



**HAL**  
open science

# VNF Software Cost Modeling Based On Telecom Network

Ghoshana Bista, Eddy Caron, Anne-Lucie Vion

► **To cite this version:**

Ghoshana Bista, Eddy Caron, Anne-Lucie Vion. VNF Software Cost Modeling Based On Telecom Network. The 6th International workshop on Software Defined Networks and Network Function Virtualization 2022 (SDN-NFV2022), Nov 2022, Paris, France. hal-03915710

**HAL Id: hal-03915710**

**<https://inria.hal.science/hal-03915710>**

Submitted on 11 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# VNF Software Cost Modeling Based On Telecom Network

Ghoshana Bista  
*Orange*  
*Inria*  
Pessac, France  
ghoshana.bista@orange.com

Eddy Caron  
*UMR CNRS - ENS Lyon -*  
*UCB Lyon 1 - Inria 5668*  
Lyon, France  
ORCID 0000-0001-6626-3071  
Eddy.Caron@ens-lyon.fr

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

**Abstract**—Network Functions Virtualization (NFV) extracts network functions, allowing software running on standardized compute nodes to install, control, and manipulate them. NFV integrates cloud and virtualization technologies to rapidly develop new network services while enhancing flexibility, scalability, and automation. The adoption and success of NFV are contingent on managing software costs and resources. It involves multiple conflicting objectives to be addressed, such as energy, licenses, resource migration, and resource cost. That makes it harder to implement resources and cost management. Thus, to address this problem, this article first module the Total Cost of Ownership (TCO) of software cost of VNF based on two crucial costs License Cost (LC) and Resources Cost (RC). LC is associated with software rights granted (and consumed) to the user. RC depends on the use of resources. The usage of resources determines the energy consumption by the VNF, which helps to estimate the Energy Cost (EC). After modeling TCO, we minimize the TCO by using the VNF resources sharing and consolidation technique. We proposed this technique by including LC and RC. Our evaluation results verify that considering RC, LC, and EC for VNF sharing minimizes the TCO with optimum resource utilization.

**Index Terms**—License cost, Simultaneous Active Users, Software licensing, VNF, Software Cost, Cloudification, Softwarization.

## I. INTRODUCTION

Virtualization of networks introduced a novel way to imagine telco networks, initiating a technological shift from hardware to cloudification and softwarization, leveraging the emergence of NFV. When ETSI proposed NFV in 2012 [1], virtualization appeared to be not only an IT topic but also a networking opportunity. NFV help in the emergence of v-EPC (virtual Evolve Packet Core) such as MME (Mobile Management Entity), AUSF (Authentication Server Function), AMF (Access and Mobility Management Function), C-RAN (Cloud-RAN), and other components which are commonly referred to as Virtual Network Functions (VNF). NFV not only

paves the way for SDN (Software Defined Network) but also works as a complementary entity to SDN and network slicing. Undoubtedly, it will be a milestone for many upcoming generations [2]. Due to the virtualization of the networks, the concept of the Network as a Service (NaaS) increased network services commercialization [3]. IDC forecasts that NFV markets will grow from \$ 7.5 billion in 2020 to over \$ 29 billion in 2025<sup>1</sup>. Hence, the use of NFV is going to increase. VNF is a virtual component of the NFV that can operate most telco functional entities such as DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name Service), and IDA (Intrusion Detection Algorithm) in a virtualized manner as software that can run on generic hardware enabling networks to become more flexible, agile, and operate at the least cost. Since VNF is software, TCO of software cost and management is necessary to achieve resources and cost optimization.

One of the common approaches to managing the software life-cycle management (deployment, changing, removing, updating, upgrading) and costs in IT using SAM (Software Assets Management), further expanding to telco and other network sectors. Although VNF act as software, only a few studies have been performed regarding software cost [4], [5]. Software cost is not only the purchase price of software to be used but includes various other costs such as license, resources, software updates, upgrades, and maintenance. The license gives the right to use software; hence it is an unavoidable constraint for software. Licenses help to frame the rights and obligations of CSPs (Communication Services Providers) to use the software. License cost depends upon the amount of rights granted to use. The volume of rights granted is contractually defined with the help of several metrics. License help to structure the users,

<sup>1</sup><https://www.idc.com/getdoc.jsp?containerId=prUS47769321>

and users determine the resource's usage. Therefore, the demand for resources is linked with the client, whose nature depends upon the nature of the license. Also, resource utilization is a crucial factor that affects energy consumption [6]. Thus, we confirm that license, resources, and energy are interdependent. Since these three components are linked, so does their cost. LC is associated with the cost of the license, RC is the cost of resources, and EC is the cost of energy consumption during operation. All these costs have a significant impact on software costs for NFV. Thus, proper estimation of TCO is vital to businesses for immediate and long-term costs. TCO is also crucial for any Return On Investment (ROI) analysis. Hence, we use these costs to estimate the TCO of software cost. Afterward, we minimized the TCO using VNF sharing and consolidation techniques. In this work, VNF sharing includes sharing its virtual resources, such as vCPU and vRAM. Consolidation strategies mean shifting the VNF from one point-of-presence (PoP) or datacenter to another. The decision for VNF sharing and consolidating VNF is based on the requirement of resources, licenses, and both at SaaS and NaaS levels, which eventually help minimize the TCO with optimum resource utilization.

This paper's fundamental goal is to provide a cost model to estimate the TCO of software for telecom architecture. After, we reduce the TCO using different VNF sharing and consolidating techniques. Finally, this paper tries to fill the gap in scientific and academic research on software cost with a techno-economic analysis of VNF for the 5G network.

The rest of the paper is organized as follows: Section II provides our related work and business model. Section III present our models and simulation and Section IV concludes this research.

## II. BUSINESS MODEL AND RELATED WORK

### A. Business Model

Our assumption of the business model is divided into three stages; (i) VNF provider, (ii) Service provider, and (iii) End-user. VNF provider is a company that provides VNF, the service provider is an entity that provides services on VNF infrastructure, and the end-user are service users. Different entities have different roles, but these can be changed, or they can be operated by the same entities too.

### B. Related Work

In this section, we briefly explore state of the art for the VNF resources allocation and cost problems.

#### a) *VNF resources allocation in general:*

Article [7] categorizes the VNF allocation problem into three-stage VNF-CC (Chain Composition), VNF-FGE (Forward Graph Embedding), and VNF-SCH (Scheduling). All these problems are related to VNF resource allocation, which affects the software cost of the VNF, but they fail to consider software cost. In articles [8] and [9], they presented resource allocation based on VNF forwarding graph embedding (VNF-FG). These articles are similar to those previously presented; they focus on the VNF mapping too, and the only difference is that research [9] focused was for 5G. Beyond still, they did not consider the software cost of VNF. In article [10] they studied the scheduling of VNF; this research helps us to know the relation between users and VNF, but they did not consider VNF cost. Research [11] presented the VNF service function chains (SFCs) placement problem with various algorithms; this research helped us to build the service chain of VNF, but they did not provide more information about VNF cost. There are many types of research such as [6], [12]–[14] that present various ways for VNF placement latency, energy consumption, and optimization of resources; thus, these show the importance of VNF resources. However, they did not consider software costs. Some of the research [3] even includes machine learning to reduce energy consumption. Thus, all these above mentions and many more articles presented the importance of optimum resource allocation on VNF. However, the research focuses much more on network traffic, traffic steering, VNF placement, energy consumption, and load balancing than the software cost of VNF.

#### b) *VNF cost in general:*

In [15] article was based on QoS of non-cooperative the domain of the network using deep learning to reduce the bidding price, cost of the network, optimization of Network Service Chain, and Forward Graph (FG), so this research gave the idea for the formulation of our resources cost. Article [16] provided the FlexShare algorithm, which gives near-optimal deployment cost, but it did not consider the software cost perspective. In research [17] they minimize the end-to-end delay while reducing the overall deployment cost; this article presented only the deployment cost of VNF. There are few research such as [18], [19], that presented the operational cost, which deals with and provides a good insight into the LC and RC budget of VNF but not the

software cost. Thus, we found a research gap in software cost; moreover, there is a lack of optimizing models for the software cost. Nevertheless, these costs are significant for companies. Therefore, we are undertaking this in this article.

### III. SYSTEM MODEL, SIMULATION

#### A. System Model

This section provided a system model for the VNF software cost and sharing. Resources sharing and consolidation are well-known strategies for the optimum use of resources, but using these strategies for software sharing is a less explored area.

a) *Software cost Analysis:* Now for the software cost analysis, we consider the following cost

- 1) **License Cost (LC):** Like any software, VNF is protected by IPR (Intellectual Property Right), which is ensured by license usage rights and restrictions at a given cost. We construed our license cost based on license reference. License Reference (LR) helps to determine license requirements for the VNF at a considered duration. LR is based on Simultaneous Active User (SAU), which is shown in Equation (1);

$$LR_{SAU} = \max_{\forall p \in D} \left( \sum_{\forall p \in V} average(SAU_r, p) \right) \quad (1)$$

where,  $V=(1,2,3,\dots,v)$  is a set of all concerned VNF (can be the same or different types) in the node.  $D$  is a set of days  $(1,2,3,\dots, D)$ . SAU are users who are active in VNF, using services provided by VNF. Other metrics to estimate LR, such as transmission per second, bandwidth, and subscriber, but we choose SAU as it is much more relevant to our 5G network presented in this research. After estimating the required licenses, we can calculate license costs based on usage: capacity, and consumption. We have talked in detail about this in our article [4]. In the capacity module, there is pre-paid amount until the threshold level; in our case, LR, if the threshold is exceeded, then extra cost will be added based on unit cost. In consumption, users pay as they consume the resources. For this article, we use the capacity module. We have the initial cost  $\rho$  up to the LR (threshold). If the license's demand exceeds the LR at any time, the extra cost will be incurred. Equation (2) shows our LC.

$$LC_i = \sum_{i=1}^n (\rho_i + \beta_i \times \omega_i) \quad (2)$$

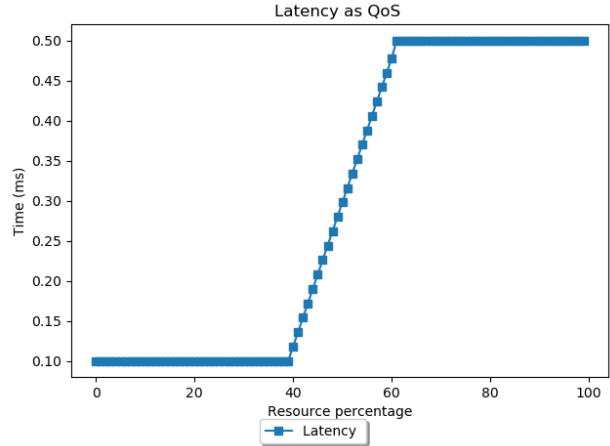


Fig. 1: Latency (QoS) of a VNF

- 2) **Resource Cost (RC):** For this study, resources include the virtual resources that operate VNF. Our virtual resources are vCPU and vRAM, but they are not limited to this; they include data centers, Virtual Machines (VM), and Virtual Networks (VN) resources. We consider the total resources of VNF with 100% in this research. In general practices in industries, we found that resources can be used to a certain percentage, in our case 40%, up to which all users will get promised QoS, but after 40% to a certain percentage, in our case, 60% QoS slowly decrease as shown in the Fig. 1. In this figure, we can see that latency started to increase as it across 40% and became worst after 60%. Thus, it depends upon the client's needs if they wish to receive good QoS every time they need to add resources as VNF reaches 40%, but if they can compromise, they should add resources after 60%. Resource cost can be expressed as;

$$RC_i = \sum_{i=1}^n (\eta_i + \alpha_i \times \gamma_i) \quad (3)$$

- 3) **Energy Cost (EC):** It is the cost depends upon the resources of VNF. For this energy consumption, we follow a similar way in ETSI [20]. It can be formulated as Equation (4) depending upon power consumption and unit energy cost. It is measured in Joule or kWh.

$$EC_i = \sum_{i=1}^n (\phi_i \times \delta_i) \quad (4)$$

4) **Total Cost Of Ownership (TCO)**: The total cost of ownership for software includes indirect and direct costs incurred by the software, which is critical in Return On Investment (ROI). As for now, in this research, we only considered RC, LC, and EC for TCO because licenses encompass the rights to use the software. In contrast, resources determine the capacity of software performance, and energy consumption depends on resources. Thus, this cost helps to quantify the cost of services at various levels of complexity in the supply chain. So, using this information, we formulate our TCO as follows:

$$TCO = \sum_{i=1}^n (RC_i + LC_i + EC_i) \quad (5)$$

and our objective is to minimize this cost

$$\min TCO = \min \sum_{i=1}^n (RC_i + LC_i + EC_i) \quad (6)$$

Subjected to

$$L \geq 1 \quad (7)$$

$$B \geq 1 \quad (8)$$

$$TH_f \preceq D_f \quad (9)$$

5) **Power Consumption of VNF**: Estimating the energy consumption of the VNF power consumption of each VNF is very important. We follow the similar paradigm presented by ETSI [20] as; VNF is a software application in which energy consumption cannot be measured separately concerning NFV. Hardware has the ideal power regardless of the system activity. For this article, the power of VNF is set following the guideline given by ETSI. The power module of VNF proposed by ETSI gave an arbitrary value to each VNF. VNF Energy Efficiency (EE) is estimated as in Equation (10), which is related to power.

$$P = P_{load} - P_{idle} \quad (10)$$

Where,  $P$  is the power consumption for the NFVI platform due to VNF deployment.

$P_{load}$  is the power consumption of the NFVI platform with VNF deployment.

$P_{idle}$  is the power consumption of the NFVI platform without VNF deployed.

It is measured in watts (W) and corresponds to the rate at which energy is converted.

6) **VNF energy efficiency**: In this work, we used VNF Energy Efficiency Ratio (EER) to measure the relation between useful output and energy consumption of VNF using our algorithm. VNF's EER is presented by ETSI, which is given as follows;

$$VNF_{EER} = \frac{Useful_{output}}{Power_{consumption}} \quad (11)$$

$Useful_{output}$  can be expressed as Packets/s, subscriber, transmission per second. In our article  $Useful_{output}$  is the SAU server by VNF.

b) *Sharing and Consolidation Algorithm*: VNF sharing and VNF consolidation are similar to many extents.

**VNF sharing**: It is the process of sharing the VNFC (VNF Components) like vCPU, vRAM, and memory. In this research, we share the resources based on LC and RC. VNF license is the right to use a VNF under terms and conditions pre-determine between client and service providers. Similarly, RC is based on resources. In our research, we share the resources after 60% of resources have been consumed by implementing our heuristic approach.

For this, we classified software sharing into two types:

- **Vertical Sharing**: It is sharing virtual resources like vCPU and vRAM with other VNFs between two PoPs, data. These types of sharing are commonly referred to as heterogeneous sharing. Only virtual resources can be shared in this sharing, not the software's license. Our research is based on this sharing. In similar ways, we can subdivide consolidation into vertical consolidation. Vertical consolidation is similar to vertical sharing.
- **Horizontal Sharing**: It is the method where resources and licenses of software can be shared with other VNFs when necessary. It is commonly referred to as homogeneous sharing. The only difference between heterogeneous sharing is that the license can be shared with other VNFs, which means that VNFs have similar functionalities. These methods are helpful when a replica of VNF is used for load balancing and redundancy. Similarly, horizontal consolidation also moves the VNF within the same network, PoPs.

**Consolidation**: The consolidation of VNF means shifting the VNF from one datacenter to another so that the previous data center can turn off and save resources and energy. The fundamental objective of this research is to minimize the TCO; for that, we use VNF sharing on this sharing when VNF can share resources up to or

more than 40%; then, in such case, we consolidate the VNF from one PoPs to another to reduce the costs of the resources. So, for us, consolidation is a case during sharing. Below we present our heuristic approach based on which we share and consolidate resources.

- **RMS (Resources Methods for Share):** For this method, we shared resources with other VNFs based on the resources required and available. A pseudo-algorithm for this process is shown in Algorithm 1. At first, we generated random users in each VNF then we calculated LR. Then in the second step, again, users were generated. This time we associate each user with some percentage of resources. The third step is to check the resource consumption of each VNF so to find which VNF will be able to share and which VNF needs resources. In step 6, conditions are checked if VNF can share resources or not; if  $VNF_i$  can share a resource with another VNF,  $VNF_l$ , then the resource requirement of  $VNF_l$  must be lesser than the other VNF in the network that required resources. If the condition is satisfied, then the resource is shared. If VNF can share resources more than or equal to 40%, then in such case, we consolidate the VNF to the other PoPs where the RC requirement is less among others. Step 10 is the summation of all resources available for sharing by each VNF. Thus, sharing happens in two steps. One is with each VNF, and the other is the summation of all VNFs. For example, if the VNF3 resources requirement is 10% and VNF1 and VNF2 can share 5% each individually, VNF can not fulfill the requirement of VNF3, but the summation of VNF1 and VNF2 can. If all the above conditions can not be satisfied, then resources will not be shared, and the cost of resources will be increased due to adding extra resources. All the cost LC, RC, EC, and TCO is determined using Equation (2) to Equation (5).
- **LMS (License Method for Share):** In this method, we share the resources of the VNF based on the license required in the VNF. This process arranges the VNF to receive resources in ascending order. Since, in this research, each user needed to be licensed. Furthermore, each user consumed some amount of resources, so there is a relation between license and resource consumption. A pseudo-algorithm for the LM is shown in Algorithm 2. In this method, almost all the steps and process is similar to RMS. Except that we share the resources between the VNF based on the least license require-

---

### Algorithm 1: An algorithm for RMS

---

**Data:** User should be at any time  $> 0$ , Resource range = (a,b),  
Number of VNF can be share= S  
**Result:** Resource share and get TCO

- 1 Calculated the LR;
- 2 Find out the resources consumption and license in each VNF by each user;
- 3 After computing whether VNF can participate in resource sharing or receiving;
- 4 **for** Up to number of VNFs that can share (S) **do**
- 5     **for** Up to number of VNFs that can receive (R) **do**
- 6         **if**  $RS(t)_i > RN(t)_l$  **then**
- 7             Check if this VNF has the least resources required or not with all the VNF in the system such as  
 $RN(t)_l < RN(t)_j, RN(t)_k$ .
- 8             **if yes check if resources can be shared  $\geq 40\%$**
- 9             **if yes consolidate VNF**, and then shared
- 10             Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
- 11             **if no share resources only** Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
- 12              $TCO_{RMS} = RC_{RMS} + LC_{RMS} + EC_{RMS}$ ;
- 13         **else if**  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  **then**
- 14             Check if this VNF has the least resources or not with all the VNF in the system such as  
 $RN(t)_l < RN(t)_j, RN(t)_k$ .
- 15             **if yes and Resources are shared with VNF**  
 $RN(x)_l \leftarrow Rs(x)_i$ ;
- 16              $TCO_{RMS} = RC_{RMS} + LC_{RMS} + EC_{RMS}$ ;
- 17         **else**
- 18             Resource cannot be shared need of extra resources

---

ment in the VNF. For example, if the license requirement in VNF1 is 10 and VNF2 is 5, resources will be shared in VNF2 if they need a resource.

- **LRMS (License and Resource methods for Sharing):** It is the combination of the methods mentioned above; resources and licenses. In this method, resource sharing is possible in that VNFs have the least license and resource requirements. This means that VNF arranges to receive the resources from the least resources and licenses to the highest. For example, if VNF1 requires 50% resources and the license requirement is 20, but in VNF2 resource requirement is 40%, and the license is 10 then VNF2 will get the resources to form VNF, which is sharing. A pseudo-code is presented in Algorithm 3.
- **CLM (Classic Method):** In this method, resources are shared without constraint; if VNF can share its resources, it will simply share with other VNFs on a first come, first serve. For this method, pseudo-code is presented in Algorithm 4. It is used to compare with our models. We use the same methods in other sharing processes to calculate LC, RC, and TOC.

---

**Algorithm 2: An algorithm for LMS**


---

**Data:** User should be at any time  $> 0$ , Resource range = (a,b),  
Number of VNF can be share= S  
**Result:** Resource share and get TCO

- 1 Calculated the LR
- 2 Find out the resources consumption and license in each VNF by each user
- 3 After computing whether VNF can participate in resources sharing or receiving
- 4 **for** Up to number of VNFs that can shared (S) **do**
  - 5 **for** Up to number of VNFs that can receive (R) **do**
    - 6 **if**  $RS(t)_i > RN(t)_l$  **then**
      - 7 Check whether this VNF has the least license required or not with all the VNF in a system such as  
 $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$
      - 8 **if yes check if resources can be shared**  $\geq 40\%$
      - 9 **if yes consolidate VNF** and then share  
Resources are share with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
      - 10 **if no share resources only** Resources are share with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
      - 11  $TCO_{LMS} = RC_{LMS} + LC_{LMS} + EC_{LMS}$ ;
    - 12 **else if**  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  **then**
      - 13 Check if this VNF has least license required or not with all the VNF in a system such as  
 $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$ .
      - 14 **if yes and Resources** are share with VNF  
 $RN(x)_l \leftarrow Rs(x)_i$ ;
      - 15  $TCO_{LMS} = RC_{LMS} + LC_{LMS} + EC_{LMS}$ ;
    - 16 **else**
      - 17 Resource cannot be shared

---



---

**Algorithm 3: An algorithm for LRMS**


---

**Data:** User should be at any time  $> 0$ ,  $R(x) = (a, b)$ , Number of VNF can be share= S  
**Result:** Resource share and get TCO

- 1 Calculated the LR;
- 2 Find out the resources consumption and license in each VNF by each user;
- 3 After computing whether VNF can participate in resource sharing or receiving;
- 4 **for** Up to number of VNFs that can share (S) **do**
  - 5 **for** Up to number of VNFs that can receive (R) **do**
    - 6 **if**  $RS(t)_i > RN(t)_l$  **then**
      - 7 Check whether this VNF has the least resources and license required or not with all the VNF in a system such as  $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$  and  $RN(t)_i < RN(t)_j, RN(t)_i < RN(t)_k$  and soon.
      - 8 **if yes check if resources share**  $\geq 40\%$
      - 9 **if yes consolidate VNF** and then share  
Resources is shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
      - 10 **if no share resources only** Resources is shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
      - 11  $TCO_{LRMS} = RC_{LRMS} + LC_{LRMS} + EC_{LRMS}$ ;
    - 12 **else if**  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  **then**
      - 13 Check if this VNF has least resources and license required or not with all the VNF in a system such as  $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$  and  $RN(t)_i < RN(t)_j, RN(t)_i < RN(t)_k$ .
      - 14 **if yes and Resources** are share with VNF  
 $RN(x)_l \leftarrow Rs(x)_i$ ;
      - 15  $TCO_{LRMS} = RC_{LRMS} + LC_{LRMS} + EC_{LRMS}$ ;
    - 16 **else**
      - 17 Resource cannot be shared, add extra resources

---



---

**Algorithm 4: An algorithm with Classic**


---

**Data:** User should be at any time  $> 0$ , Resource range = (a,b)  
**Result:** Resource share and get TCO

- 1 Calculated the LR;
- 2 Find out the resources consumption and license in each VNF by each user;
- 3 After computing whether VNF can participate in resource sharing or receiving;
- 4 **for** Up to number of VNFs that can share (S) **do**
  - 5 **for** Up to number of VNFs that can receive (R) **do**
    - 6 **if**  $RS(t)_i > N(t)_l$  **then**
      - 7 Resources is share with VNF  $RN(x)_l \leftarrow Rs(x)_i$  ;
      - 8  $TCO_{CLM} = RC_{CLM} + LC_{CLM} + EC_{CLM}$  ;
    - 9 **else if**  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  **then**
      - 10 Resources are shared with VNF  $RN(t)_l \leftarrow Rs(t)_i$ ;
      - 11  $TCO_{CLM} = RC_{CLM} + LC_{CLM} + EC_{CLM}$  ;
    - 12 **else**
      - 13 Resource cannot be shared, add extra resources

---

**B. Deployment Flavor (DF)**

We use the concept of flavor for the VNF. The flavor is based on VNFC (Virtual Network Function Components), vCPU, vRAM, and vStorage. For this study, we constructed the flavor based on vCPU and vRAM as shown in TABLE I. Since different flavors support a different amount of users and have different resources capacity, using the different flavors of VNF has an impact on RC, LC, and resource sharing. Which ultimately has an impact on software TCO. Thus, we have used DF along with resource sharing in our simulation and presented its impacts.

TABLE I: DF table

Flavor	Type	vCPU	vRAM (GB)
A	Small (S)	2	2
B	Medium (M)	4	4
C	Large (L)	8	8
D	Extra Large (XL)	16	16

TABLE II: Key Notation

Notation	Description
t	is time slot
$RS(t)_i$	Resource to be shared by VNF i at time t
$RN(t)_l$	Resource needed for VNF l at time t
$LN(t)_i$	License needed for VNF l at time t
(a, b)	a is the lower limit, and b is the upper limit for resource range
i,j,k,l	all these are the different VNF with the different function i,j,l $\in$ S,R
$\eta$	The initial resources cost
$\rho$	The initial license cost
$\gamma$	Resources exceeded
$\beta$	Unit cost for a license after the threshold is surpassed
$\delta$	Power consumption by VNF
$\omega$	Number of licenses surpass the LR
$\alpha$	Unit cost for the resources
$\phi$	Unit cost for Energy

---

### C. Use case

The primary purpose of presenting this use case is to show how the users can optimize energy cost and its impact on TCO and the environment. We adopted the 5G Gaming use case from article [21]. For this use case, we consider different VNFs from different VNF providers which offer the same function with different parameters. In addition, we consider the server, which will consume different power depending upon the VNF provider. Company A: 0.1Kw, Company B: 0.2 Kw, and Company C: 0.5 kW in an hour.  $CO_2$  emission per Kw is 0.709kg, and electric cost per Kwh is 0.108€. Thus, using this, we formed the TCO for a day as shown in TABLE III.

From this case, we can see that different VNF provider provides the same services with different TCO and carbon footprints. We can see from the TABLE III that company A emits less carbon in a day than any other VNF, but it provides less bandwidth. Also, we can see that company C provides good performance with a bandwidth of 1 Gbps, but it comes with a lot of costs, and its carbon footprint is higher than any other VNF. So it is up to the client what they want. If they are unwilling to contribute to the planet and only need the best performance, they can go for company C VNF. However, on the other hand, if they want to contribute to green energy and also meet their requirements, then company B VNF can be a good solution. Thus, from this use case scenario, we can see that carbon footprint depends on energy consumption and energy with resources, license, and users requirement. Hence, all these costs are interdependent. Therefore, reducing TCO and carbon footprint depends upon the client's requirements and systems.

### D. Simulation setup

We used the 5G architecture as shown in Fig. 2. Which we got from the telco industries, orange, France. Our focus is on the 5GC (core) network supported by VNFs. Which is PCF (Policy Control Function), UDM (Unified Data Management), AUSF (Authentication Server Function), AMF (Access and Mobility Management Function), and SMF (Session Management Function). For each VNF, we calculate the LR using Equation (1). After estimating LR in these VNFs, we considered that each user consumes some % of resources, basically from (0.1- 5)% and (1- 10)%. All the parameters used in this execution are presented in the TABLE IV. Then, we checked whether the user and resources exceeded the threshold or not. The threshold for the license is LR,

and for resources is 60% of resources. VNFs that have not exceeded their resources are the VNFs that can share resources with the VNFs that need resources. For sharing of resources, we implemented our methods and obtained the cumulative results for 30 days as shown from Fig. 3-8. For the software cost formulation, we consider VNF has initial resources cost and license cost. If VNF can share its resources with other VNFs, then resource cost will not increase, but if it cannot get resources from another VNF in the system, then cost will be increased. Also, if the VNF can share more than 40%, then VNF can be consolidated. We estimate the resources, license, energy, and TCO, and we use Equations (2) to (5). For the EE and latency, when the resources needed by VNF are fulfilled by sharing between the VNF, then latency and EE will not degrade. DF used in the simulation corresponds to vCPU and vRAM. DF1 is with 2 vCPU and 2GB vRAM in which each user consumes resources between (0.1-5), and DF2 is with 2vCPU and 2GB vRAM in which resources consumption is between (1-10)% by each user. DF3 is with 4GB vCPU and 4GB vRAM with resources consumption between (1-10)% by each user. A similar goes for DF4 with 8GB vCPU and 8GB vRAM with (1-10)% resources consumption by each user.

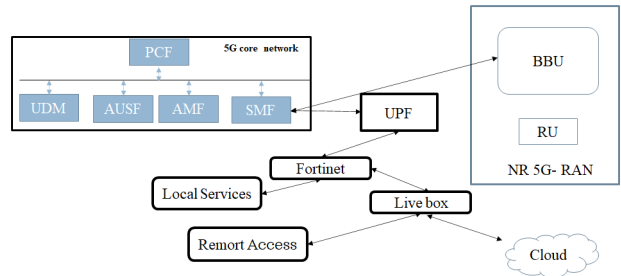


Fig. 2: 5G Network Architecture.

### E. Performance evaluation Results

**License cost.** Our methods play a significant role in LC. Fig. 3 shows cumulative license cost for 30 days with different flavors using different methods. We can easily see that flavor has an impact on all of the methods. Another significant result we can depict is that LRMS methods help reduce the license cost in DF1, DF3, and DF4 except in DF2, which is estimated at a higher cost than others but less than classic. This is because, at that instant, in DF2, VNF did not require both resources and licenses. Thus, VNF sharing in these cases did not happen. However, CLM has a high LC. **Resource cost.**



TABLE III: Different VNF providers

VNF Providers	Type	S FC set	Carbon footprint (Kg per Kwh)	Bandwidth	EC	RC	LC	TCO (€ Per day)
Company A	Gaming	NAT-FW-VOC-Woc-IDPS	1.7016	50 Mbps	0.259	10	20	30.259
Company B	Gaming	NAT-FW-VOC-Woc-IDPS	3.403	65 Mbps	0.518	20	30	50.518
Company C	Gaming	NAT-FW-VOC-Woc-IDPS	8.508	1 Gbps	1.296	30	35	66.296

TABLE IV: Simulation parameter.

Parameters	Value
License unit cost( $\beta$ )	5 €/SAU
Resource unit cost ( $\alpha$ )	7 €/resource
Energy unit cost( $\phi$ )	10 €/watt
Power consumption when consolidation	5 watt
Power consumption when sharing	7 watt
Power consumption without sharing	15 watt
User range between	(1, 50), (1, 500) and (1, 5000)
Resource range	(0.1, 0.5) and (1,10)
DF1	50 users (1-10)%
DF2	50 users (1-10)%
DF3	500 users (1-10)%
DF4	5000 users (1-10)%
Latency of each VNF (sharing)	0.1 [21]
Initial LC for CLM	1200 €
Initial RC for CLM	3500 €
Latency without sharing	0.5
$\rho$	2000
$\eta$	1000

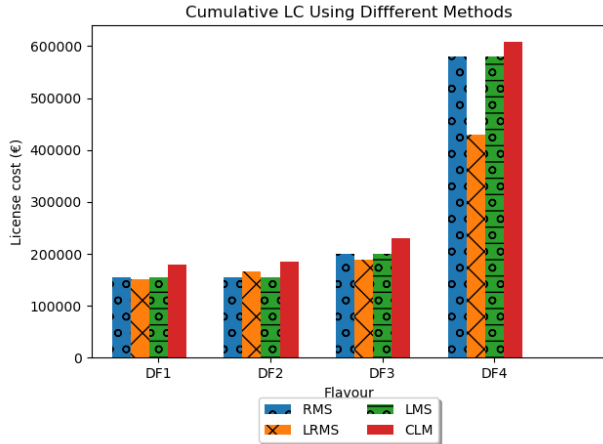


Fig. 3: License Cost Using Different methods

Fig. 4 shows that LRMS estimated lower costs in almost all DF except DF1 and DF2. In this, it is equal to the other two methods because, in DF1 and DF2, user and resource consumption are less they did not surpass the VNF license or resource usage simultaneously. However, for the LMS and RMS methods, resources cost is highest

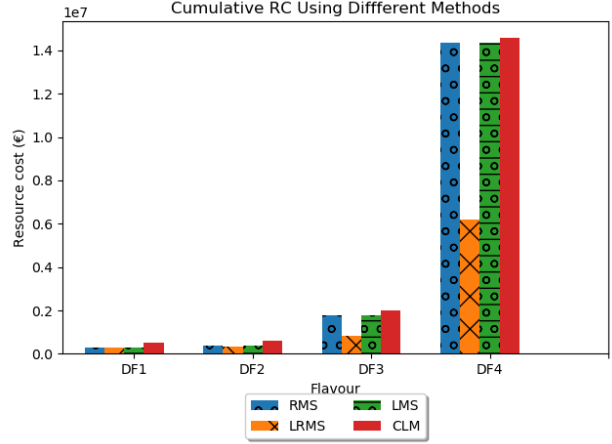


Fig. 4: Resource Cost Using Different methods

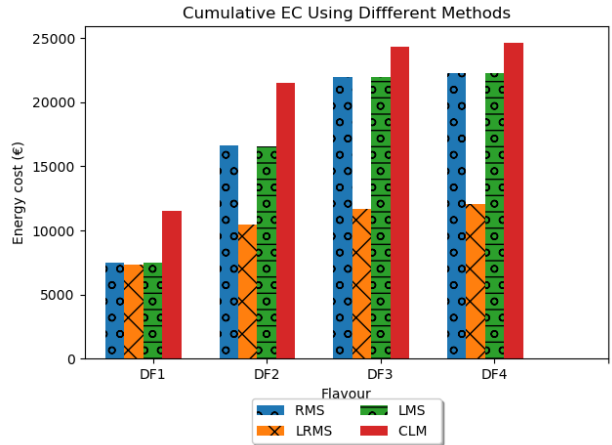


Fig. 5: Energy Cost Using Different methods

in DF3 and DF4; this is because VNF that required resources exceed the license or resources threshold respectively, so sharing is not possible and extra resources are added. Also, we can see that our methods outperform the CLM method. **Energy cost.** Our energy cost is related to resources, so its nature in Fig. 5 is similar to RC. LRMS outperformed the other methods. This means fewer resources are added, and more resource

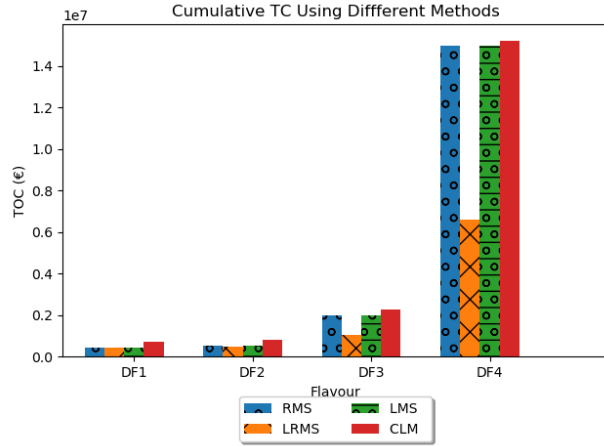


Fig. 6: TCO Using Different methods

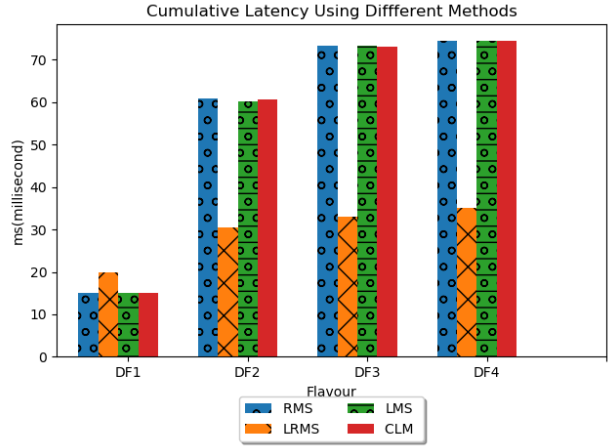


Fig. 8: Latency(ms) Using Different methods

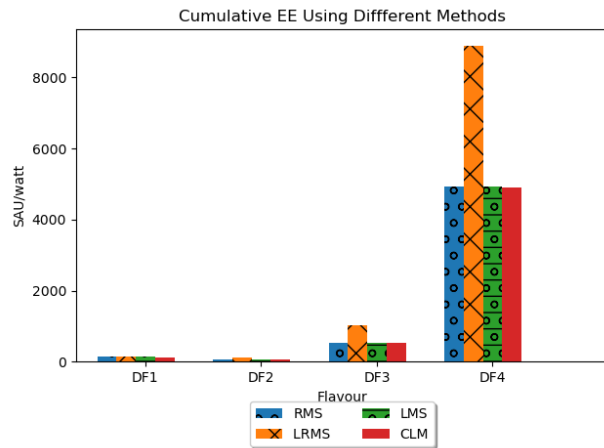


Fig. 7: Energy Efficiency Cost Using Different methods

sharing is performed in each case. Energy cost is less when resources share or consolidation has taken place. **Total Cost of Ownership.** This is our primary objective to module the software TCO and minimizes it. Fig. 6 shows our TCO estimation by different methods using different flavors. We can see that using LRMS reduces the TCO, and the CLM method is the worst among other methods. This is because more VNFs required licenses and resources at the same time. Also, we can see that as DF changes, LRMS shows significant results because as DF changes, resource capacity, amount of users, and quantity to share or need of resources changes. Finally, we can see that both other resource and license methods are similar because each user must be licensed, and each user consumes some resources. Due to this, we say that

licenses and resources are interdependent and cannot be neglected. **Energy Efficiency.** It is the amount of SAU served per watt by VNF. From Fig. 7, we can depict that in almost every case, LRMS outperforms the other methods. As the DF keeps increasing, its efficiency gets better. This happens because more resources are shared whenever resource sharing is available and possible. Therefore, more Numbers of SAU are being served by consuming less energy. **Latency.** Fig. 8 show the **cumulative time** for latency. From this, we can see that when using the LRMS method, the system's latency is better than other methods. This is also due to sharing resources efficiently. As we mentioned earlier, our latency depends upon the sharing; if possible, the cost will not augment. If sharing is not possible, the cost will increase because when sharing is not possible, they need extra resources, so they may have to wait longer.

#### IV. CONCLUSION

In this paper, we presented the study on VNF software cost in a 5G VNF use case. At first, we constructed the total cost of ownership for the software cost of VNF using very crucial costs LC (License Cost), RC (Resource Cost), and EC (Energy Cost). Then after we proposed several heuristic processes for resource sharing dynamically that have an impact on LC, RC, and EC. We use these heuristics strategies along with Deployment Flavour (DF). Our results show that it greatly impacts resource sharing and costs. We also presented a 5G use case scenario showing that user requirements impact energy and Total Cost Of Ownership (TCO). Moreover, user can minimize their costs considering their carbon footprint. Through our simulation of the 5G scenario,

we demonstrate that the heuristic LRMS can efficiently share resources reducing the TCO depending upon the DF and improving the energy efficiency and latency. Our results also conclude that software costs and the reduction of TCO does not solely depend upon one parameter or factor, or metric. It depends on several factors, such as scenarios, users' requirements, requirements based on budget, performance, and metrics used. Our future work will be on slicing and complex 5G and beyond cases.

## REFERENCES

- [1] ETSI. (2017, Dec.) Network functions virtualisation(nfv) release 3 licensing management. [Online]. Available: <https://www.etsi.org/deliver/etsigr/NFV-EVE>
- [2] N. Kiran, X. Liu, S. Wang, and C. Yin, "Vnf placement and resource allocation in sdn/nfv-enabled mec networks," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2020, pp. 1–6.
- [3] Y. Mu, L. Wang, and J. Zhao, "Energy-efficient and interference-aware vnf placement with deep reinforcement learning," in *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 2021, pp. 1–9.
- [4] G. Bista, E. Caron, and A.-L. Vion, "Total cost modeling for vnf based on licenses and resources," in *CLOSER 2022-12th International Conference on Cloud Computing and Services Science*, 2022.
- [5] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi, and R. Glietho, "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.
- [6] O. Soualah, M. Mechtri, C. Ghribi, and D. Zeghlache, "Energy efficient algorithm for vnf placement and chaining," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, 2017, pp. 579–588.
- [7] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.
- [8] F. Schardong, I. Nunes, and A. Schaeffer-Filho, "NFV resource allocation: a systematic review and taxonomy of vnf forwarding graph embedding," *Computer Networks*, vol. 185, p. 107726, 2021.
- [9] B. Zhang, Q. Fan, X. Zhang, Z. Fu, S. Wang, J. Li, and Q. Xiong, "A survey of vnf forwarding graph embedding in B5G/6G networks," *Wireless Networks*, pp. 1–24, 2021.
- [10] Q. Li, X. Wang, T. Zhao, Y. Wang, Z. Li, and L. Rui, "An improved genetic algorithm for the scheduling of virtual network functions," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–4.
- [11] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for loac balancing in nfv networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [12] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware vnf placement and chaining based on a flexible resource allocation approach," in *2017 13th international conference on network and service management (CNSM)*. IEEE, 2017, pp. 1–7.
- [13] M. A. Khoshkholghi, J. Taheri, D. Bhamare, and A. Kassler, "Optimized service chain placement using genetic algorithm," in *2019 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2019, pp. 472–479.
- [14] A. Mouaci, É. Gourdin, I. Ljubić, and N. Perrot, "Virtual network functions placement and routing problem: Path formulation," in *2020 IFIP Networking Conference (Networking)*. IEEE, 2020, pp. 55–63.
- [15] P. T. A. Quang, A. Bradai, K. D. Singh, and Y. Hadjadj-Aoul, "Multi-domain non-cooperative vnf-fg embedding: A deep reinforcement learning approach," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 886–891.
- [16] F. Malandrino, C. F. Chiasserini, G. Einziger, and G. Scalosub, "Reducing service deployment cost through VNF sharing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2363–2376, 2019.
- [17] A. Leivadeas, G. Kesidis, M. Ibnkahla, and I. Lambadaris, "Vnf placement optimization at the edge and cloud," *Future Internet*, vol. 11, no. 3, p. 69, 2019.
- [18] H. N. Van, F. D. Tran, and J.-M. Menaud, "Autonomic virtual resource management for service hosting platforms," in *2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*. IEEE, 2009, pp. 1–8.
- [19] A. C. Adamuthe, R. M. Pandharpatte, and G. T. Thampi, "Multiobjective virtual machine placement in cloud environment," in *2013 international conference on cloud & ubiquitous computing & emerging technologies*. IEEE, 2013, pp. 8–13.
- [20] ETSI, "Environmental engineering (EE), measurement method for energy efficiency of network functions virtualisation (NFV) in laboratory environment." [Online]. Available: [https://www.etsi.org/deliver/etsi\\_es/203500\\_203599/203539/01.01.01\\_60/es\\_203539v010101p.pdf](https://www.etsi.org/deliver/etsi_es/203500_203599/203539/01.01.01_60/es_203539v010101p.pdf)
- [21] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware vnf placement for service-customized 5G network slices," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2449–2457.