inner-layer meta-paths (as illustrated by the dotted connections in Fig. 1(c)).

Candidate sub-solution generation: After each backward search, we generate a series of sub-solutions based on the results of corresponding forward and backward search process. As each forward search process should base on a sub-solution of the last layer, the sub-solutions would naturally have dependencies and the relationship between sub-solutions would follow a tree topology (named Sub-Solution Tree, SST). Thus, each leaf node of the sub-solution tree could uniquely indicate a complete candidate solution, which is formed by connecting all sub-solutions on the acyclic path from the leaf node to the root of the SST. At last, relying on the SST, BBE will select the cheapest solution from the candidates as the final solution.

IV. Performance Evaluation

With simulated network graphs and VNFs, we validate the effectiveness of BBE by comparing with two baseline algorithms, named **RANV** and **MINV**. The RANV and MINV algorithms select the VNFs in the random and optimal manner, respectively. They connect the VNFs via the minimum cost path obtained by the Dijkstra algorithm.

Fig. 2 shows the results from our simulations, from which we can see that BBE algorithm outperforms the baseline algorithms by up to 30% cost reduction. Moreover, as the network condition changes, the performance of our BBE algorithm keeps stable.

V. Conclusions

In this work, we studied the embedding problem of hybrid SFC, which aimed to jointly minimize the total VNF rental cost and link cost in SFC embedding. Firstly, we proposed an explicit DAG abstraction model to simplify and standardize the description of the hybrid SFC. Then, we proposed a breadth-first based greedy method to tackle the NP-hard problem. Finally, our simulation results showed that our approach can much reduce the hybrid SFC embedding cost.

References

[1] Y. Zhang, B. Anwer, V. Gopalakrishnan, B. Han, J. Reich, A. Shaikh, and Z. L. Zhang, "Parabox : Exploiting parallelism for virtual network functions in service chaining." *SOSR 2017*.
[2] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "Nfp: Enabling network function parallelism in nfv," in *ACM SIGCOMM 2017 Conference*.