A Study On Optimizing VNF Software Cost

Ghoshana Bista, Eddy Caron, Anne-Lucie Vion

A Study On Optimizing VNF Software Cost

Ghoshana Bista
ENS de Lyon, Inria, CNRS, Université de Lyon. LIP
Orange S.A
Pessac, France
ghoshana.bista@orange.com

Eddy Caron
ENS de Lyon, Inria, CNRS, Université de Lyon. LIP
Lyon, France
eddy.caron@ens.lyon.fr

Anne-Lucie Vion
Orange S.A
Pessac, France
annelucie.cosse@orange.com

Abstract—**Network Functions Virtualization (NFV) is one of the promising and futuristic technologies in the area of network architecture. It was launched as a leading network technology by networking and telco companies promising to lower Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) with greater flexibility to scale up and lower the resources. VNF software (Virtual Network Function) functionalities, metrics, rights, etc. are not yet thoroughly explore which in turn services provisioning and licensing complexes, error, and lethargic. We call this problem VNF-SC (Virtualize Network Function Software Cost) or VNF-LC (Licensing Cost). In this study, we have tried to explore the existing problems and complications in VNF licensing/cost and recommend the novel idea using use-cases which help to optimize the VNF software cost. This study also discusses the challenges that need to be addressed in the upcoming days. In SAM (Software Asset Management) universe usually, software cost and software license costs are used interchangeably so in our article we also follow the same trends.**

Index Terms—**Virtual Network Function (VNF), VNF licensing, Software Asset Management (SAM), Software cost/licensing.**

I. INTRODUCTION

The term name as "middlebox" was widely used in the networking world to provides telecommunication, networking services for a few decades. Scaling up or down to resources in the middlebox is time-consuming and highly expensive [1]. To have a grip on these issues NFV had been proposed. NFV is a network architecture that was first proposed by ETSI in 2012 [2]. The fundamental idea of NFV is to replace hardware middleboxes such as Firewall, Network Address Translation (NAT), proxy, etc. into virtualized function in a virtual infrastructure. In VNF the functions are implemented in the Virtualize manner in a dynamic way as software so the role of SAM (Software Asset Management) is crucial. Organizations spend at least 30 percent of the IT budget on software licenses [3]. So SAM is not an option it is an absolute need. Software license which is the subset of SAM is not only some sort of legal agreement, neither it is some sort of policy. It is an X-factor for service provider and end-user

too. With the assist of a standardized, authentic, and reliable software license model a service provider can generate its revenue similarly service user or organization can save its expenses. If we traced back to the history of software licensing then we will found in 1987 the "networking license server" by Apollo computer and in 1988 "FLEXIm" from Highland software was introduced in the commercial license manager world [4]. As commercial licenses were getting heated, open-source software licenses were also getting it's market such as GNU, OpenBSD (Berkeley Software Distribution), etc. In 1999, N.F hosling *et al.* proposed a software probation model which help to categories the five existing problems at that time that need to be addressed by the organization they were legal (counterfeiting), ethical (intellectual property right respect), managerial and economic issues (true-up cost) and technical (monitoring) [5]. In today's world, popular licensing methods are, subscription, flat, and pay as you grow. Since network architecture is expanding rapidly these above mentions models are not enough to cover all features of future licensing so to incorporate this swift we are in desperate yearning for some standard new models but VNF license models are not an easy road there are lots of challenges. Licensing in VNF is at the beginning stage which gives reason to lots of vendors, services provider companies to forwards their own licensing models this is creating chaos, confusion in the SAM environment. [6] which pushes us for this study. We call this problem VNF-LC (Virtualize Network Function – Licensing Cost) problem or VNF-SC (Software Cost) problem. A new model needed to be implemented without being complicated so that it will meet new services and demands. When and what needs to be a charge for customer license enablement must be connected with a business or CRM (Customer Relation Management) [6]. Moreover how to charges or billing the usage services is a pivotal aspect in VNF licensing. Actually, software cost includes lots of costs like hardware cost, hardware maintenance cost, upgrade cost, etc. Till this research has been writing as far as our knowledge there hasn't been any comprehensive research

regarding VNF licensing costs and models. Most of the articles published in past were focused on the problem in NFV traffic, networking, scheduling, and the overall cost not in licensing characteristics. Here in our research, we try to categorise all the solution which were proposed in past and we show how they are related to VNF license cost and how they missed to cover software cost, metrics, etc.

Section II presents categorizes of existing problems. Section III provides our algorithm, use cases to optimize software cost. Section IV concludes this articles.

II. Classification of existing trends in VNF software cost

In this research authors categories, existing trends of VNF software cost in three parts A) Total cost including software cost B) Total cost without including software cost C) Others (CPU allocation, scheduling, etc.).

A. Total cost including software cost

There are lots of papers [1], [5], [7], etc. that talk about the total cost of VNF which are for deployment, buying cost of VNF but very few research had been conducted including license cost. In the paper [1] authors present a cost-efficient idea for VNF chaining placement. They forwarded some interesting ideas like centrality computer module, fitness compute module, etc. to find the best and optimal placement of VNF in a way that overall cost is minimized. This research is focused on overall cost with respect to network metrics, not with software cost. In [7] authors purposed SET (Simple and Effective Technique). This is the extension of [1] in which they used feasible permutation to find optimal orders and placement of VNF. They find the order of VNF after they used their previously developed model which shows orders of VNFs would help to minimize the total cost, again here analysis was not based on software licensing features much more focus on reducing overall cost. In [8] they considered the VNF opening cost which is the cost of running idle VNF. It considered energy cost, cost of installing a VM, link, processing, and licensing cost for each VNF. In this paper also, they didn't talk about software cost it was focused on VNF placement and chains. In [9] authors presented to support service differentiation in terms of availability and delay minimizing the cost for VNF placement. Generally, they forwarded two solutions ILP (Integer Linear Programs) formulation and heuristic approaches to obtain near-optimal solutions. In order to estimate path cost, they find the least path cost satisfying the QoS constraint according to the agreed service chain SLA (Service Level Agreement). In [10] placement of virtual Deep Packet Inspection (vDPI) was presented and solve using a centrality based greedy algorithm. Algorithm considered fitness value composed of network resources cost and license cost of vDPI. Well, this manuscript analyzed was ILP and heuristic base which were made to the optimization of cost of vDPI and placement. Although it considered a license cost it didn't fully explore the software cost and metrics. Thus, from these few papers presented above all considered the software

licensing cost but they never explore more on optimizing on software cost of VNF.

B. Total cost without including software cost

To calculate proper CAPEX and OPEX software cost cannot be ignored. In [11] which was the most comprehensive paper related to the VNF orchestration problem. To optimize the OPEX they presented four components 1) VNF deployment cost 2) energy cost 3) cost of forwarding traffic 4) penalty for SLO (Service Level Objectives) violation. Their second aim was to minimize resources active server and links by the possibility to increase accommodating more traffic on the same physical resources. Although, it was the finest paper they didn't consider software cost. In paper [12] the author proposed mixed-integer linear programming presenting resource constraints, routing constraints, and QoS constraints. They show that MMRQC (Multi-Source Multicast Routing with QoS Constraints) was a Steiner tree problem and to solve they proposed Multi-source Multicast Tree Construction (MMTC). The algorithm estimated to find common links so that it can place SFC (Service Function Chains) which will help to improve resource utilization. In [13] they considered anti-affinity between VNFC (Virtual Network Function Components), they used constraint-based heuristically applied in virtualized mobile network infrastructure providing EPCaaS (Evolved Packet Core as Service). The cost of deployment was measured in terms of DC (Data Center) utilization such as compute and networking. The two constraint-based strategies were VSD (Vertical Serial Deployment) and HSD (Horizontal Serial Deployment). By using these two strategies they measure the performance in terms of load distribution on overall servers. Thus, all these articles were conducted without considering software cost.

C. Others

A lot of researches were focused on other aspects such as resource allocation of CPU, proper handling of network traffic, some were based on trust contract, service chaining, etc. In a paper like [14], they focused on efficient NFV deployment in data centers networks. They try to slow down the growth of east-west traffic and minimizing the data center resources. They used solving bin packing problems in each node classifying outputs bin in good and bad. This process generated trees structure which helps to locate VNF for controlling traffic growth. So basically their algorithm was based on flow assignment. Some of the research was concerned with smarts licensing as in [15] it presented a way to decentralize and provides security on pay-as-you-grow models in an the automated and flexible way of using blockchains. Their system model was based on Communication Services Providers (CSP) and Network Software Vendor (NSV) relationship. So this paper talked more about software management based on its availability and flexibility using Ethereum blockchain. In [16], for proper deployment configuration machine learning model was used along with resources affinity for the automation deployment. This paper analyzed the VNF flavor configuration

between OvS (Open vSwitch) and SR-IOV (Single Root Input-Output Virtualization) infrastructure which help to construct architecture for selecting the best VNF flavor automatically showing that VNFC were not memory or CPU bound.

III. Models, simulation and experimental setup

To address the above VNF-SC problem we present different use cases scenario which is trivial but crucial. We proposed models for selecting a combination of VNF with one of the vital metrics *i.e.* is SAU. But SAU is not the only metric used for analyzing VNF license. Depending upon infrastructure, there are lots of other metrics for example in core telco they use active users and attached users too but SAU is widely accepted and commonly used metrics for VNF licensing. So we focus our analysis on SAU. SAU can be expressed as,

$$SAU = \max(\sum_{\text{All Vnf}} \text{Hourly Average SAU}) \qquad (1)$$

SAU is a reference for a license, max is maximum over the day and all VNF is all concerned VNF instances in the network. In this article whenever we say the best combination of VNF that means VNF which meets threshold QoS or meets minimum cost or SAU. Since we cannot present real value (cost) in each table provided by the vendor due to confidentiality we consider arbitrary value close to it.

1) Case I: If an organization can choose to use different VNF from different companies theoretically it will lower the prices and the organization will definitely reduce CAPEX and OPEX. Our foundation ideas is shown in Fig. 1. Different shapes of VNF are used to differentiate different VNF.
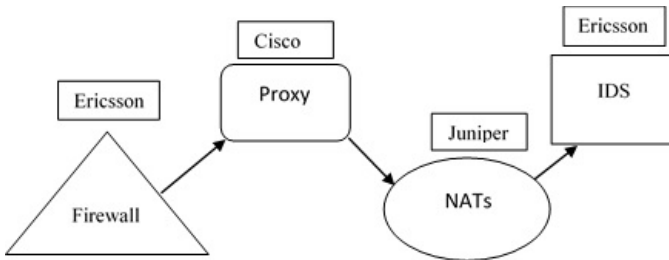

Fig. 1. Implementation of different VNF using different VNF provider .

Fig. 1 can be simplified in Table I. We can see that users can choose a cheap vendor, Cisco for a firewall which is (50 €), for IDS user can choose Ericsson (45 €), and for proxy, the user can select Ericsson or Juniper (50 €). Thus, by selecting different vendor organization can optimize the CAPEX and OPEX price.

TABLE I
Cost of VNF function according to different VNF providers.

Companies	Firewall	IDS (Intrusion Detection System)	Proxy
Ericsson	60 € (annual)	45 € (annual)	50 € (annual)
Cisco	50 € (annual)	60 € (annual)	60 € (annual)
Juniper	55 € (annual)	50 € (annual)	50 € (annual)

Our case II is methods of flavor and range.

2) Case II: The Flavor of VNF generally means categorizes of VNF depending upon its infrastructure capabilities. By using the deployment flavor also we can optimize the software cost. It can be shown in Fig. 2.
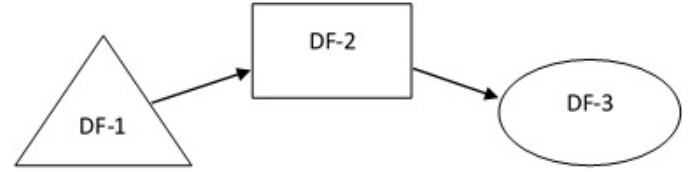

Fig. 2. VNF implementation using different flavor.

If we have lots of flavors then we can form the combination of the flavor and check if our threshold QoS can be met by this combination or not. Fig. 2 is further analysis in Table II.

From Table II if our need is 2vCPU and 2GB RAM then the organization can select DF1-Firewall, DF-2 Proxy, and DF-3 IDS. Thus, by choosing different flavors depending upon the QoS requirement we can minimize software cost. We set up some experiments to analyze the relation of SAU and deployment flavor and we perform some simulation on it. which is our case III.

3) Case III: From Table III depending upon our needs of SAU and requirements of the system we can select any flavor. For this experiment, we suppose our need for SAU for the firewall is 100 to 200. Similarly, for IDS SAU is from 600 to 700 and for proxy, SAU is from 1000 to 1100. So our simulation select 100 for the firewall which is DF-1, simulator select the lowest one from the range. Likewise, for IDS it selects DF-2 and for Proxy, it selects 1000 which is DF-2. Thus, the simulator makes the combination of DF-1, DF-2, and DF-2 for firewall, IDS, and proxy respectively. We present this problem in different ways too. We can consider of the highest or lowest combination of SAU for different flavors too. For example, we can make a combination of these functions such as (100, 500, 900), (100, 600, 1000), (100, 700, 1100), etc. Basically, we are calculating the permutation of each function or flavor with each other. After the permutation simulator will select the sum of each combination if users need the highest combination of SAU simulator will select the highest combination of SAU if users need the lowest combination of SAU simulator select lower combination of SAU. So using this simulator we can find the flavor depending upon the SAU either lowest or highest depending upon the needs. These operations can also be further diagnosed in different ways in case IV.

4) Case IV: We can make a combination of SAU and cost for each VNF, well there is lots of software cost (maintenance cost, installation cost, etc.) but over here in Table IV we take license cost only, unit price but it will not change if we consider the total cost.

TABLE II
Relation of flavor with SAU.

Flavor/function	DF-1			DF-2			DF-3		
Firewall	2vCPU	2GBRAM	50 €(annual)	3vCPU	1GBRAM	60 €	2vCPU	3GBRAM	55 €
IDS	2vCPU	2GBRAM	60 €(annual)	4vCPU	3GBRAM	75 €	4vCPU	4GBRAM	85 €
Proxy	2vCPU	1GBRAM	45 €(annual)	2vCPU	2GBRAM	55 €	4vCPU	4GBRAM	60 €

TABLE III
Relation of flavor with SAU.

Flavour	SAU for Firewall	SAU for IDS	SAU for Proxy
DF-1	100	500	900
DF-2	200	600	1000
DF-3	300	700	1100

TABLE IV
List of cost with SAU.

Flavour	SAU for Firewall	Cost	SAU for IDS	Cost	SAU for Proxy	Cost
DF-1	100	110.56 €	100	120.63 €	100	100.56 €
DF-2	500	110.56 €	500	120.63 €	500	100.56€
DF-3	1000	60.16 €	1000	50.12 €	1000	40.56 €

We can analyze flavor according to SAU or cost. We build a simulator which helps to select according to cost or SAU as per the need of the users. If cost is selected users have to make choice highest or lowest cost. In our example let say the user selects the lowest cost it will be 60.16 € for the firewall and likewise for other VNF too. It should be noticed that if the user then selects the lowest cost it will have the highest SAU, in this case, it is 1000. Similarly, users can use SAU for selecting suitable flavor, in these examples we take the highest SAU i.e. 1000, similarly for other VNF too, which will help to select the lowest cost. The most important point to be noted here is that when we have the highest SAU we have less cost but our infrastructure cost (vCPU, vRAM) will increase. It is obvious that if SAU is higher more vCPU is required and a higher size of vRAM is required. Thus, organizations, enterprises must have their threshold cost and SAU before choosing the flavor or service providers which help to optimize software cost.

IV. Conclusions

The contemporary virtual networking market is already been introduced with 5G; because of which NFV is undoubtedly going to be the future of networking architecture. Thus, a deeper understanding of NFV is the absolute need of the hour. From our use-case scenario, we clearly show that we can optimize the software cost using the VNF chain of different vendors. Also, a combination of the cost of VNF help to reduces the software cost. Similarly, a combination of SAU and deployment flavor have a significant role to reduce the software cost acquiring the threshold QoS. We are already in 5G now but still, we didn't have the proper mechanism to handle software licensing/cost in VNF. Right now we are dealing with the VNF chain and its cluster but we will have a cluster of NFV so how we going to properly implement license in this cluster is still obscure. Soon we will have a slicing network so how we going to implement licensing in

each slice is still in the infant stage. All these and other issues will be covered in our future works.

References

[1] 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)*. IEEE, 2017, pp. 1–9.

[2] ETSI. (2017, Dec.) Network functions virtualisation(nfv) release 3 licensing management. [Online]. Available: https://www.etsi.org/deliver/etsigr/NFV-EVE

[3] D. Y. P. Luxembourg. (2018, Jan.) omtco.eu. [Online]. Available: http://omtco.eu/references/sam/it-costs-the-costs

[4] M. Christiano. (2018, Jan.) A brief history of software license management. [Online]. Available: http://reprisesoftware.com/blog/2007/01/a-brief-history-of-software-license-management-2

[5] A.-L. Vion, "Software asset management and cloud computing," Ph.D. dissertation, Universitè Grenoble Alpes, France, Sept 2018. [Online]. Available: https://dumas.ccsd.cnrs.fr/LIGSRCPRERODS/tel-01901991v1

[6] D. Clarke. (2018, Jan.) Nfv license management: The missing piece of the puzzle. [Online]. Available: https://medium.com/@CableLabs/nfv-license-management-the-missing-piece-of-the-puzzle-ce97cf48f066

[7] S. Ahvar, M. M. Mirzaei, J. Leguay, E. Ahvar, A. M. Medhat, N. Crespi, and R. Glitho, "Set: a simple and effective technique to improve cost efficiency of vnf placement and chaining algorithms for network service provisioning," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 28–29.

[8] M. Obadia, J.-L. Rougier, L. Iannone, V. Conan, and M. Brouet, "Revisiting nfv orchestration with routing games," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2016, pp. 107–113.

[9] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "Qos-driven function placement reducing expenditures in nfv deployments," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.

[10] M. Bouet, J. Leguay, T. Combe, and V. Conan, "Cost-based placement of vdpi functions in nfv infrastructures," *International Journal of Network Management*, vol. 25, no. 6, pp. 490–506, 2015.

[11] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 725–739, 2016.

[12] K. Xie, X. Zhou, T. Semong, and S. He, "Multi-source multicast routing with qos constraints in network function virtualization," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[13] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 60–66, 2015.

[14] P.-W. Chi, Y.-C. Huang, and C.-L. Lei, "Efficient nfv deployment in data center networks," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5290–5295.

[15] F. Magnanini, L. Ferretti, and M. Colajanni, "Efficient license management based on smart contracts between software vendors and service providers," in *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2019, pp. 1–6.

[16] V. Riccobene, M. J. McGrath, M.-A. Kourtis, G. Xilouris, and H. Koumaras, "Automated generation of vnf deployment rules using infrastructure affinity characterization," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, 2016, pp. 226–233.