SET: a Simple and Effective Technique to improve cost efficiency of VNF placement and chaining algorithms for network service provisioning

Shohreh Ahvar, Mohammad Mirzaei, Jeremie Leguay, Ehsan Ahvar, Ahmed Medhat, Noel Crespi, Roch Glitho

SET: a Simple and Effective Technique to improve cost efficiency of VNF placement and chaining algorithms for network service provisioning

Shohreh Ahvar
Telecom SudParis
Evry, France
shohreh.ahvar@telecom-sudparis.eu

Mohammad M. Mirzaei
University of Tehran
Tehran, Iran
mehdi.mirzaei@ut.ac.ir

Jeremie Leguay
Huawei Technologies
France Research Center
jeremie.leguay@huawei.com

Ehsan Ahvar
Univ. Rennes, Inria, CNRS, IRISA
Rennes, France
ehsan.ahvar@inria.fr

Ahmed M. Medhat
Technical University of Berlin, Germany
a.hassan@campus.tu-berlin.de

Noel Crespi
Telecom SudParis, Evry, France
noel.crespi@mines-telecom.fr

Roch Glitho
Concordia University, Montreal, Canada
glitho@ece.concordia.ca

***Abstract*—Network Functions Virtualization (NFV) is a promising solution to provide cost-efficient, scalable and rapid deployment of network services. It allows the implementation of fine-grained services as a chain of Virtual Network Functions (VNFs). In order to place the VNF chains in the network, several cost-efficient methods have been already proposed. However, a few works have considered order of VNFs to reduce the cost.**

In this paper, we propose a Simple and Effective Technique (SET), which can be easily combined with VNF placement methods to dramatically improve their cost efficiency by considering different possible orders for the VNFs in the chain.

As a proof-of-concept, we combine the proposed technique with one of the recent cost-efficient VNF placement and chaining algorithms called CCVP. The results show that the combination can yield significantly better cost than CCVP operating solo.

I. Introduction

Nowadays computer network services are considered as a key component in keeping the network running at all times.

To provide various network services, the telecommunication service provider′s network includes a number of middleboxes. They can support various types of functions, and for this reason, middleboxes are important for network operators.

By increasing network requirements in both scale and variety, service providers have to add new middleboxes and upgrade already existing middleboxes continuously. One recent research shows that the number of different middleboxes in enterprise network is comparable to the number of physical routers [1]. However, adding and updating middleboxes comes out with high Capital Expenditures (CAPEX) and Operational Expenditures (OPEX). In addition, the services deployed traditionally are not scalable.

To address these issues, Network Functions Virtualization (NFV) [2] [3] has been proposed to transform middleboxes from specialized hardware appliances to software running on inexpensive, commodity hardware (e.g., x86 servers with 10Gb NICs) [4].

An NFV-based (i.e., software-based) middlebox is a good solution to reduce CAPEX, as network operators no longer need to buy specialized hardware. Moreover, software-based middleboxes can be deployed and managed dynamically without the necessity of having network administrators which reduces OPEX [3]. In addition to improving OPEX and CAPEX costs, NFV gives an opportunity to network operators to manage their network functions easily. The software-based middleboxes are referred to Virtual Network Functions (VNFs) in NFV terminology.

In order to use VNFs, they can be placed on computational nodes (e.g., servers, switches, data centers) that meet their resource demands. The computational nodes must provide NFV Infrastructure (NFVI) functions to support the execution environment. The placed VNFs can also be chained together to provide a required service.

Placement and chaining of the VNFs can affect on both quality of service (QoS) and cost. For this reason, it has recently attracted attention of both network providers and researchers.

Many methods have been proposed to place chains of VNFs with the goal of minimizing the cost for the service providers [1]- [3] and [5]-[20]. However, a few of them considered the effects of ordering of VNFs on minimizing the cost by taking into account the compression/decompression rate, capacity and requirement of VNFs and servers. Compression of traffic happens with firewalls and compression/decompression is a feature of media gateways.

This paper makes the following contributions. First, we question the effects of the VNF ordering on placement algorithms in the achievement of the SET objective (i.e., cost efficiency). To this end, we put ourselves in the shoes of designers and describe a typical VNF placement process through a use case. We then propose a Simple and Effective Technique (SET) which can work in combination with VNF placement and changing algorithms. SET considers different

possible ordering for the functions (e.g., a firewall) which may not have ordering constraints.—This flexibility optimizes resource usage as it may reduce the number of activated functions and decrease the network footprint."

Afterwards, by combining SET with CCVP (i.e., a recently proposed VNF placement algorithm), we show that the combination of the proposed method with CCVP can actually outperform the CCVP alone with changing the order of VNFs in the chain.

The rest of the paper is organized as follows. In Section II, we overview related work. A use case is presented in Section III to demonstrate the importance of VNF ordering. Section IV presents the main contribution of this paper which is basically the SET technique. The assessment of SET is covered in Section V. Finally, Section VI concludes the paper.

II. Related Work

Several VNF placement and chaining algorithms which pay special attention to aspect of cost efficiency have been recently proposed. Some of them focused on specific individual costs. Luizelli et al. [5] and Fang et al. [6] tried to minimize the number of instances, Moens et al. [7] minimized the number of used servers and Qu et al. [8] targeted minimizing the communication bandwidth usage.

Unlike the above-mentioned works with the simple objective, Ghaznavi et al. [9] presented a solution where the elasticity overhead and the trade-off between bandwidth and host resource consumption were considered for placement of a same type of VNF. In another study, Ghaznavi et al. [10] completed their work for multiple VNF instances placement. Mechtri et al. [11] tried to maximize the providers revenue based on the number of accepted CPUs and bandwidth resources.

Riggio et al. [12] minimized the links and nodes utilization to increase the accepted service chain requests in enterprise WLANs. The authors then extended their work in [13] where a VNF placement heuristic, called WiNE (Wireless Network Embedding), was proposed. Finally, a few studies have attempted to consider more comprehensive cost models. Lin et al. [14] presented an MILP and Game Theory based VNF placement with the goal of minimizing the cost of deploying VNF instances as well as the computing and network cost in optical networks. Zeng et al. [15] considered the cost of IT resource and spectrum utilization of fiber links as their objective in addition to the cost of VNF deployment (instantiating) for the VNF placement in optical data centers. Bouet et al. [16] proposed a solution where the cost includes the network as well as the license cost per site and vCPU in VNF instances. Bari et al. [1] considered also a penalty cost to be paid to the customer for the Service Level Objective (SLO) violations.

Our previous work [17] considered a complete cost model including license, computing and communication cost. They introduced a cost adaptive model for low and high traffics and different units of cost.

None of the above mentioned works have considered the effects of VNFs ordering of the chain on cost where modification rate, VNFs capacity and requirement are taken into account in VNF placement and chaining algorithms. By introducing a use case, next section shows this effect.

There are also quite a few works which have considered ordering of VNFs. Allybokus et al. [18] considered partial ordering as a constraint and proposed a heuristic solution based on a linear relaxation of the problem. However, their work does not consider the rate modification for the VNFs which makes the effect of ordering more critical. Also, our work is different from their study, because we provide a simple and effective technique to improve the cost efficiency which works with any VNF placement and chaining algorithm. When we have rate modification feature in a VNF, then the next VNF in the chain receives less traffic, so it can be shared between more flows and fewer VNF instances can be used as a result. In this regard, Addis et al. [19] considered this feature for the VNFs which is called compress/decompress rate. In their work, they introduced tunneling VNFs which require decompression. However, unlike our work, they have not considered the effect of this feature on having different cost for different order of VNFs in the chain.

III. USE CASE

In this section, we present a use case to show the effects of VNF ordering on cost where the VNF placement algorithm takes into account the VNFs requirement and capacity in the placement and chaining of VNFs. To this aim, we define a chain with a partial order constraint for its VNFs. Let us consider a chain including 3 VNFs with the size of 1 vCPU and 2 GRAMs. Assume V1 is a Deep Packet Inspection (DPI), V2 is a traffic shaper (we call it QoS VNF), and V3 is a firewall (i.e., FW). We can also assume that all the flows pass through the same service with the same order. The objective is to find optimal order and placement for VNFs.

After considering the ordering constrains, 3 ordering options are possible : Option 1-DPI, QoS, FW; Option 2-DPI,FW,QoS; and Option 3-QoS.FW,DPI.

We implemented one of the recently proposed cost-efficient VNF placement and chaining algorithms (CCVP [17]) which considers VNF capacity and requirement in taking decision for VNF placement and chaining. In CCVP, the same service with the same order is used for all the flows. In order to better evaluate the performance of our SET method, we extend CCVP by considering the rate modification for the VNFs as well.

CCVP first creates a graph G of the network topology. In the second step, based on the graph G characteristics and other received information (e.g., available instances and requested flows), it selects appropriate servers (i.e., server with highest centrality and enough capacity), deploys instances on them and indicates each flow passes through which selected server for all VNF types of the chain. The procedure of placing new instances of VNFs of the chain is continued until the total cost becomes larger than the former ones (and the former one could

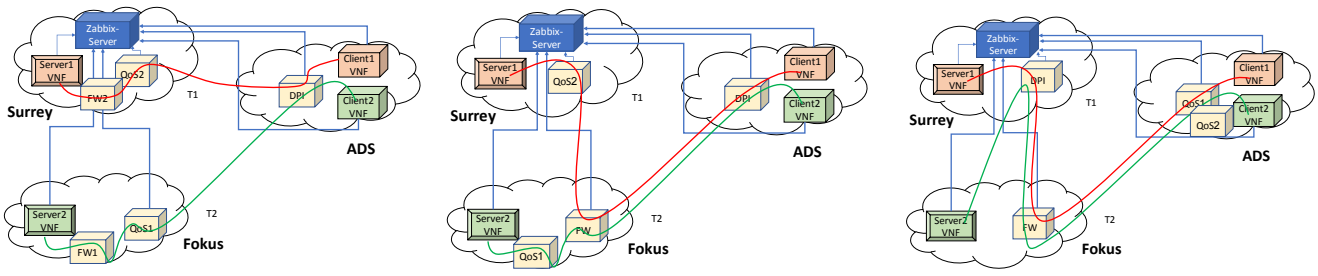

Fig. 1. feasible options: (a) Option 1 placement, (b) Option 2 placement, (c) Option 3 placement

assign all requested flows successfully). It is worth mentioning that CCVP modifies each source of flows by replacing a server which hosts already allocated VNF instance with a previous source of flows (i.e., flow source modification technique).

For the topology, we consider a real platform called SoftFIRE. Three European testbeds federated in SoftFIRE project are considered in this use case. A computing cost is considered for each testbed based on Azure cost model. Since our VNF size is considered as size of 1 vCPU and 2 GRAMs, we use the computing per instance per month for instance type A1 v2 in Azure. The cost for this instance based on the location of the testbeds in Azure is as follows : Ads (Europe)= 30.50 ,Surrey (UK) = 37.20 , Fokus (Germany): 34.97.

For the communication cost (i.e., the sum of the bandwidth used by the chains in the network), we used communication cost for outbound data for each Gb per month in Azure (i.e., 0.696). Based on Azure cost model, the traffic inside the testbeds is free[2]. Also, for VNFs License cost with refer to FortiGate-VM00 (http://www.avfirewalls.com/), we assume 1250 dollar per VNF. Capacity of the VNFs is considered 4 Gb/s, and for the testbeds available capacity we consider 2 vCPU, 4G RAM for each testbed. 2 traffics have been assumed:
ADS $\longrightarrow$ Surrey (T1), ADS $\longrightarrow$ Fokus (T2).
Loads of the flows are as follows:
T1 = 1000 Gb/month and T2=3000 Gb/month.

The CCVP placement decisions, based on this ordering as an input, are shown in figure 1).

With calculating the cost shown in I, we can see the effects of ordering on the cost when VNF placement and chaining algorithm takes into account VNFs capacity and requirement. Option 1 leads to deploy more VNF instances. Therefore it has the highest amount of license and computing cost. Also, the communication cost of the placement by CCVP for different ordering on all 3 options is different. Therefore, by trying differing ordering, we can reduce the cost for the provider. However, it takes time to check all the possible orders especially where the topology is large and the chain is long. Therefore, there is a need of a heuristic to narrow the search space.

TABLE I
COST DETAILS FOR DIFFERENT ORDERING OF VNFS IN CCVP

	License	Communication	Compute	Total
Option1	6250	2784	290.16	9324.16
Option2	5000	3480	211.3	8691.3
Option3	5000	7656	214.26	12870.26

IV. A SIMPLE AND EFFECTIVE TECHNIQUE (SET)

This section presents our proposed SET technique. As depicted in Section III, ordering of VNFs plays a significant role on the total cost of VNF placement and chaining algorithms. In the proposed method, we applied the Tabu search to find the best order. In fact, the SET technique is a Tabu search which prunes various feasible order of VNFs in the chain by considering the partial order constraints. Then, it selects the best permutation based on its placement costs. SET works with any VNF placement and chaining algorithm which calculates the cost of each permutation.

A. Tabu search

A feasible permutation $fp : [f_1, f_2, ..., f_n]$ is an order of VNFs, f_i, which satisfies partial order constraints. With n VNFs, there is at most $n!$ feasible orders of VNFs. If we have k out of $n!$ feasible permutations, then, there is a set of all feasible permutation $FP : \{fp_i | i \in [1, k]\}$. The task of the Tabu search is to find the best order of VNFs (i.e., an order with the minimum cost, among all the feasible orders)

We define three actions for our Tabu search which can generate a new permutation np from another ordered VNFs. However, it is possible that the generated permutation is unfeasible $np \notin FP$. In such cases, the permutation is not considered. Our actions can generate all possible orders and are listed below:

- **Insertion.** This action inserts f_i in place of f_j and shifts others.
- **Swap.** This action swap f_i and f_j.
- **Reversion.** this action revers a given sub sequence of an order.

The Tabu search starts with a random permutation $ro \in FO$ as the current order. It then analyzes the neighbors of this order, which can be generated by authorized actions and after

that, selects the best neighbor to be considered in the next iteration. When a neighbor is selected, the related action is added to the Tabu list. In each iteration, authorized actions are the actions which are not included in Tabu list and at the end of each step, the most old action is removed from the Tabu list. This steps is repeated in all iterations as long as the cost is under a predefined threshold or a fixed number of iterations is done.

B. The underlying algorithm

Our proposed technique is capable to work with all VNF placement and chaining algorithms. It is worth mentioning that the level of success for SET depends on the items that the VNF placement algorithm takes into account. CCVP already considers the capacity and requirement of VNFs and server properties. However, the VNFs with compression/decompression features were not considered in the algorithm decision. In this paper, we extend CCVP [17] by adding the modification rate consideration for VNFs to perfectly show the effect of ordering on minimizing the cost.

The pseudo code of SET integrated with the extended version of CCVP algorithm is illustrated in Algorithm 1.

Algorithm 1: Pseudo code of SET integrated with CCVP

```
Tabu_list = { }
choose current_order randomly
best = current_order
best_cost = ∞
while all iterations done or best_cost ≤ threshold do
    best_neighbor = [ ]
    best_neighbor_cost = ∞
    for action_i ∉ Tabu_list do
        Generate the neighbor of current_order called
          N by action_i
        if N ∈ FP then
            cost = CCVP(N)
            if cost ≤ best_neighbor_cost then
                best_neighbor_cost = cost
                best_neighbor = N
    current_order = best_neighbor
    Add actions{best_neighbor} to Tabu_list
    remove oldest action from Tabu_list
    if best_neighbor_cost ≤ best_cost then
        best_cost = best_neighbor_cost
        best = best_neighbor
return (best, best_cost)
```

V. Performance Evaluation

The goal of this section is to show the effect of SET on cost efficiency of VNF placement algorithms.

SET has been implemented in MATLAB (the source code is available here: https://github.com/mehmir/set). The simulation setup and the metrics, which are used for the evaluation, are described in the following.

TABLE II
Simulation Parameters

Simulation Parameters	
Bandwidth cost (Dollar/Mbps)	10
License cost per VNF instance (Dollar)	1000
Operational cost (Dollar/vCPU)	5
Flow size per source and destination pair (Gbps)	1
Number of flows	4,8

A. Simulation Setup

Network Topology: We used a real network topology from university data center research network [20] composed of 23 nodes and 42 links.

Traffic data set: the number of flows changes from 4 to 8. The sources and destinations are generated randomly by using MATLAB. It is worth highlighting that, even though the flows configuration is set randomly, once set, it remains fixed across the runs of all tested algorithms ensure comparability of the results. Chains of 5 VNFs have been applied in the simulations. The capacity of the physical servers, physical links, VNF instances and server running costs, which we have used in this section, have been derived from research papers [1] [16]. Table II shows the server and VNF data assumption which have been used in the evaluation. The capacity, requirement and compression/decompression rate for the VNFs are selected differently for the each VNF to show the effect of these items in differentiating the cost of different orders. The first, second and third numbers after each VNF in the following list, show the capacity, requirement (i.e., Required CPU cores for the VNF) and compression rates for the VNF respectively. VNF1 5000 1 1; VNF2 4000 2 0.8; VNF3 3000 3 0.9; VNF4 2000 4 1; VNF5 1000 5 1. We also considered a partial order constraint for the VNFs where VNF2 should be before VNF3.

B. Cost Evaluation

Integrating SET with VNF placement algorithms which take into account the server and VNFs properties can help to reduce the cost. We verify this claim by integrating SET with CCVP.

We set the maximum number of iterations to 20 for Tabu search. As we can see in the table III, for 8 flows more orders meet the partial order constraints.

Fig. 2 depicts the results of the overall cost of the data center topology for 4 and 8 flows.

As Fig. 2(a) shows, SET selects ordering ID 10 as input of CCVP. If we consider, without SET, we might select the ordering ID 8 as the input of CCVP, SET can improve cost of CCVP around 50% by selecting the right ordering (i.e., ordering ID 10) as input. Similarly, as Fig. 2(b) shows, SET could improve cost of the CCVP algorithm up to 40% for 8 flows.

Both Figs. 2(a) and (b) validate that our proposed SET technique could noticeably improve cost of CCVP and it can

TABLE III
ORDERING OF VNFS IN CCVP

Order ID	4 flows	8 flows
1	5 2 4 3 1	2 4 5 3 1
2	5 4 2 1 3	1 2 4 3 5
3	4 2 5 3 1	4 1 2 3 5
4	5 1 4 2 3	2 3 4 1 5
5	2 1 4 5 3	2 5 4 3 1
6	5 2 4 1 3	2 1 3 4 5
7	2 4 1 5 3	2 1 5 3 4
8	1 4 5 2 3	2 1 4 5 3
9	1 4 2 5 3	2 5 3 4 1
10	5 4 2 3 1	2 4 3 1 5
11	5 2 1 4 3	2 5 1 4 3
12		2 4 3 5 1

(a)

(b)

Fig. 2. Simulation results: Overall Cost for different orderings (a) for 4 flows, (b) for 8 flows

be used, as a promising technique, for improving cost of VNF placement and chaining algorithms.

VI. CONCLUSION

This paper presented a Simple and Effective Technique (SET), which works based on Tabu search, to consider VNF ordering for improving cost of VNF placement algorithms. We believe SET can be integrated with any VNF placement and chaining algorithm (which takes into account capacity, requirement and compression rate of VNFs) to reduce the cost for the provider. As a proof of concept, we tested SET with integrating it with an extended version of one of the recent cost-efficient VNF placement and chaining algorithms (i.e., CCVP). The results proved our idea. As future work, we plan to integrate SET with more VNF placement and chaining algorithms.

REFERENCES

[1] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 50–56.

[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] 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.

[4] ETSI, "Etsi gs nfv 001, network function virtualization (nfv) use cases, v1.1.1." ETSI, , 2013.

[5] 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.

[6] 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, Aug 2016.

[7] H. Moens and F. D. Turck, "Vnf-p: A model for efficient placement of virtualized network functions," in *10th International Conference on Network and Service Management (CNSM) and Workshop*, Nov 2014, pp. 418–423.

[8] L. Qu, C. Assi, K. Shaban, and M. Khabbaz, "Reliability-aware service provisioning in nfv-enabled enterprise datacenter networks," in *IFIP/IEEE 12th International Conference on Network and Service Management*, Oct 2016.

[9] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *IEEE 4th International Conference on Cloud Networking (CloudNet)*, Oct 2015, pp. 255–260.

[10] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, and R. B. R. Ahmed, "Service function chaining simplified," http://arxiv.org/abs/1601.00751, coRR, vol. abs/1601.00751, 2016.

[11] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 533–546, Sept 2016.

[12] R. Riggio, T. Rasheed, and R. Narayanan, "Virtual network functions orchestration in enterprise wlans," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 1220–1225.

[13] R. Riggio, A. Bradai, D. Harutyunyan, T. Rasheed, and T. Ahmed, "Scheduling wireless virtual networks functions," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 240–252, June 2016.

[14] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Demand-aware network function placement," *Journal of Lightwave Technology*, vol. 34, no. 11, pp. 2590–2600, June 2016.

[15] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type vnf forwarding graphs in inter-dc elastic optical networks," *Journal of Lightwave Technology*, vol. 34, no. 14, pp. 3330–3341, July 2016.

[16] M. Bouet, J. Leguay, and V. Conan, "Cost-based placement of vdpi functions in nfv infrastructures," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–9.

[17] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi, and R. Glitho, "Ccvp: Cost-efficient centrality-based vnf placement and chaining algorithm for network service provisioning," in *2017 IEEE Conference on Network Softwarization (NetSoft)*, July 2017, pp. 1–9.

[18] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi, and E. Gourdin, "Virtual Function Placement for Service Chaining with Partial Orders and Anti-Affinity Rules," *ArXiv e-prints*, May 2017.

[19] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, Oct 2015, pp. 171–177.

[20] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," *ACM SIGCOMM conference on Internet measurement*, pp. 267–280, 2010.