



**HAL**  
open science

# Total cost modeling of software ownership in Virtual Network Functions

Ghoshana Bista

► **To cite this version:**

Ghoshana Bista. Total cost modeling of software ownership in Virtual Network Functions. Information Theory [cs.IT]. Ecole normale supérieure de lyon - ENS LYON, 2023. English. NNT : 2023ENSL0003 . tel-04074989

**HAL Id: tel-04074989**

**<https://theses.hal.science/tel-04074989>**

Submitted on 19 Apr 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.



Numéro National de Thèse : 2023ENSL0003

**THÈSE**  
**en vue de l'obtention du grade de Docteur, délivré par**  
**l'ÉCOLE NORMALE SUPÉRIEURE DE LYON**

**École Doctorale N° 512**  
**École Doctorale en Informatique et Mathématiques de Lyon**

**Discipline:** Informatique

Soutenue publiquement le 24 Janvier 2023, par :

**Ghoshana Bista**

---

---

**Total cost modeling of software ownership in Virtual  
Network Functions**

---

**Modélisation du coût total de la propriété du logiciel dans Virtual  
Network Functions**

---

---

Devant le jury composé de:

Cérin Christophe	Professeur des Universités	Université Sorbonne Paris Nord	Rapporteur
Myoupo Jean-Frederic	Professeur des Universités	Université de Picardie Jules Verne	Rapporteur
Petit Franck	Professeur des Universités	Université Pierre et Marie Curie Paris 6	Examinateur
Moulierac Joanna	Maître de Conférences	Université Côte d'Azur	Examinatrice
De Palma Noel	Professeur des universités	Université Grenoble Alpes	Examinateur
Eddy Caron	Maître de Conférences HDR	ENS de Lyon	Directeur
Vion Anne-Lucie	Responsable du groupe SAM	Orange S.A	Co-encadrante



# Forward

---

It is essential to state that this Ph.D. work is CIFRE (Conventions Industrialise de Formation par la Recherche) and part of the orange SAM (Software Assets Management) Team. As a part of SAM, our Ph.D. framework is diverse and wide range.

It is the collaboration between ENS Lyon (Lyon, France), INRIA, and the Orange Pessac (Bordeaux, France). During my Ph.D., I traveled between Lyon and Bordeaux, but it was not very often due to the pandemic.

# ACKNOWLEDGEMENTS

---

Getting to the point of writing this acknowledgment section results from the help, motivation, encouragement, and support from many people around me for whom I will be thankful forever.

I would like to express my sincere gratitude to rapporteurs in my Ph.D. committee: Christophe Cerin and Jean-Frederic Myoupo, for reviewing my thesis and giving valuable feedback. I am also thankful to my thesis examiners, Joanna Moulhierac, Noel de Palma, and Franck Petit, for dedicating their valuable time to be part of this thesis committee.

I would like to express my deepest gratitude to my thesis director, Eddy Caron, and supervisor, Anne-Lucie Vion, for their invaluable efforts, personal and scientific guidance, and enthusiastic endeavor during this thesis. I am thankful for providing me with these opportunities, their fruitful discussion, patience, and confidence in me.

I consider myself fortunate to get Eddy as my thesis Director. I am thankful for his consistent support and encouragement throughout our thesis. Eddy is not only a supervisor; he is also my role model to whom I look up. He has an infectious personality, and he teaches me patience, a rigorous manner, and above all, self believing. I am thankful for believing in me, keep pushing me, and motivating me throughout this journey.

I am grateful to Anne-Lucie Vion for introducing me to the industrial world. I have learned how academic knowledge will be used in industries and how different it is. She teaches me how to look at research from an industrial perspective. I am always thankful for her creative critics of my writing, which help to improve my writing skills. Being her first Ph.D. student made this journey more exciting and memorable, knowing that we were both learning how to cope with the challenges that came our way during this journey.

I am fortunate enough to have had the chance to work closely with Eddy and Anne- Lucie Vion. I am thankful to both of my supervisors; despite not speaking French or being familiar with French culture, they believed in me and gave me this opportunity.

I want to express my gratitude to Mme. Noëlle Baillon (former director head of SAM, Orange, France), Arthur Chevalier and Arthur Peyrichou former orange Ph.D. student and trainee, for their benevolent suggestion and guidance, helping me to get an apartment and showing me

beautiful Bordeaux city and teaching me the way of bordelaise.

I would like to thank Mr. Christian Perez, the team leader of the Avalon team and team member of Avalon, for the group meeting, presentation, and valuable discussion.

During this journey, I meet many incredible people in Bordeaux and Lyon. I like to thank all of them for their suggestion, advice, and opinions. Their opinions and guidance not only help me on a personal level they are helpful in my study also. Thank you for being kind and there for me when I need it.

Finally, I would like to express my deepest gratitude to my beloved family for their continuous encouragement and support during this critical phase of my life. First, I would like to thank my parents, Pratap Bista and Shova Baniya, my brother Bihungum Bista, sister-in-law Durga Karki and my nephew Nirvik Jung Bista, who have given me support, strength, courage, motivation, and positive energy. Thank you for being there whenever I need them most. I would like to thank and remember my grandparents, whom I lost during this journey.

*With eternal gratitude to my parents,  
brother and family for always standing by me.*

ॐ सर्वे भवन्तु सुखिनः  
सर्वे सन्तु निरामयाः ।  
सर्वे भद्राणि पश्यन्तु  
मा कश्चिद्दुःखभाग्भवेत् ।  
ॐ शान्तिः शान्तिः शान्तिः ॥

**Om, May All be Happy,  
May All be Free from illness.  
May All See what is Auspicious,  
May no one Suffer.  
Om Peace, Peace, Peace.**

आनो भद्राः क्रतवो यन्तु विश्वतः । ऋग्वेद – 1.89.1

*Let noble thoughts come to us from all directions. Rig Veda 1.89.1*

# Abstract

---

Today, a massive shift is ongoing in telecommunication networks with the emergence of softwarization and cloudification. Among the technologies which are assisting these shifts, one of them is NFV (Network Function Virtualization). NFV is the network architecture that decouples network functions from hardware devices (middleboxes) with the help of a virtual component known as VNF (Virtual Network Function). VNF has shifted the network technological paradigm. Before: Network Function was performed by physical equipment, and service providers acquired its property for the lifetime of the relying hardware (instead counted in years). Today, Network functions are software that service providers develop or acquire purchasing licenses. A license defines software's Right to Use (RTU).

Therefore, if licensing in NFV is not appropriately managed, service providers might (1) be exposed to counterfeiting and risk heavy financial penalties due to non-compliance; (2) might overbuy licenses to cover poorly estimated usages. Thus, mastering network function license through implementing Software Asset Management and FinOps (Finance and DevOps) is essential to control costs. In this research, our primary problem is to minimize the TCO (Total Cost of Ownership) of software cost (VNF), providing Quality of Services (QoS) to a specific amount of users. Software costs include various costs, from development to maintenance, integration to release management, and professional services. Our research focuses on proprietary software (developed by a publisher and sold via a paid license). We considered that TCO consists of the software license cost, the resources necessary to execute and operate SW, and the energy consumed by this execution. In this research, first, we have identified the need for a standardized VNF licensing model, which is highly dependent on the VNF provider's creativity; This lack of standards places CSPs (Communication Service Providers) at risk of having to delegate the management of rights to suppliers. Hence, we proposed a licensing model based on the metrics, which help to quantify the usage of the VNF. After estimating the license of VNF, we estimated the license cost. Afterward, we presented several ways to minimize the license cost depending upon the different use cases, which depend on the user's scenario and needs. Then after, with the help of industrial knowledge, we found that reducing resource con-

sumption to minimize the TCO providing QoS affects the deployment of the VNF directly or indirectly, which impacts the licensing. Thus, the licenses and resources are interdependent. We used these costs to construct the software's total cost. After that, we proposed several ways to reduce the software's total cost by fulfilling the client's requirements. Then after, we considered the energy and its associated cost of VNF. The energy consumption of the VNF is dependent on resource consumption, and resources usages impact the license. Thus, we can see that these three costs are interdependent: license, resources, and energy cost of VNF. Hence, we consider these costs and constructed TCO. Minimizing TCO fulfilling the client's requirements is challenging since it is a multi-parameter. Therefore, we proposed several heuristical algorithms based on resource sharing and consolidation to reduce the TCO depending on the license, resource preference, and the client's scenarios.

# RÉSUMÉ EN FRANÇAIS

---

Aujourd'hui, un changement massif est en cours dans les réseaux de télécommunication avec l'émergence de la softwarisation et de la cloudification. Parmi les technologies qui accompagnent ces mutations, l'une d'elles est la NFV (Network Function Virtualization). NFV est l'architecture réseau qui découple les fonctions réseau des périphériques matériels (middle-boxes) à l'aide d'un composant virtuel appelé VNF (Virtual Network Function). VNF a changé le paradigme technologique des réseaux. Avant : la fonction de réseau était assurée par un équipement physique et les fournisseurs de services acquéraient sa propriété pour la durée de vie du matériel de confiance (au lieu de compter en années). Aujourd'hui, les fonctions réseau sont des logiciels que les fournisseurs de services développent ou acquièrent en achetant des licences. Une licence définit le droit d'utilisation (RTU) du logiciel.

Par conséquent, si l'octroi de licences dans la NFV n'est pas géré de manière appropriée, les fournisseurs de services pourraient (1) être exposés à la contrefaçon et risquer de lourdes sanctions financières en cas de non-conformité ; (2) pourrait suracheter des licences pour couvrir des usages mal estimés. Ainsi, la maîtrise de la licence de fonction réseau via la mise en place du Software Asset Management et du FinOps (Finance et DevOps) est indispensable pour maîtriser les coûts. Dans cette recherche, notre problème principal est de minimiser le TCO (Total Cost of Ownership) du coût du logiciel (VNF), en fournissant une Qualité de Services (QoS) à un nombre spécifique d'utilisateurs. Les coûts des logiciels comprennent divers coûts, du développement à la maintenance, de l'intégration à la gestion des versions et des services professionnels. Nos recherches portent sur les logiciels propriétaires (développés par un éditeur et vendus via une licence payante). Nous avons considéré que le TCO se compose du coût de la licence du logiciel, des ressources nécessaires pour exécuter et faire fonctionner le SW, et de l'énergie consommée par cette exécution. Dans cette recherche, premièrement, nous avons identifié le besoin d'un modèle de licence VNF standardisé, qui dépend fortement de la créativité du fournisseur VNF ; Cette absence de normes expose les CSP (Communication Service Providers) au risque de devoir déléguer la gestion des droits aux fournisseurs. Par conséquent, nous avons proposé un modèle de licence basé sur les métriques, qui aide à

quantifier l'utilisation du VNF. Après avoir estimé la licence de VNF, nous avons estimé le coût de la licence. Ensuite, nous avons présenté plusieurs façons de minimiser le coût de la licence en fonction des différents cas d'utilisation, qui dépendent du scénario et des besoins de l'utilisateur. Puis après, avec l'aide des connaissances industrielles, nous avons constaté que la réduction de la consommation de ressources pour minimiser le TCO fournissant la QoS affecte le déploiement de la VNF directement ou indirectement, ce qui impacte l'octroi de licences. Ainsi, les licences et les ressources sont interdépendantes. Nous avons utilisé ces coûts pour construire le coût total du logiciel. Après cela, nous avons proposé plusieurs façons de réduire le coût total du logiciel en répondant aux exigences du client. Puis après, nous avons considéré l'énergie et son coût associé de VNF. La consommation énergétique du VNF est dépendante de la consommation des ressources, et les usages des ressources impactent la licence. Ainsi, on voit que ces trois coûts sont interdépendants : licence, ressources et coût énergétique de VNF. Par conséquent, nous considérons ces coûts et le TCO construit. Minimiser le coût total de possession en répondant aux exigences du client est un défi car il s'agit d'un multi-paramètres. Par conséquent, nous avons proposé plusieurs algorithmes heuristiques basés sur le partage et la consolidation des ressources pour réduire le TCO en fonction de la licence, de la préférence des ressources et des scénarios du client.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	General Introduction . . . . .	1
1.2	Problem Statement . . . . .	3
1.2.1	Research Problem (Questions) (Main Issues) . . . . .	4
1.3	Contributions . . . . .	5
1.3.1	Report Structure . . . . .	6
1.3.2	Accepted research publications . . . . .	7
<b>2</b>	<b>Software Licensing: Basic, History and Challenges</b>	<b>8</b>
2.1	Introduction Network Function Virtualization (NFV) . . . . .	8
2.2	Software license . . . . .	14
2.2.1	Introduction to software cost . . . . .	14
2.3	History of Software license . . . . .	17
2.3.1	Different types of license classification . . . . .	20
2.4	VNF license . . . . .	21
2.5	SAM (Software Assets Management) . . . . .	23
2.6	Conclusion . . . . .	26
<b>3</b>	<b>Modeling of Software Cost For VNF</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Literature Review: Software cost and Existing State . . . . .	28
3.3	Cost Modeling: RC, LC using Use-cases . . . . .	32
3.3.1	Different VNF providers Use Case (Case I) . . . . .	33
3.3.2	Flavor based Use Case (Case II) . . . . .	34
3.3.3	Flavor and SAU combination Use Case (Case III) . . . . .	35
3.3.4	SAU, Flavor and Cost combination Use Case (Case IV) . . . . .	36
3.4	Conclusion . . . . .	37
<b>4</b>	<b>Total Cost Modeling For VNF Based On Different Use Cases And Scenario</b>	<b>38</b>
4.1	Introduction . . . . .	38

4.2	Related Works . . . . .	40
4.3	Proposal Model . . . . .	41
4.3.1	Traditional ways . . . . .	41
4.3.2	Cost model for Virtual Network Function . . . . .	43
4.4	Use cases with different scenarios . . . . .	44
4.4.1	Scenario 1: VNF instances . . . . .	45
4.4.2	Scenario 2: Users . . . . .	46
4.4.3	Scenario 3: Nodes . . . . .	48
4.4.4	Node scenario . . . . .	50
4.4.4.1	Single Node . . . . .	50
4.4.4.2	Result evaluation for single node . . . . .	50
4.4.4.3	Multiple Nodes . . . . .	53
4.4.4.4	Result Evaluation for multiple nodes . . . . .	54
4.5	Conclusion . . . . .	59
<b>5</b>	<b>VNF Software Cost Modeling Based On Telecommunication Network</b>	<b>61</b>
5.1	Context . . . . .	61
5.2	Introduction . . . . .	62
5.3	Related Work . . . . .	63
5.3.1	System Model . . . . .	64
5.3.1.1	Software cost Analysis . . . . .	64
5.3.2	Sharing and Consolidation Algorithm . . . . .	68
5.4	Proposed model . . . . .	69
5.4.1	Deployment Flavor (DF) . . . . .	71
5.4.2	Use case . . . . .	72
5.4.3	Utility Based . . . . .	75
5.4.4	Simulation setup and evaluation . . . . .	76
5.4.4.1	Evaluation of Results . . . . .	77
5.5	Conclusion . . . . .	83
<b>6</b>	<b>Conclusion</b>	<b>84</b>
6.1	Summary . . . . .	84
6.2	Contributions . . . . .	85
6.3	Future perspectives . . . . .	87
	<b>Bibliography</b>	<b>89</b>
	<b>List of Figures</b>	<b>99</b>





# Introduction

*“You are only entitled to the action,  
never to its fruits.”*

— Bhagavad Gita.

## Contents

---

1.1	General Introduction . . . . .	1
1.2	Problem Statement . . . . .	3
1.3	Contributions . . . . .	5

---

## 1.1 General Introduction

Virtualization and cloudification of telecom network architecture and IT infrastructure change the traditional telecom network and IT infrastructure to a layered model by replacing physical equipment with virtual network functions running on standard servers platform [1]. Firmly merging the Telco and IT infrastructure softwarization has revolutionized current network technologies and opened the gates for many opportunities and complexities. In present, Software Define Networking (SDN) and Network Function Virtualization (NFV) are commonly practice virtualization architecture.

NFV is proposed by European Telecommunication Standards Institute (ETSI) in 2012 [2] with the promises that it will optimize Capital Expenditures (CAPEX) and Operational Expenditures (OPEX). CAPEX is an investment invested by the company to acquire NFV and its services. It stands for Capital Expenditures. OPEX is the operational cost paid by the enterprise to operate the NFV. It stands for Operational Cost. NFV decouples the hardware and software, which reduce vendor lock-in and bring agility to the services by efficient resources life cycle

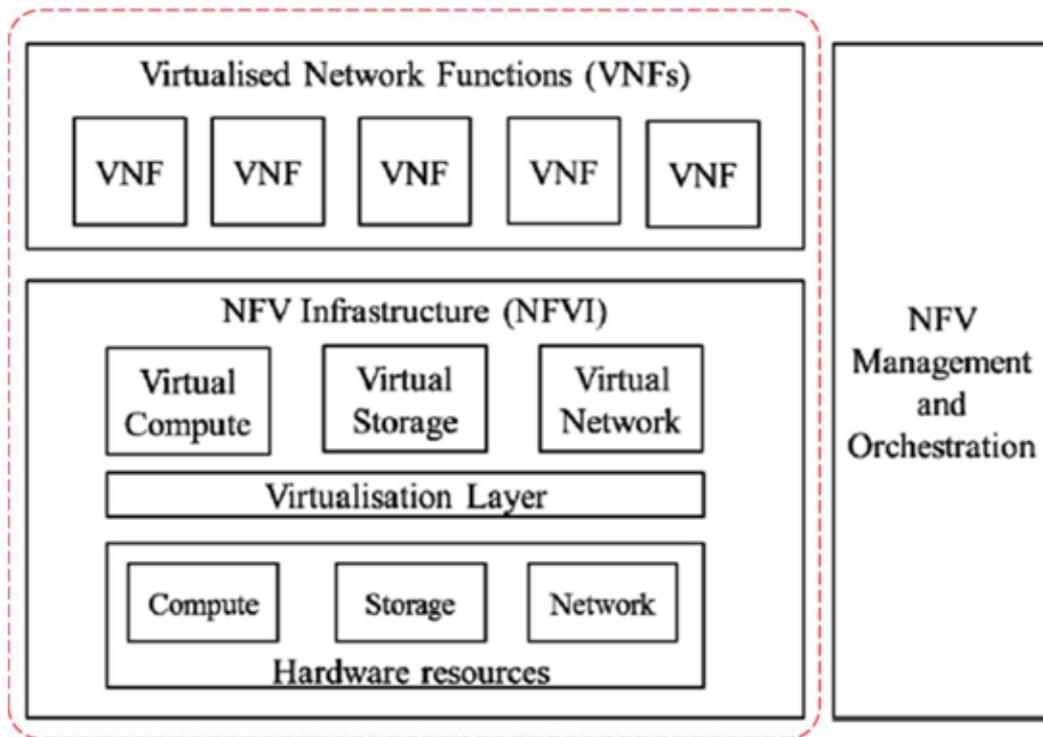


Figure 1.1: High level NFV framework [3]

operation and management, engraving the way for virtualization of the network. VNF (Virtual Network Function) is the virtual component of the NFV that is capable of operating network functions in virtual manners such as Intrusion Detection System (IDS), firewall, and Network Address Translation (NAT). It allows flexibility, efficient design, deployment, and management of network services, and setting free the Telco industries from vendor syndicate and lock-in. Furthermore, NFV help to separate the Control plane and Data plane, which play a crucial role in the SDN (Software Defined Network). Optimal placement of the Control plane and Data plane not only improves the Quality Of Services (QoS), such as latency, delays, and throughput but also serves an important role in network security. NFV has three main layers or domains, as shown in Figure 1.1. NFV Infrastructure (NFVI) is the combination of physical and virtualized resources to support the execution of VNFs. Management and Orchestration (MANO) includes the orchestration and life cycle management of virtual or physical resources (that support the life cycle management of VNF), and VNF is the virtual component that implements network function in software form.

## 1.2 Problem Statement

NFV market is projected to grow from 12.9bn in 2019 to 36.3bn dollars by 2024<sup>1</sup>. Service providers and clients need to adapt to these shifts quickly and efficiently. Virtual Network Function (VNF) is a virtual component of NFV that operates the network function in software form with the help of NFVI. As networks become software failures controlling software spending destroys the promises of NFV efficiency. Today's enterprises are investing a huge (around 30-50) percent of their investment in software; thus, proper software management is not an option. Understanding this, we considered our research major problem to minimize the cost, Total Cost of Ownership (TCO) of software (VNF), by providing agreed QoS to the clients. TCO consists of lots of costs from human resources to the application (software). In this research, we considered the three different costs, license, resources, and energy.

It is vital to monitor the usage of VNFs based on the license agreement to obtain the economic benefits of NFV long or short terms planning. To meet the continuity of the service, usage of VNF licenses needs to be handled proactively. NFV-MANO uses the license for the VNF instantiation and operation [1]. As SW (Software) is protected by Intellectual Property Right (IPR) over a license, it becomes essential to ensure the compliance of SW deployments regarding the acquired rights. Furthermore, optimizing license costs to obtain the benefits from NFV for clients is also essential. For this, one of the standard practices in IT extending to the network is to practice Software Assets Management (SAM). Implementing SAM guarantees the avoidance of counterfeiting, waste, and risk of non-compliance.

SW license frames the rights and obligations of the Communication Service Providers (CSP) to use SW. License is associated with a cost (License Cost (LC)) which depends on the volume of rights granted. The volume granted and associated conditions of use are contractually defined by one or several metrics. There are currently no standards on metrics and their definition; they depend on the creativity of software providers. It could facilitate every supplier of VNF to propose their metrics, model, and tools.

In an industrial context, orange Telcom (it can be any telecom operator) does not want to be controlled by a third party, a VNF provider. Thus, not to lose sovereignty, standardization of metrics is essential. Therefore, we considered another research problem (sub-problem 1) to ease these difficulties in modeling an efficient VNF licensing method.

Resources are required to operate the software, VNF. For our study, we consider vCPU and vRAM as resources. When resources are minimized to reduce TCO by providing agreed QoS, this will affect the way the VNF is deployed and ultimately affect the VNF license. Also, based on our observation of the telcos industries, VNF metrics can be linked, along with usage, resource allocation, and resource consumption. Thus, it is not wiser to consider LC and Resource

---

<sup>1</sup><https://www.f5.com/fr/fr/company/blog/why-nfv-is-more-relevant-than-ever>

Cost (RC) independently, but we need to consider that LC can depend on resources and vice versa. Thus, our next problem (sub-problem 2) is to minimize total cost considering resources and evaluate its impact on the license depending on the scenario and the client's needs.

Another aspect of TCO is energy. The energy consumption of VNF is associated with resource consumption and metrics. However, these metrics are mainly user-specific or resources dependent. Moreover, energy consumption depends on the resources being used. Thus, to reduce energy resource uses need to be minimized. Therefore, we can see that TCO minimization is not trivial and is multi-parameters. To address these problems, we formulate the following Research Questions.

### 1.2.1 Research Problem (Questions) (Main Issues)

As we already stated, The problem statement of this Ph.D. is how to minimize the Total Cost of Ownership of VNF software that satisfies a given Quality Of Services (QoS) for a fixed number of users. Therefore, it appeared that the following Research Question (RQ) should be investigated and solved;

- **RQ1) Software Costs formulation:** VNF Software cost includes a wide range of costs like software updates, maintenance cost, license cost, upgrade cost, resources (virtual, hardware) cost required to run VNF, and human resources. So our question is, What are the costs we will consider for the modulation of VNF software cost?
- **RQ2) License module and cost formulation:** How to compute efficient and authentic licenses for VNF that can be used to estimate license and other costs?
- **RQ3) Finding others interdependent cost:** What are the other costs depending on the license of VNF?
- **RQ4) How to minimize Total cost and TCO:** After determining the other interdependent costs like resources (virtual in our case) and energy, how can these costs be minimized so that the Total Cost of Ownership (TCO) of software cost will minimize?

Our work is not only limited to licensing. We tried to work for the sovereignty of companies. The current situation is that organizations must share their valuable data with the VNF provider for proper software management, creating a cold environment between the service provider, VNF provider, and VNF users. Organizations charged for using VNF or VNF services are in the dark they do not know how they are being charged. Thus, with our work, we are not only proposing estimating and minimizing TCO and other costs; we are trying to provide confidence, sovereignty, and freedom to clients using the VNF.

Also, our framework is not limited to NFV; it can be implemented on SDN, network slicing, container, and different network services.

### 1.3 Contributions

This thesis focuses on software cost optimization in VNF, considering license cost and resources cost. Our thesis can be categorized into three parts, first is analyzing the existing literature, finding gaps, and pointing out future research directions for software cost in a virtual network. The second is modeling the software cost using various scenarios or use cases for VNF, and finally, minimizing the Total cost of ownership. The Scientific Contributions (SC) of this thesis can be categorized as follow;

- **(SC1) Reliable and efficient modules for license estimation and other interdependent costs:** NFV revolutionized the way network services are deployed. It saves lots of revenue by replacing lots of hardware with virtual functions, *softwarization*. One of the major problems is how to license these virtual functions efficiently.

Thus, Our first contribution is we propose a possible efficient license model based on our knowledge acquired from the industries using the SAU and BW metrics, use cases, and scenarios. We show that licensing with these metrics will reduce the cost and work as a convenient model to charge the clients, or VNF users can use it to estimate the license cost and others costs. For this study, we have tried to explore the existing problems and complications in VNF licensing/cost and recommend a novel idea using use cases that help to optimize the VNF software cost. We present the different ways to minimize the license and resources cost using different use-case and scenarios based on client needs and requirements.

- **(SC2) Cost model for telecom virtualized networks based on VNF:** The telco industry struggles to converge and standardize licensing and cost models. At risk: the network cloudification benefits could be swept away by poor management of resources (hardware and software). Also, very few works have dealt with the economic dimension of softwarisation. Thus we presented a preliminary model for optimizing the total cost of a VNF based on the RC and LC. This analysis is inspired by measurement and licensing practices commonly observed in the Telcos industries.

We estimate the license cost using our model based on the license and unit cost. Also, in an industrial ecosystem, considering license cost independently is not logical because the license is dependent on resource usage and vice versa. Thus, we formulate the total software cost using license and resource costs.

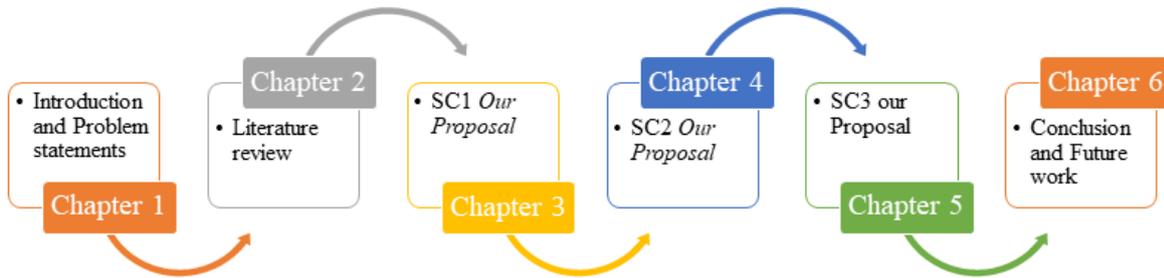


Figure 1.2: Chapters Arrangement

- **(SC3) Heuristical models for minimization of Total cost of Ownership:** Our third contribution is reducing the Total Cost Of Ownership (TCO) based on resource sharing and consolidation of VNF resources using a heuristic algorithm. TCO of software cost of VNF is based on three crucial costs LC, RC, and EC.

LC is associated with software rights granted (and consumed) to the user. Therefore, 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. The result from our simulation verified that depending upon the client's needs, whether it is to reduce license cost or resources cost, or both, they can choose any algorithm from the three heuristic algorithms we proposed. We also presented the use case to show how a user can minimize energy consumption based on various VNF providers minimizing environment impact.

### 1.3.1 Report Structure

The chapters are grouped into six parts: Background and introduction, literature overview, proposal of our algorithm, models, and end conclusion. The organization of the thesis is shown in Figure 1.2.

**Chapter 2** presents Virtual Network Function (VNF) introduction, significance, and history. First, we introduce the basic concepts of NFV and its virtual components, VNF, and flavors. In this chapter, our focus is on the existing state of the software cost. We first present a brief history of software cost evolution, from device lock software to softwarization. Then,

we present the state of the art of VNF based on software cost and resource allocation. From this, we identify the current research gaps in existing research and position our work in the framework of software cost. Additionally, we present the foremost reason for undertaking the software cost aspect of VNF and its licensing.

**Chapter 3** Next, we module software costs with the help of LC and RC. Then, through our different use cases, we show how the software cost of VNF can be minimized and how the users can decide which VNF they need depending on their requirements, scenarios, and metrics.

**Chapter 4** After presenting a way to minimize the software cost, we also present how to formulate and minimize the total software cost. Then, we propose efficient ways to estimate the necessary amount of licenses for the VNFs based on metrics. Afterward, with the help of this estimated license, we analyzed LC and Resources cost (RC). Next, we implement these modules in node and use cases that verify that these models help to minimize total software cost.

**Chapter 5** In this chapter, we present another contribution considering the energy attribute of VNF. Energy has a significant impact on carbon footprint, and energy efficiency is one of the benefits of NFV. Thus, we consider energy costs, licenses, and resources to construct our software TCO. After formulating a model to estimate TCO, we proposed an algorithm that is based on resource sharing and heuristic methods for the minimization of TCO. Finally, we validate our methodologies using use cases and implementing them in the 5G core network.

**Chapter 6** Finally, this chapter will conclude our research by presenting future perspectives and conclusions. In this, we present the possible future research direction using our research as footsteps. Also, we briefly analyze and present the possible challenges that need to be addressed. Then, we conclude our research by presenting our research conclusion.

### 1.3.2 Accepted research publications

1. G. Bista, E. Caron, and A.-L. Vion. A study on optimizing VNF software cost. In Global Information Infrastructure and Networking Symposium, Tunis, Tunisia (virtual conference), October 28-30 2020. hal- 03027945.
2. G. Bista, E. Caron, and A.-L. Vion. Total cost modeling for VNF based on licenses and resources. In Scitepress, editor, CLOSER 2022. 12th International conference on Cloud computing and Service Science., ISBN 978-989-758-570-8, pages 246–253, Online Streaming, 27-29 April 2022. 8 pages. hal number hal-03660857.
3. G. Bista, E. Caron, and A.-L. Vion “ VNF Software Cost Modeling Based on Telecom Network” Accepted in The Ninth International Conference on Software Defined Systems (SDS-2022)

# Software Licensing: Basic, History and Challenges

*“Everybody know the use of the useful,  
but nobody knows the use of the  
useless!.”*

— Zhuangzi

## Contents

---

2.1	Introduction Network Function Virtualization (NFV) . . . . .	8
2.2	Software license . . . . .	14
2.3	History of Software license . . . . .	17
2.4	VNF license . . . . .	21
2.5	SAM (Software Assets Management) . . . . .	23
2.6	Conclusion . . . . .	26

---

## 2.1 Introduction Network Function Virtualization (NFV)

In traditional networks, vendor-specific software and hardware is used as a combination for the implementation of Network Functions (NFs). The term “middlebox” was widely used in the networking universe to provide telecommunication and networking services for a few decades. As networking services expand, the popularity and the use of middlebox increase. As a result, complexities increase in services and middleboxes. In addition, scaling up or down to resources

in the middle box is time-consuming and highly expensive [4]. Figure 2.1 shows how the NFV changed the traditional model from hardware to virtual network function.

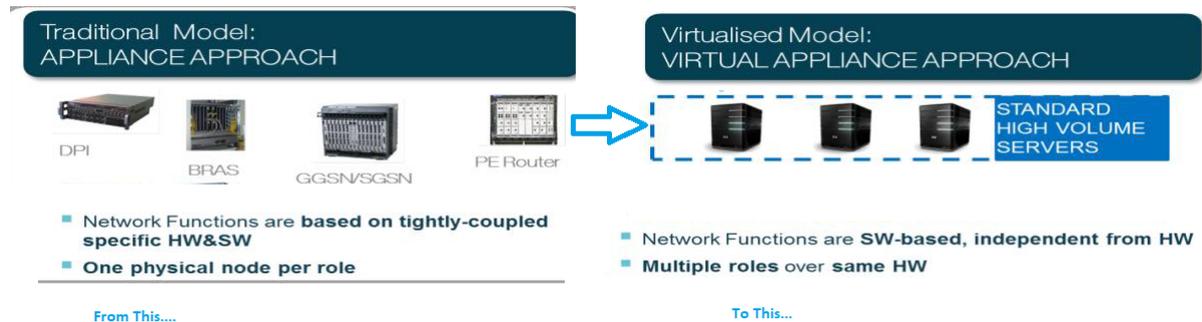


Figure 2.1: Network Function Virtualization

NFV-based architecture is expected to reduce OPEX and CAPEX by minimizing the use of specialized hardware (middlebox). Also, it gives full or partial control to a service provider, organization, and users to manage the network functionalities. The fundamental idea of NFV is to replace hardware middlebox such as firewalls, NAT, IDS, and proxy with virtualized functions in a virtual infrastructure. NFV helps to decouple the software from the hardware; this helps to enable the software and hardware to progress separately. Furthermore, NFV assists in flexible network function deployment since the software and hardware are detached; they can perform different functions simultaneously. This helps to deploy new services faster.

NFV architecture has different benefits, some of which are briefly listed below;

- **Hardware Flexibility:** In NFV, Telco or network operators can choose and build hardware according to their requirements and needs because NFV uses Commercial-Off-The Shelf (COTS) hardware.
- **Efficient deployment:** Network services can be deployed more quickly based on demand and on-need of clients. Which is crucial to obtain or providing uninterrupted services
- **Scalability:** With NFV, network services can be easily scaled up or down depending upon the demands.
- **Reduce Capital Expenditures (CAPEX) and Operational Expenditures (OPEX):** Decreasing the use of hardware and increasing the automation operation by NFV help to reduce CAPEX and OPEX
- **Energy efficient:** Proper consolidation and sharing of NFV resources help reduce energy consumption and the service's complexities.

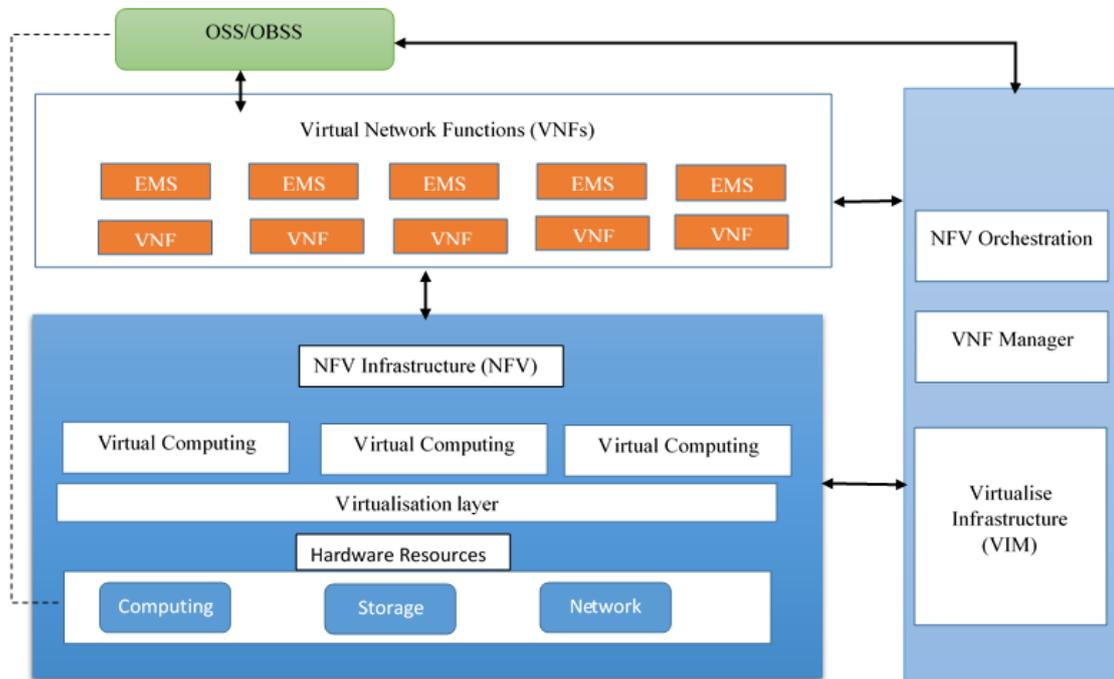


Figure 2.2: NFV Reference Architecture Framework [3]

Figure 2.2 shows the NFV framework, which consists of different functional domains presented by ETSI.

Some of the important functional domains of NFV and their function are;

- **VNF**: This helps to implement network function as the software which operates over the NFVI.
- **NFVI**: is the combination of virtual and physical resources that help operate the virtual network functions.
- **Element Management System (EMS)**: This helps to manage and controls the functionality of the VNF.
- **NFV Management and Orchestration (NFV MANO)**: It includes the orchestration and life-cycle management of physical and software resources that support the infrastructure virtualization and life cycle of the VNFs. MANO includes the following functional domains;
  1. **Virtualized Infrastructure Manager (VIM)**: It is the virtual infrastructure to perform network services such as OpenStack or Kubernetes.

2. VNF Managers: It manages the VNF life cycle, including VNF instantiation, update, upgrading, scaling, and termination.
3. NFV Orchestration (NFVO): It helps to orchestrate the optimum uses of physical hardware, NFVI, and software resources for efficient services.

Among different layers of NFV, VNF acts as software. VNF consists of VNF Components (VNFC) which consist of many descriptions including Deployment Flavor (DF) (capacity, storage, Virtual CPU (vCPU), Virtual RAM (vRAM) as shown in Figure 2.3. Which is generally encapsulated in VNFD (VNF Descriptor). In our research, we use the DF for the VNF cost analysis. VNFD defines VNF properties such as;

- a) Resources needed.
- b) Connectivity as External Connection Point (CP), Virtual Links Descriptor (VLD), Internal Connection Points (CP).
- c) Life-cycle management characteristic (instantiation, scaling)
- d) VNF specific Parameters affinity, anti-affinity.

Another significant and beneficial feature of VNF is VNF scaling. Scaling the VNF means increasing or decreasing the capacity of the VNF, depending upon the situation and scenario. VNF can be scaled in two ways:

- Scale up/down: It is known as vertical scaling. It is increasing or decreasing the resources allocated to VNFC instances in the VNF instances.
- Scale out/in: It is adding/removing VNFC instances to a VNF instance.

Since there is always confusion and curiosity about SDN and NFV, we showed the high-level difference and how they can complement each other. Nevertheless, the reader should not forget that this is not our research objective; this is just for a clear understanding of two architectures.

- SDN decouples the control plane from the user/data/forwarding plane.
- NFV decouples the software from the hardware.

Let us see the high-level figure for the apparent difference. The figure presented below didn't include all the elements of both architectures.

In the Figures, we see that from Figure 2.4a to Figure 2.5b how traditional, NFV, and SDN architecture are different from each other using Control Plane and Users Plane. SDN does not need NFV and vice versa, but together they can complement each other

For more clearance and understanding. We include a few answers to FAQ from the ETSI [5];

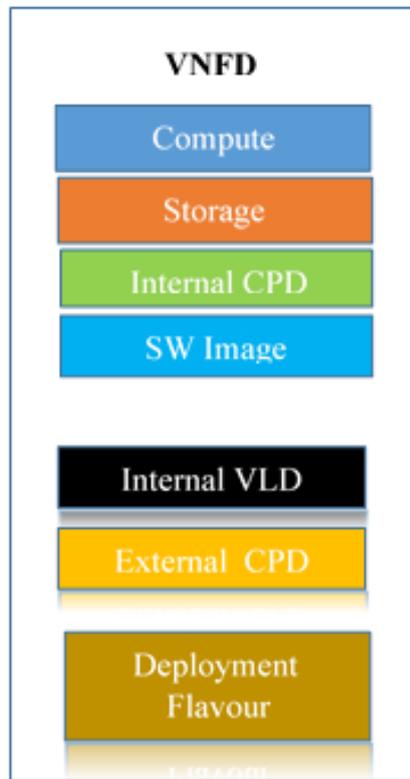
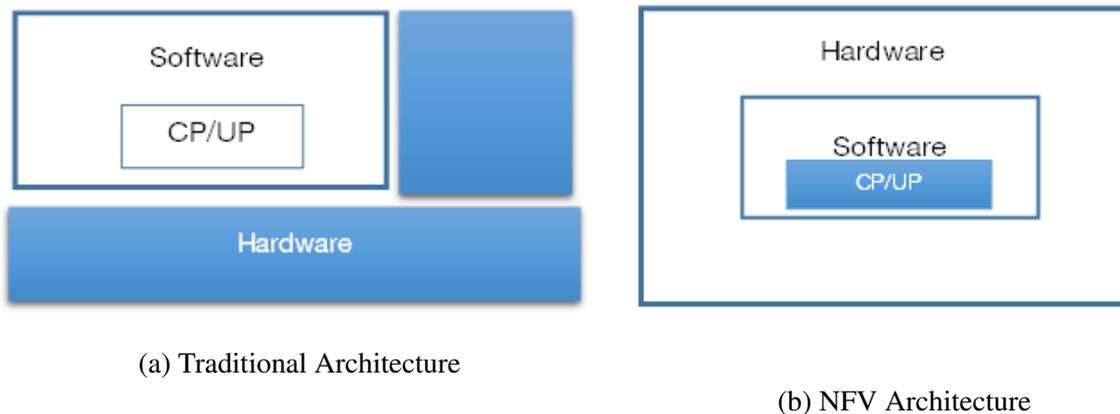


Figure 2.3: VNF Descriptor



(a) Traditional Architecture

(b) NFV Architecture

Figure 2.4: Traditional to NFV

**1. Will VNFs be soon replaced with CNFs?**

No. The question should not be put in these terms. The VM- base VNF deployments will be complemented with Container-based deployments. IT will give VNF designers and

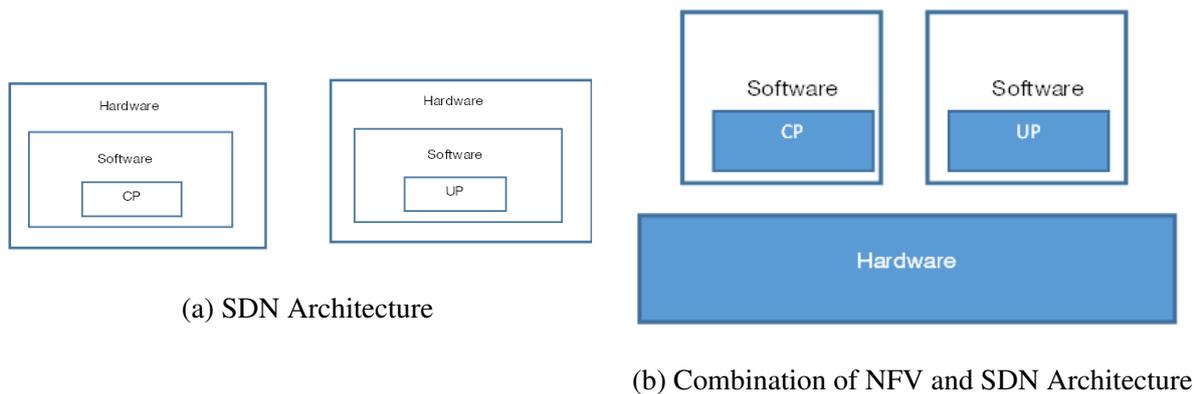


Figure 2.5: SDN and NFV

network operators more choices while using a common management system. CNF stands for Cloud-native Network Function. CNF is a particular type of VNF. It's a VNF that is designed, deployed, and managed using cloud-native technologies. This, in practice, means that the VNF is deployed using container technologies.

**2. What is the best practice for VNF deployment? Is it VN base or container-based?**

Most of the current Telco NFV deployments are based on VMs, which suits most of the VNF implementations currently provided in the market. However, more recent implementations are starting to introduce VNFs based on containers. In addition, container-based virtualization is getting more consideration due to the industry momentum introduced by the requests for cloud-native VNFs. One benefit of container technologies is that they can run more VNFC instances on the same physical host. These instances start quicker since there is less overhead compared to VM-based technologies.

**3. Does NFV focus only on virtualizing the telecom network functions?**

Any enterprise or environment that shares the same characteristics and the requirements of NFV (For example, carrier-grade performance, high availability, and telco-like workloads) can leverage the potentials of NFV and the framework defined by ETSI NFV.

**4. Does PNF dependability differ from VNF resiliency?**

The two terms cover the same concept: the ability to limit disruption and return to normal or, at a minimum, acceptable service delivery level when facing a fault, failure, or even that disrupts the regular operation.

Now, we know that VNF behaves as software; like the software protected by a license, VNF is also protected by a license. Let us see about the software license.

## 2.2 Software license

### 2.2.1 Introduction to software cost

There are multiple definitions of the software in use. For our research, the software is a set of programmed instructions stored in memory that the processor executes. The first person to come up with the theory of software in 1935 was Alan Turing <sup>1</sup>.

Traditionally software cost means the purchase price of the software. Users can buy and use the software for as long as they want. Slowly concepts of software maintenance, updating, upgrading, and licensing of software arise. Today software cost includes a wide range of costs from HR to technical, professional services, updates (maintenance), upgrading, license cost, and devices cost, as shown in Figure 2.6. In broader aspects, software costs include the cost from when the software started to develop until the software is completely deployed in the respective devices or nodes. In this research, we consider VNF, so the software cost of VNF for us includes resources, license, and energy costs. We describe this in detail in upcoming sections and chapters. In some cases, software cost has been used interchangeably with license cost.

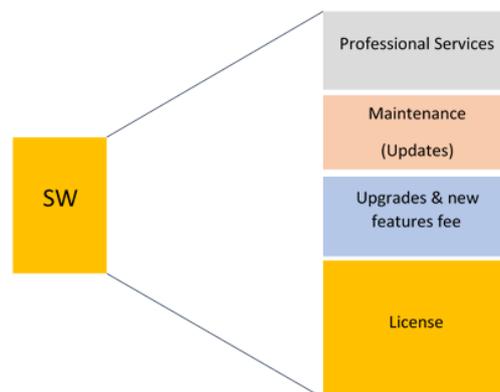


Figure 2.6: Software Cost Classification

Let see the few terminologies related with Software and its cost;

- **Software license:** It is the legal right to use the software under the terms and conditions specified by the software licensor<sup>2</sup>.
- **Software licensee:** The person or organization granted a license to use a specific software product<sup>3</sup>.

<sup>1</sup><https://www.iso.org/standard/63598.html>

<sup>2</sup><https://www.iso.org/obp/ui/iso:std:iso-iec:19770:-5:ed-2:v1:en,3.41>

<sup>3</sup><https://www.iso.org/obp/ui/iso:std:iso-iec:19770:-5:ed-2:v1:en,3.42>

- **Software licensor:** A person or organization holds the right to issue a software license for a specific software package. This person or organization may or may not create the software <sup>4</sup>.
- **Software cost estimation:** It is a complex activity that demands knowledge of several key attributes about the projects for which the estimate is being constructed [6]. It is even termed “parametric estimating” because many parameters that affect software outcomes need to be understood to estimate software cost accurately. Parameters can be software specifications, bugs or defects that are likely to be encountered, capabilities of the development team, license, methodologies that will be utilized, cost and schedule, and metrics constraints set by the client’s needs and scenarios.

Software cost estimating has been a crucial but problematic task since the beginning of the computer era in the 1940s [6] as the size and importance of software applications have grown. As a result, the need for accuracy in software cost estimating has grown so much. In this research, VNF is software, and we use various methodologies to evaluate and optimize its software cost using different metrics, such as resource allocation, consumption, and licenses.

- **License Cost (LC):** LC is the cost to use the software according to terms and rights. Licensing is a very complex task. It depends on one or several metrics, scenario but the irony of it is that it never receives proper recognition in the academic stream. Very few researches have been conducted for the sole purpose of licensing. However, in contrast, in the industrial discipline, it was, is, and most probably will be regarded as one of the crucial factors for ROI (Return Of Investment).

Most researchers or people generally have a myopic view of the license; it is just a unit cost. However, it is not just unit cost. It depends on various costs and factors of any software. In our context, it is a VNF license. It can be seen in Figure 2.7. Where the tip of the iceberg is compared to the other hidden cost or factors, the parameters it impacts, and it gets the effect.

- **Resource cost (RC):** These are the cost related to the resources to operate the software. It can be virtual resources or physical hardware. In our case, we include virtual resources to serve the VNF that depend on the licensing of VNF, scenarios, and other metrics used to measure the license or usage of VNF.
- **Total cost:** Total cost is an economic analysis that sums up all expenses, including the production of a product and equipment. It is the combination of all fixed costs and

---

<sup>4</sup><https://www.iso.org/obp/ui/iso:std:iso-iec:19770:-5:ed-2:v1:en, 3.43>

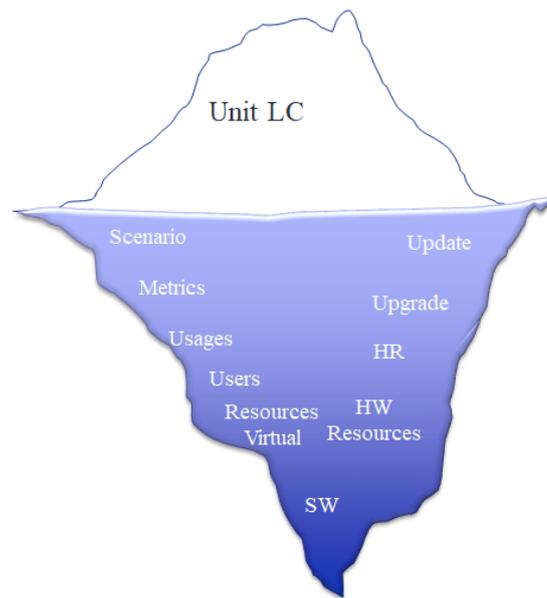


Figure 2.7: License Cost and its hidden impacts.

variable costs.

- **Fixed Cost:** These are costs that are fixed for a specific duration of time, generally 1 to 5 years. In NFV, these include license, software cost, and the components of Hardware (HW) and Software (SW).
- **Variables Cost:** These are the cost that is not fixed. They keep changing frequently; this can be traffic cost, link cost, Human Resource cost (HR), and offers by service providers. In our research, the total cost is constructed using the License Cost (LC) and Resource Cost (RC). We present the total cost in Chapter 3. LC can be considered here as the fixed cost, but not always, and for RC, it is not so easy to classify between variable and fixed because VNF resources can be scaled up and down quickly, which impact the license of VNF. Thus, it is not convenient to classify LC and RC are fixed costs or variables; rather, it depends on the contractual agreement between users and service providers.
- **Total Cost Of Ownership (TCO):** It includes all the associated costs acquired by a buying enterprise during purchasing goods or services from external providers [7]. It helps quantify the purchase's cost throughout the product's lifecycle. It forms the factual basis for determining the value cost vs Return On Investment (ROI). Overall, TCO considers all the direct and indirect costs during the software manufacturing process. It is an essential tool to ensure the cost savings of an organization.

One of the essential benefits of NFV is that it helps to reduce the CAPEX and OPEX by minimizing the use of hardware with virtual functions. In NFV, TCO is the summation of

the CAPEX, and OPEX can be categorized as;

- The total cost of ownership OPEX and CAPEX
- $OPEX = \text{Services cost} + \text{Fixed cost} + \text{Variable cost}$
- Services cost = maintenance, software service, HW services, etc.
- Fixed cost = license + software cost + component of HW /SW etc
- Variable cost = Traffic cost + link cost + Services, offers, etc.
- $CAPEX = \text{Services cost} + \text{Fixed cost} + \text{Variable cost}$  Services cost = maintenance, software service, HW services, etc.
- Fixed cost = license + software cost + component of HW /SW etc.
- Variable cost = Traffic cost + link cost + Services, offers, etc., equipment cost.

Generally, TCOs are negotiated through the Request For Proposal (RFP process) (Sourcing) for five years. We can see that TCO consists of various costs. TCO can be summarized as shown in Figure 2.8. TCO is the summation of CAPEX and OPEX. For the convenience and granular analysis of the costs in this research. The TCO is constructed using the LC, RC, and EC summation, which we will see much more in Chapter 4.

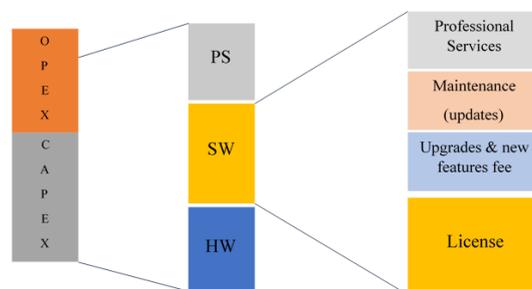


Figure 2.8: OPEX and CAPEX General Classification

## 2.3 History of Software license

If we trace back the history of software licensing. We will find that software licensing concerns increase as technology expands its horizons. Around 1980 users needed to buy the products and get a license to use them. However, from the early 90s, software licensing became a substantial concern as floppy disks became popular. In the beginning, software licenses were often "node-locked," so they would run only on one device. Still, in node-locking, complexities arose as

the number of packages increased [8]. To ease these issues at the 1987 "networking license server" by Apollo computer and in 1988, "FLEXIm" from Highland software was introduced in the commercial license manager world [8]. As commercial licenses were getting a heat, open-source software licenses, such as GNU and OpenBSD (Berkeley Software Distribution), were also getting their market. By the end of 1999, software used had risen astronomically, so organizations desperately needed tools that assisted in software management. From this point's need for SAM arose.

In 1999, N.F hosting et al. proposed a software probation model which helped to categorize the organization's five existing problems that needed to be addressed [9]. They were legal (counterfeiting), ethical (intellectual property proper respect), managerial and economic issues (true-up cost), and technical (monitoring). Now, the scenario is gradually changing. The software, which is coupled with hardware, is starting to decouple. That means the software is no longer related to the hardware; they are all virtualized. As the technologies evolve, licensing models we are using today will not be sufficient because of the vastness of the traffic and usage. Especially in NFV, there is a vast diversity of license management tools across the software industries. Each VNF provider and service has a different licensing, enforcement, and rate of charge for the software. It makes services provisioning and license renewal operations more complex, error-prone, and time-consuming [10]. These issues can be resolved by using the standard NFV license management architecture.

In point of a telco company, as an orange. The present situation is that we buy VNF from VNF providers either they proposed their license tools, or we need to hire a third party, which is trusted by both the VNF provider and the company, for the license management. Thus, the problem that arises due to the lack of standardized methods are:

- We need to share our data with a third party.
- We don't know how we (as an enterprise) are being charged, license management is like a "**black box**" for the company in the present scenario.
- We lose our sovereignty and freedom. We need to invest extra investment for license management which increases our expenses.
- Since a third party conducts license management, there is always the chance of a security breach. Moreover, our valuable data might not be secure at all. Also, the average audit takes 194 working hours and takes around 7 months to complete. These are the reasons for orange, a telco company, to push these kinds of research.

A software license differs according to the rights granted to use the software. Classification of license generally depends on the entitlement of license. In software licensing, there are two crucial actor licensor and the licensee.

- VNF licensor: It is the organization or the person who holds the right to issue a VNF license for a specific VNF package [1].
- VNF licensee: They are the person or organizations granted a license to use a specific VNF [1].
- License entitlement: A VNF license entitlement consists of the information about software, in our case, VNF rights, limits, and agreed metrics between the licensor (VNF provider or service provider) and the licensee (service provider, client). This licensee, licensor, and actor can be further classified into three actors, as shown in Figure 2.9.

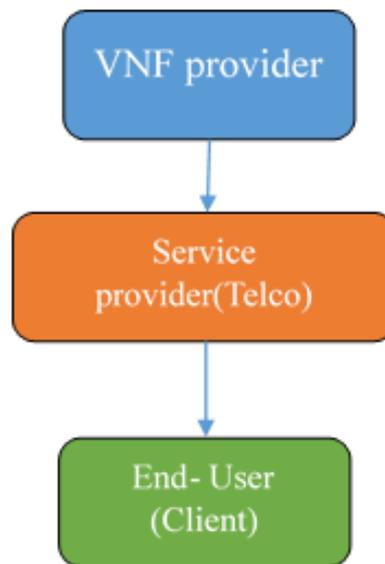


Figure 2.9: License Management Actors and Different Roles.

- VNF provider: These are the entity that provides the VNF. It develops and issues the VNF packages to the service provider.
- Service Provider (SP): These entities provide the services on the VNF under license entitlement.
- Clients: These are the actor using the services SP provides in VNF.

NOTE: Single entity can play several roles, VNF providers can be service providers, and services providers can be clients. So in NFV, it is not easy to categorize. It depends on several factors, such as scenario and metrics.

VNF license entitlement can evolve through transactions between VNF providers and service providers. This evolution can be due to deployment or increasing/decreasing VNF capacity. VNF license entitlement help to enforcement of the VNF license. There are

different types of license entitlement enforcement among these declarative and nominative are popular in industries, we present briefly about these two here. Software licenses are also distinguished by the way the compliance with the license entitlements is implemented.

- Declarative: The term declarative license denotes a license in which the licensee is responsible for implementing this compliance and demonstrating it to the licensor. This is the preferred model for Orange (and already applied in IT with most publishers) because the SAM process must be in control to perform cost optimization.
- Nominative: In the nominative license (primarily used for public software), a license key is associated with the licensed software. Which allowed its authentication and also used by the licensor to authorize (or not!) the use of the software. This method presents several risks (the most important being a risk of interruption of service).

### 2.3.1 Different types of license classification

There are lots of business models for licenses today, and many of them can be used. Some of them are depicted in Figure 2.10 in the context of NFV [1]. Generally, it can be subdivided into two categories:

- a) Open Sources software license.
- b) Proprietary software license

Rights	Public domain	Free and Open Source Software (FOSS)				Proprietary	Trade secret
		Open Source (Without copyleft)	Open Source (Weak copyleft)	Open Source (Strong copyleft)	Freeware, Shareware, Freemium		
Examples	SQLite, ImageJ	Apache, BSD, MIT	GNU LGPL, MPL	GNU GPL, GNU Affero	Irfanview, Winamp	Windows	World of warcraft
Copyright retained	No	Yes	Yes	Yes	Yes	Yes	Yes
Right to use	Yes	Yes	Yes	Yes	Yes	Yes	No
Right to copy	Yes	Yes	Yes	Yes	Often	No	No
Right to modify	Yes	Yes	Yes	Yes	No	No	No
Right to distribute	Yes	Yes, under same license	Yes, under same license	Yes, under same license	Often	No	No
Right to sublicense	Yes	Yes	No	No	No	No	No

Figure 2.10: Overall classification of software license .

- a) Open Source software license : These licenses allow the source code to be available according to the license terms that would enable the licensee to use, modify, and distribute this code. Based on the copyleft criteria, these licenses can be distinguished into three types. Copyleft is based on copyright, which preserves the right of any user to use, modify and distribute the software.
- *Strong copyleft*: This implies that if the software is modified completely, the software is subject to the same license as the original copyleft software.
  - *Weak copyleft*: It allows the composition of the copyleft software components with any other software or modified software; only the original components keep the original copyleft license.
  - *Without copyleft*: After the software is modified license does not depend on the original software. It can be any type, depending upon the person or enterprise that modified it.
- b) Proprietary software license: This license type is for commercial use where the licensor grants the use of one or multiple copies of software under the End-User License Agreement (EULA). Based on the proprietary software license, commercial business models can be constructed. Commercial business models are:
- Flat: It is a perpetual method. Where users purchase software at a fixed price, prices depend upon the features of the license.
  - Pay As You Grow (**PAYG**): For this type of business model, single or multiple metrics are used to measure the growth of the usage of the software. As the services grow purchase of software grows progressively.
  - Subscription: It is right to use the software for a specific time, maybe a year or multiple. At the end of subscription time, renewal of a license is necessary.
  - Declarative software license: For this process, the business agreement is done before using the software through negotiation. Then the licensor allows using of the software. However, to ensure that the software has been used according to the license terms and conditions offline audit will be performed. The VNF provider or third party can perform the audit. Today's practices are conducted by a third party trusted by the licensor and licensee.

## 2.4 VNF license

Software licenses are applicable in the different layers of NFV architectures example, Operation Support System (**OSS**), Business Support System (**BSS**), **MANO**, VNF, and NFVI. However,

licensing is complicated and risky in VNF because these are virtualized network functions where many services are deployed. For a proper and deep understanding, we tried to answer some questions;

### **Why licensing is important for VNF ?**

- a) To obtain the economic benefits of NFV, long or short-term planning, it is vital to monitor the usage of VNFs based on the license agreement.
- b) To meet the continuity of the service, usage of VNF licenses needs to be handled proactively.
- c) NFV- MANO uses the license for the VNF instantiation and operation.
- d) VNF license needs to be implemented in a flexible and extensible way to receive the benefits of the cost efficiency.
- e) VNF is software, and protecting the software intellectual rights license is very important.
- f) VNF providers need to know that their software is being used according to the terms agreed.
- g) Service provider needs to know that any failure in license entitlements will not lead to a service outage.
- h) Currently, there are no standardized methods for licensing VNF; these have facilities for many outside vendors to propose their model, which may or may not be profitable to service provider or clients.

Thus, we can say that the VNF license is crucial and impacts different aspects of NFV. But today, there is a vast diversity in VNF licensing. Therefore, there is a need to standardize the VNF license management.

### **Why is there a need to standardize the VNF licensing?**

- Standardization of VNF could facilitate services providers or VNF provider to software is being used according to the agreed terms.
- It will avoid customizing license management for each type of VNF and VNF provider.
- Standardization creates the possibility for simplifying the acquisition of license management, which will help scale the VNFs.
- It will reduce licensing errors, which could interrupt service continuity.
- It might help to create a competitive ecosystem for VNF service providers.
- Also, it helps to create uniformity in the VNF licensing.

License management is performed by the VNF License Management Entity (LM) with the help of NFVO. VNF LM is an entity that manages VNF license entitlement and also makes this information available for other functions to use. According to ETSI, LM is supposed to be in OS/BS (Operation System/ Business System). A high-level flow diagram between NFVO and LM is shown below:

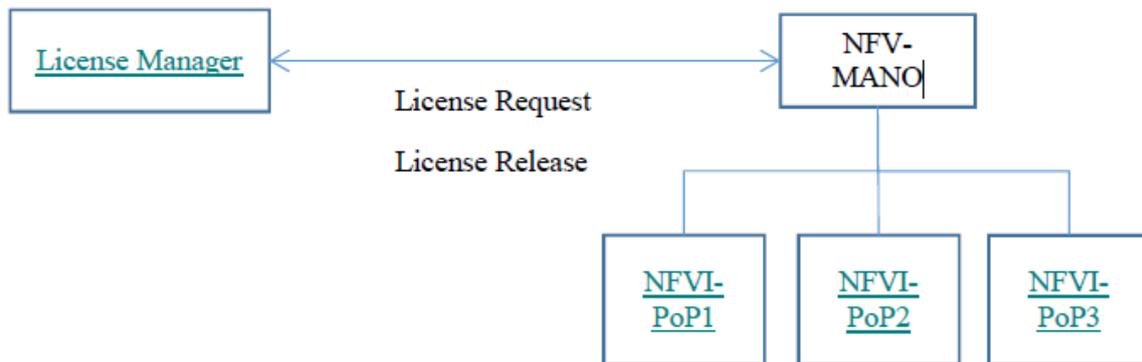


Figure 2.11: Information flow between NFV- MANO and License Manager.

Brief description of flow and figure 2.11;

- When there is a need for additional VNF instances in any NFVI-Points of Present (POPs), NFVO will request an additional license to LM.
- If there are surplus VNF instances in NFVI-POPs, NFVO will shut down the instances and release the license back to the pool.
- NFVO can move the VNF instances across the NFVI-POPs without consulting VNF LM

Thus, we see the role and importance of license management. One of the common practices for license management in IT, right now extending towards telecommunication operators, is using SAM.

## 2.5 SAM (Software Assets Management)

SAM is a tool used to assist and automated the process of software assets management [11]. SAM assists in the protection and control of software and information-related assets within an organization which is crucial for the protection and control of the software assets. The definition given by the ISO for SAM can be stated as, "All of the process and infrastructure necessary

for the effective management, control and protection of the software assets throughout their life-cycle within an organization " [11].

### **The necessity of SAM in today's time.**

SAM enables the software tracking process with the finest possible granularity. The goal of SAM is to monitor the real uses of software with usage rights acquired from the software provider and service provider to optimize and control the risk of counterfeiting (non-compliance). The foremost responsibility of the SAM is to manage the software throughout its lifecycle accurately, the moment software instances are requested through procurement, deployment, recycling, and retirement. SAM helps to lower the costs and handle the risks. It reduces the cost by controlling overbuying software. SAM helps to find the exact quantity of software needed for an organization that prevents the risk of non-compliance. SAM also helps to find the under-buying software in an organization; when the organization uses more software than contractual clauses, then the organization has to deal with counterfeiting. SAM tries to eliminate the reliance on software editors for software license usage by providing the organization with an account of their software usage and licenses to minimize counterfeiting and waste on software licenses. Thus, SAM can anticipate the requirement and non-requirements of software within an organization. Due to all these beneficial features, SAM is not an option; it is a requirement of today's technologies. Due to the rapid evolution of the infrastructure paradigm. The need for SAM is much more than ever now. Moving towards cloudification and softwarization is challenging for the SAM due to a lack of metrics, usage acquisition, and standardization. With SDN/NFV and 5G, most network functions become software. This means massive use of software in various activities to provide Telco services. Moreover, Telco services are evolving towards on-demand services. This evolving context introduces new challenges for the SAM processes of telecom operators. This paradigm shift brings breaks, changes, and challenges to SAM, as shown in Figure 2.12, which can directly impact the operator's value chains and economic performance. Some of the challenges due to cloudification and softwarization are;

- Network equipment is replaced by SW appliance deployed on the infra-cloud
- And so the notion of equipment ownership becomes a right to use a software
- The life cycle of this software is no longer related to the life of equipment but to that of virtual services in a digital economy and is, therefore, likely to be shorter (which means closer SW releases to be deployed by the operator).
- Since network SW is used directly in the value chain, it becomes crucial that the license and metrics be well adapted to the operator's business models and services.

- But the most crucial point is undoubtedly related to operations: It is becoming increasingly evident today that business process automation is one of the levers that will make this NFV transformation profitable. It is, therefore, essential to automate the license management, which will help manage the increased associated risks.

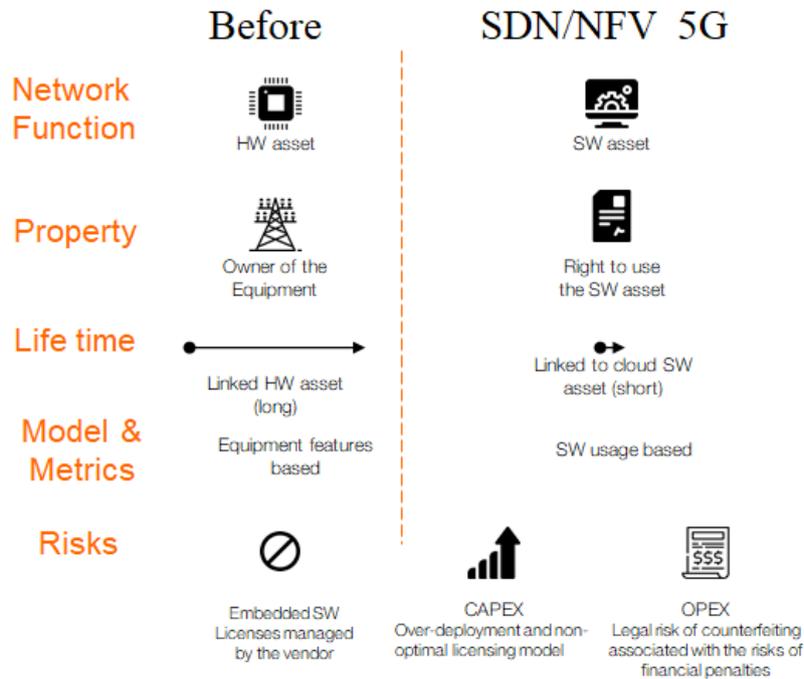


Figure 2.12: Impacts in SAM due Cloudification and Softwarization.

## 2.6 Conclusion

We can see that the emergence of software technologies has a significant impact on the licensing of software. For the efficient management of the software, enterprises use the SAM. Effective implementation of the SAM can save the organization from risks and loss. However, due to the novel arrivals of the technologies such as NFV, SDN, and 5G. Effective and reliable implementation of licensing in software is becoming a challenge. An efficient method is required for the accurate measurement of the usage of software so that it can be used to construct the license analysis. Throughout this research, we tried to show how LC is not just unit cost, how much it is essential and how it is related to other aspects of VNF. This chapter presents a literature review of the evolution of software, technologies, and their benefits and current challenges. Also, it shows the importance of licensing in software.

Chapter 3 present our proposal, contribution for methods to licensing VNF, and way to minimize the software costs.

# Modeling of Software Cost For VNF

*"You cannot be lonely if you like the person you're alone with."*

— Wayne W. Dyer.

## Contents

---

3.1	Introduction	27
3.2	Literature Review: Software cost and Existing State	28
3.3	Cost Modeling: RC, LC using Use-cases	32
3.4	Conclusion	37

---

## 3.1 Introduction

Network Functions Virtualization (NFV) is one of the promising technologies in 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/down and reduce the resources. VNF (Virtual Network Function), on the other hand, is the virtual component of NFV whose functionalities, metrics, rights, etc. are not yet thoroughly explored, which in turn services provisioning and licensing complexes, error, and lethargic. We call this problem Virtualize Network Function Software Cost (**VNF-SC**).

Licensing is a complex task. Currently, there is a vast, diverse licensing mechanism in software industries that ultimately impact the license management, renewal of license contracts, and monitoring usages. This could encourage VNF providers to come up with their licensing

models, which could create confusion and insecurity for clients [10]. It may lead to clients for non-compliance and risk of loss of a massive amount of money. Service providers want NFV to provide services on NFV, but they want to get license arrangements as they want. At the present moment, this desire of the client cannot be fulfilled due to the lack of a license model. A new model needs to be implemented without being complicated so that it will meet new services and demands.

Software cost includes more comprehensive costs, as mentioned in Chapter 2. But for this study, we had license and resource costs. **LC** is the cost to use the software under agreed rights and terms. **RC** is the cost of vCPU and vRAM to operate the VNF.

In this study, we have explored the existing problems and complications in VNF licensing and cost. Afterward, we propose a model to estimate the license requirement for the client network system. Using the LC and RC, we formed the software cost. After that, we show a use case of how a client can minimize their software cost. This study also discusses the challenges that need to be addressed in the upcoming days. In the SAM field, usually, software cost and software license costs are used interchangeably, so in this chapter, we also follow the same trends.

Until this research was written, as far as our knowledge, there hasn't been any comprehensive research regarding VNF licensing costs and models. Most of the research published in the past were focused on the problem in NFV traffic, networking, scheduling, and the overall cost, not on licensing characteristics.

This chapter is organized as follows: Section 3.2 presents a state of the art of the total cost of VNF, including software cost, total cost not including software cost, and others. Next, section 3.3 describes our proposed use case for minimizing software cost, and Section 3.4 concludes the chapter.

## 3.2 Literature Review: Software cost and Existing State

Software cost is the cost to produce the software from the initial phase to till deployment of the software. Thus, software costs include varieties of costs from HR, resources (virtual or physical), and maintenance. In our work, we consider LC and RC as software costs. LC help to protect the right of software to use, and RC charges the user as per the resource being used. In the beginning, we tried to analyze the existing research to understand the current situation of software cost and its trends.

This section provides a comprehensive literature review of software costs. In this research, the authors categorize 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.)*. This section shows how they relate to VNF license cost and missed covering

software cost, metrics, and licensing.

- **Total cost including software cost**

There are lots of papers like [4, 11, 12] that talk about the total cost of VNF is for deployment and buying costs of VNF, but very little research has been conducted, including license costs. In the paper [4] authors present a cost-efficient model for VNF chaining placement. They forwarded some interesting ideas like centrality computer module; fitness computes module, etc. to find the best and optimal placement of VNF in a way that overall cost is minimized. This research focuses on overall costs concerning network metrics, not software costs. In [12] authors proposed SET (Simple and Effective Technique). This is the extension of [4] 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, shows that orders of VNFs would help to minimize the total cost. Again, here analysis was not based on software licensing features but more focused on reducing overall cost. Article [11] presented how the SAM needs to change to adapt to softwarization.

In article [13], 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 did not talk about software costs. It was focused on VNF placement and chains. In article [14] authors presented a solution 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. To estimate path cost, they find the least path cost satisfying the QoS constraint according to the agreed service chain SLA (Service Level Agreement); however, it fails to consider the software cost. In article [15] placement of virtual Deep Packet Inspection (vDPI) was presented and solved using a centrality-based greedy algorithm. The algorithm considered fitness value composed of network resources cost and license cost of vDPI. Well, this manuscript analyzed ILP and heuristic base, which were made to optimize the cost of vDPI and placement. Although it considered a license cost, it did not fully explore the software cost and metrics.

Thus, these papers presented above considered the few cost component, but they never explore optimizing software cost and licensing of the VNF.

- **Total cost without including software cost**

For calculating proper CAPEX and OPEX software costs cannot be ignored. In article [16] was the most comprehensive paper on 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) and penalty for SLO (Service Level Objectives) violation. Their second aim was to minimize resources, active servers, 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 [17] 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 to solve the 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 article [18] 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). Using these two strategies, they measure the load distribution performance on overall servers. Thus, all these articles were conducted without considering software cost.

- **Others**

Much research was focused on other aspects, such as resource allocation of CPU and proper handling of network traffic; some were based on trust contracts, service chaining, etc. In a paper [19], they focused on efficient NFV deployment in data center networks. They try to slow down the growth of east-west traffic and minimize the data center resources. They used solving bin packing problems in each node, classifying outputs bins into good and bad. This process generated a tree structure that helps locate VNF to control traffic growth. So basically, their algorithm was based on flow assignment.

The other research was concerned with smart licensing as in [20] it presented a way to decentralize and provides security on pay-as-you-grow models in an automated and flexible way of using blockchains. Their system model was based on CSP and Network Software Vendor (NSV) relationship. So this paper talked more about software management based on its availability and flexibility using the Ethereum blockchain. In article [21], for proper deployment configuration machine learning model was used. Affinity is essential for maximizing performance and optimizing the number of allocation resources. Article [21] incorporates resources affinity for the automation deployment. This paper analyzed the VNF flavor configuration between Open vSwitch (OvS) and Single Root

Input-Output Virtualization (SR-IOV) infrastructure. This helps to construct architecture for selecting the best VNF flavor showing that VNFC were not memory or CPU bound. With the help of Figure (3.1), we classified the VNF research.

Often software costs are carried out offline (static). In the offline process, all client requirements are first analyzed depending on constraints, scenarios, and decisions. In this work costs, we are presenting are offline.

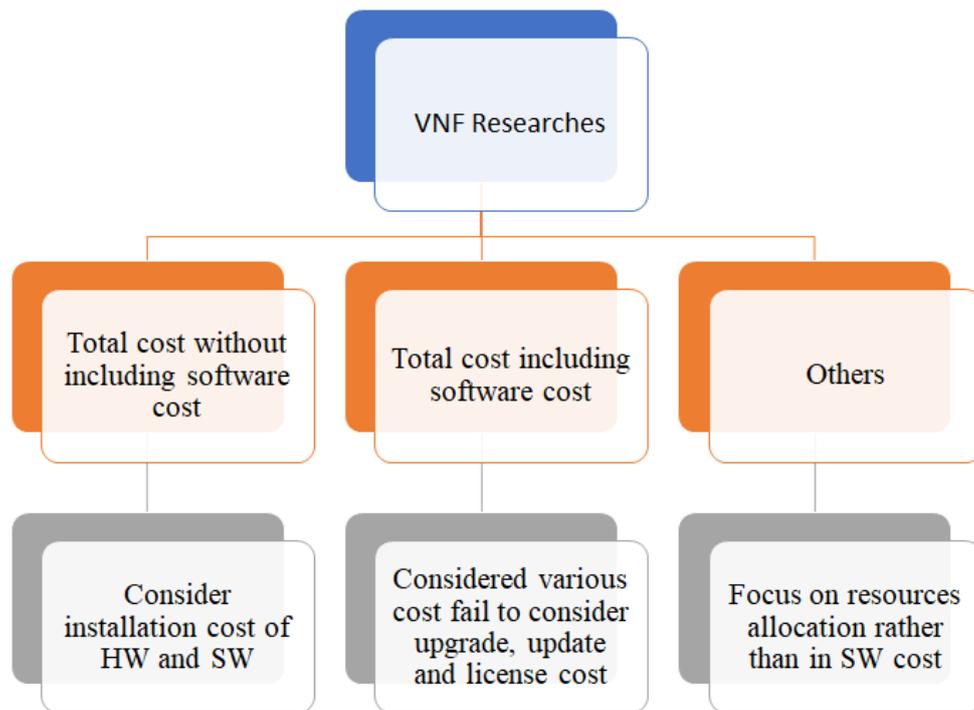


Figure 3.1: Taxonomy of research in VNF

**So, for these problems, what can be the probable solution?**

Open Sourcing is the most pragmatic approach to developing a common network software license management, a solution that must address the different concerns of ecosystem stakeholders.

A function of software license management represents different issues depending on the actors considered:

- For licensors, the issue is basically to ensure the revenue from the sale of licenses by controlling the use of licensed software
- For the licensee, the issue is to ensure compliance with the license’s terms but also to optimize the purchase and exploitation of these rights of use of software while keeping the mastering on the use of their SW assets

Both actors can, however, agree on common findings:

- Compliance with the license contract must be ensured
- A confidence in the process/tooling is mandatory
- No intrinsic market value in license management itself

Development by an open source community is the most pragmatic way to achieve the standardization goals of the network software license management solution because this co-construction will involve the different actors of the ecosystem, thus ensuring interoperability while allowing to share of the costs.

### 3.3 Cost Modeling: RC, LC using Use-cases

To address the software cost and licensing problem. We named these existing problems of software and licensed the VNF-SC problem. We present different use cases scenario that is unique and crucial. We develop models for selecting a combination of VNF, with one of the essential metrics, is SAU. Nevertheless, SAU is not the only metric used for analyzing VNF licenses. Depending upon infrastructure, there are lots of other metrics. For example, core telco uses active and attached users, transmission per second, since SAU is a widely accepted and commonly used metric for VNF licensing. In this section, we focus our analysis on SAU. Importance point here is not to be confused that SAU can behave as metrics, and usage, too; in our case, SAU is metrics. Figure 3.2 show the example of SAU on the vEPC for Ericsson. This figure shows how the vEPC is supported by the VNF and the measurement point for Ericsson. It also represents that using SAU license reference can be estimated.

Domain	Vendor	VNF	Formula	Conceptual Metrics	Measurement points (Ericsson)
vEPC	Ericsson	vSGSN	SAU - Simultaneous Attached Users. Formula based on theoretical metrics	attached users	MM.NbrActAttachedSub.G (MM/B2B), MM.NbrActAttachedSub.U (MM/B2B), VS.M2M.NbrAttachedEcUe.G (M2M/IOT)
		MME		attached users	VS.MM.NbrActAttachedSub.E (MM/B2B), VS.MM.NbrActAttachedSubCatM1.E (M2M/IOT), VS.IoT.MM.NbrActAttachedSub.NB (M2M/IOT)
		vSGW		attached users	sgwNbrOfUes, nbrOfNblctUes
		vPGW		attached users	pgwApnActiveEpsBearer, pgwApnActiveDedicatedEpsBearer, ggsnApnActivePdpContextCount
		vPDG		single users registered with at least one PDN connection	ePDG Active UEs Connected

Figure 3.2: SAU Example on the vEPC

- Figure 3.2 gives the example of the Simultaneous Active Users (SAU) metric specified for the vEPC of Ericsson.

- The calculation formula of the SAU considers the daily maximum of the hourly average number of SAU as in Equation (3.1). Which is globalized on all deployed instances.
- The right column specifies the meters which must be used for the Ericsson vEPC solution

$$SAU = \max(\sum_{All\ VNF} \text{Hourly Average SAU}) \quad (3.1)$$

Here we use SAU as metrics which is used to estimate the reference for a license, max is maximum over the day, and all VNF is all concerned VNF instances in the network. Now let us see different ways to minimize the software cost. Since we cannot present real value (cost) in each table, the vendor provided to orange due to confidentiality. We considered arbitrary values close to it in each use case. The formula is based on the maximum average value of each VNF; this formula benefits both VNF providers and VNF users. The main logic of the formula is creating a profitable situation for providers and users. Now let see the use cases

### 3.3.1 Different VNF providers Use Case (Case I)

If an organization can choose to use different VNFs from different companies, it will lower the prices, and the organization will reduce software costs. Our foundation idea is shown in Figure 3.3. Different shapes of VNFs are used to differentiate different VNFs.

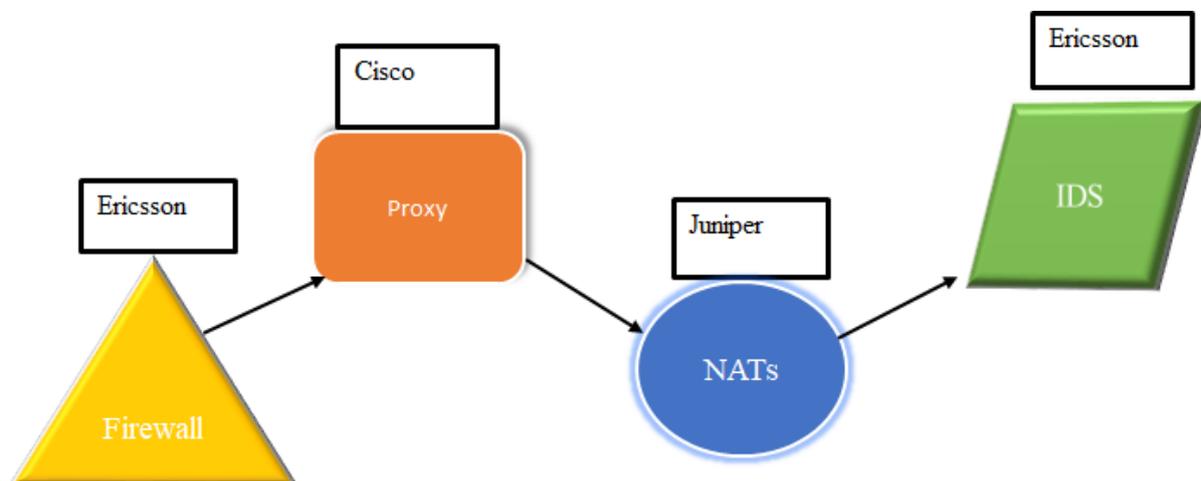


Figure 3.3: Implementation of different VNF using different VNF provider Case I .

Figure 3.3 show the different VNF in chain. Figure 3.3 can be simplified in Table 3.1. We can see that there are many choices to form the service chain that will provide similar quality services. If the client requires to minimize the cost of each VNF, then we can see that users can choose a cheap vendor, Cisco, for a firewall that is (50 €), for IDS, the user can choose

Table 3.1: 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)

Ericsson (45 €), and for proxy, the user can select Ericsson or Juniper (50 €). Thus, selecting different vendor organizations can optimize the software price.

This process sounds theoretically possible only, but an organization like Cisco is slowly adapting this kind of model. Figure 3.4 show the NFV implementation approach by Cisco where VNF from the different vendors has been used. Different colors indicates different 3<sup>rd</sup> parties [22]. It is not only limited to VNF. It goes beyond on computational domain too.

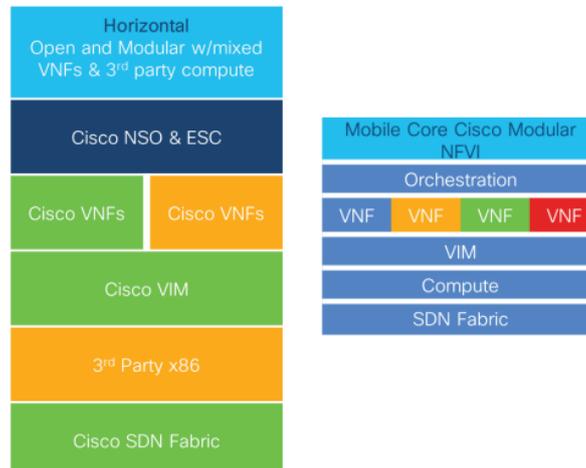


Figure 3.4: Different types of VNF and compute using third party in Cisco

Our case II is a use case based on flavor and a different range of costs.

### 3.3.2 Flavor based Use Case (Case II)

The Flavor of VNF generally means categorizing VNF depending upon its infrastructure capabilities. The flavor is based on the VNFC (vCPU and vRAM). Flavour can be classified as small (DF1), medium (DF2), and large (DF3), as shown in Table (3.2). By using the deployment flavor, users can optimize the software cost. It can be shown in Figure 3.5.

If clients/users have lots of flavors available, then users can form the flavor's combination and check whether or not this combination can meet the client's threshold QoS. Figure 3.5 is further analysis in Table 3.3.

From Table 3.3 if the client need 2vCPU and 2GB RAM , then the organization can select D2-Firewall, DF-2 Proxy, and DF-2 IDS. Clients who need 4vCPU and 4GB RAM can choose

Table 3.2: Flavor classification.

Size	Flavour	vCPU and vRAM
Small	DF-1	1vCPU, 1GB vRAM
Medium	DF-2	2vCPU, 2GB vRAM
Large	DF-3	4vCPU, 4GB vRAM

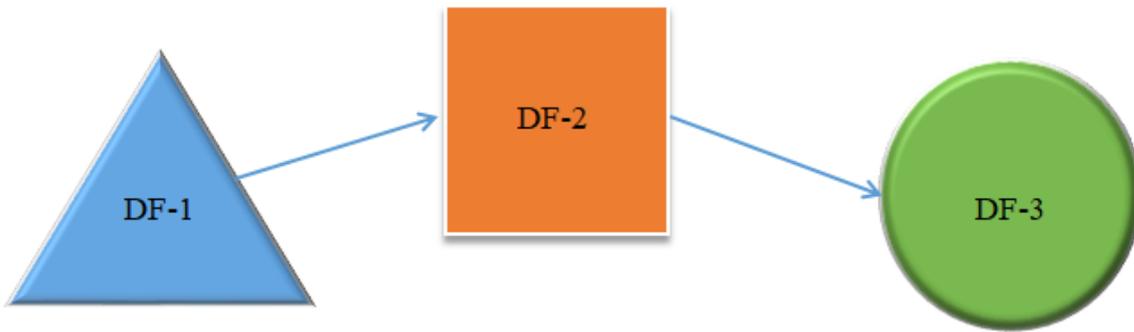


Figure 3.5: VNF implementation using different flavor.

Table 3.3: Relation of flavor with SAU.

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

DF-3 for firewall, proxy, and IDS. Thus, we can see that clients can minimize software costs by choosing different flavors depending on their QoS requirements.

We conducted experiments to analyze the relation between SAU and deployment flavor and performed simulations, which is our case III.

### 3.3.3 Flavor and SAU combination Use Case (Case III)

This case is for the VNFs service chain. From Table 3.4, depending upon the client's needs of SAU and the system's requirements, the client can select any flavor. For this experiment, we suppose that the user's 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. For this, we build the python based simulator. Our simulation selected 100 for the firewall, which is DF-1; the simulator chose the lowest one from the range if available if not, it suggests the highest. 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.

Table 3.4: 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

We presented this problem in different ways too. We can also consider the highest or lowest combination of the sum of SAU for different flavors. For example, we can make a combination of these functions such as (100, 500, 900), (100, 600, 1000), (100, 700, 1100), and so on. We are calculating the combination of each function or flavor with each other. We formed the combination following the specific order from the firewall, IDS, and proxy during the combination. After the combination simulator selects the sum of each combination, if users need the highest combination of SAU simulator will choose the highest combination of SAU. If users need the lowest combination of SAU simulator selects a lower combination of SAU; also, if users request the mean, it will serve a combination of the mean SAU. So using this simulator, the client can choose the flavor depending upon the SAU, either the lowest or highest depending upon the needs.

These operations can also be further diagnosed in different ways, which is our case IV.

### 3.3.4 SAU, Flavor and Cost combination Use Case (Case IV)

We can make a combination of flavor, SAU, and cost for each VNF; well, there is a lot of software cost (maintenance cost, installation cost), but over here in Table 3.5 we take arbitrary value for license cost only. However, it will not change if we consider the other costs.

Table 3.5: List of cost with SAU and flavour.

Flavour	SAU for Firewall	Cost	SAU for IDS	Cost	SAU for Proxy	Cost	Infrastructure Cost
DF-1	100	110.56 €	100	120.63 €	100	100.56 €	60 €
DF-2	500	110.56 €	500	120.63 €	500	100.56 €	80 €
DF-3	1000	60.16 €	1000	50.12 €	1000	40.56 €	100 €

We can analyze flavor according to SAU or cost. We build a simulator. That helps to select according to price or SAU as per the need of the users. Users have to choose the highest or lowest cost if the cost selected.

For example, let us say the user selects the lowest cost. It will be 60.16 € for the firewall. It should be noted that if the user selects the lowest cost, it will have the highest SAU; in this case, it is 1000. It is because we follow the current trends in the telecoms industries. Similarly, users can use SAU for selecting suitable flavors; in these examples, we take the highest SAU, i.e., 1000, and similarly for other VNFs too, which will help to choose the lowest cost.

The most crucial 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. This is because license costs remain unchanged after a certain number of licenses (license threshold), but infrastructure costs keep growing; this is the current trend in industries. Moreover, it's business policies to lucid the customer. Since SAU is higher, so it obvious that more vCPU, and a larger size of vRAM is required.

Thus, clients and users must have their threshold cost and SAU before choosing the flavor or service providers, and vice versa, which helps to optimize their software cost.

### **3.4 Conclusion**

The implementation of NFV is rapidly increasing alongside 5G. Therefore, NFV is undoubtedly the future of networking architecture. Thus, standardized tools for licensing VNF are the absolute need of the hour. In our study, we present a module for licensing and use it to construct software costs. First, we gave the current state of the software cost and license, and then we showed how they are not enough for VNF. After that, 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 helps to reduce the software cost. Similarly, a combination of SAU and deployment flavor significantly reduces the software cost of the client acquiring the threshold QoS.

Chapter 4 will see how this software cost can be used to form the software total cost and minimization of the total cost based on various cases and scenarios.

# Total Cost Modeling For VNF Based On Different Use Cases And Scenario

*"You are an aperture through which the universe is looking at and exploring itself."*

— Alan W. Watts.

## Contents

---

4.1	Introduction	38
4.2	Related Works	40
4.3	Proposal Model	41
4.4	Use cases with different scenarios	44
4.5	Conclusion	59

---

## 4.1 Introduction

Due to the fast technological evolution, there are increasing new service user cases (IoT, eHealth, Smart cities, AR/VR (Augment Reality, Virtual Reality)). Each use case imposes different network requirements related to speed, latency, and isolation. To deliver this demand at a large scale, the underlying network needs to be software driven and automated. This is the reason why we need NFV. Moving to NFV (Network Function Virtualization) and SDN (Software Defined Network), Telco cloud architectures face four key challenges: interoperability, automation, reliability, and adaptability. All these challenges involve optimizing resources, whether

increasing the utilization of hardware resources (virtualization) or delivering shared computing resources and functions in real-time (cloudification). The Softwarization of networks is a consequence of telecom cloudification. Virtual Network Function (VNF) is protected by IPR (Intellectual Property Right) like any software, ensured by a license describing usage rights and restrictions at a given cost. Until now, limited studies have happened on the economic dimension linked to softwarisation. Currently, the telco industry struggles to converge and standardize licensing and cost models. At risk: the network cloudification benefits could be swept away by poor management of resources (Hardware and Software). This section presents a preliminary model for optimizing the total cost of a VNF based on the Resource Cost (RC) and License Cost (LC). This analysis is inspired by measurement and licensing practices commonly observed in the Telcos industries, i.e., consumption and capacity.

NFV aims to increase automation and network reliability for better and quicker service delivery. From the research and markets, the global NFV market is projected to grow from 12.9bn dollars in 2019 to 36.3bn dollars by 2024<sup>1</sup>. Also, it is predicted that between 2022- 2032 NFV market growth rate will be 6.6%<sup>2</sup>. Service providers, specifically telco companies, must adapt quickly and efficiently to this shift. More than only technological challenges, historical telcos must face the arrival of large hyper-scalers, partners, and aggressive competitors. Service Providers will benefit from NFV if they can enable new services with a faster time-to-market and rapidly scale resources up and down, lowering the costs. The key challenges facing NFV are thus linked with resource optimization. Also, success relies on the ability to monitor and use standards and interoperable resources: in other words, to mix and match various software components on standard COTS hardware.

As the network becomes software, failure in controlling software spending destroys the promises of NFV efficiency. In addition, the paradigm shift from equipment property toward Software (SW) Right To Use (SW RTU) adds complexity to resource management. As IPR protects SW over a license, it becomes essential to ensure the compliance of SW deployments regarding acquired rights. As well, it becomes necessary to optimize license costs. For this, one of the standard practices in IT, extending to networks, is to practice SAM. Implementing end-to-end SAM guarantees that users buy all the licenses user need; only the license needs: to avoid counterfeiting and waste [11, 23].

SW license frames the rights and obligations of the CSP to use SW. License is associated with a cost (LC) which depends on the volume of rights granted. The volume given and associated conditions of use are contractually defined by one or several metrics. There are currently no standards on metrics and their definition; they depend on the creativity of software providers. This could facilitate every supplier of VNF to propose their metrics, model, and tools. More-

---

<sup>1</sup><https://www.f5.com/fff/company/blog/why-nfv-ismore-relevant-than-ever>

<sup>2</sup><https://www.factmr.com/report/1327/nfv-market>

over, this is an intriguing and complex task for VNF service providers and users. Based on our observation of the telcos industries, metrics can be linked with usage, resource allocation, and resource consumption. Thus, it is not wiser to consider LC and RC independently, but we need to consider that LC can depend on resources and vice versa.

One of the significant hurdles that exist till now is that end-users have to give access to their valuable data to vendors or service providers to help optimize the licensing and total cost. This creates doubts and fear among clients. As a result, there is a lack of sovereignty, motivation, and confidence in clients and a lack of appropriate options for optimizing the total cost for clients' systems.

Thus, to address these and other related issues associate with VNF licensing. We propose a model to evaluate and optimize the Total cost of VNF based on LC and RC. We base our model on two metrics, SAU and BW, as we observed that they are well known in the telcos industries and can fit both resource consumption, allocation. Total Cost (TC) is calculated based on LC and RC. Although resources cost occasionally include LC and other costs such as link, maintenance, upgrade, and hardware costs.

In this study, we considered only RC as VNF instances required to operate the necessary amount of license (license reference). License is an agreement that comes with rights and duties. The right is to use a certain amount of SAU or BW in the respective VNF, and the responsibilities are to comply.

## 4.2 Related Works

In the past, some research was focused on optimizing the network cost and the network path most of them included LC as a constant entity. In article [24], they have dealt with finding the best place of VNF for a better routing path for each demand. This article helps our research for finding the placement for SFC (Service Function Chain). Article [4] had a significant impact on licensing, but the authors did not focus on the licensing cost or providing better options for clients. In [25] they developed the VNFPPRA problem, which finds the optimal placement of VNFs in SDN/NFV-enabled MEC (Mobile Edge Computing) nodes to reduce the deployment and resource cost using genetic and mixed-integer problems. From this article, we get an idea about resource allocation. In [26] they focused on the placement of the VNF and traffic steering using network cost, node cost, and VNF placement cost. However, this article did not discuss the total cost and LC estimation and optimization but has illustrated an excellent insight into the in-network cost.

In [27] formulated problems for joint optimization and traffic cost optimization using the Markov Approximation (MA), in which they added their matching approach called SAMA. This research helps us to get a good idea of VNF instances. Although this work was vast and

concrete, they have not considered license cost or overall total cost. In [28], authors focused on resource allocation of NFV components using Markov Decision Process and Bayesian learning, which helped to allocate NFV components dynamically. In [29] solutions are based on a flexible resource allocation focused on analyzing delay rather than optimizing cost. They have dealt with the placement of chain VNF to minimize the delay between ends to end VNF using the Mixed Integer Quadratically Constraint Program (MIQCP).

Article [30] was based on QoS of non-cooperative network domain using deep learning to reduce the bidding price, network cost, network service chain optimization, and Forward Graph (FG). This study is not focused on the license and its associated cost. Article [31] performed research about resource allocation and VNF placement, where they used service delay as a critical point performance. This research is not based on the cost analysis of NFV; nevertheless, it gave us good insight into the delay and VNF placement. Research paper [32] provides a survey article that talked about resource allocation in VNF. Categories in three-stage VNF-CC (Chain Composition), VNF-FGE (Forward Graph Embedding), and VNF-SCH (Scheduling). Thus, this research focus on various aspects, excluding the economic dimension of the NFV.

All these articles mentioned above have no license cost or optimizing models for the software cost. Nevertheless, these costs, LC, RC are significant for companies [4, 11, 23].

## 4.3 Proposal Model

### 4.3.1 Traditional ways

There are many different pricing and cost models for the software still used in the Cloud, NFV, and SDN. However, these models are not enough for current evolving technologies, which increase different new use cases. The famous traditional cost models are a perpetual license, pay-as-you-grow, and subscription. In this section, we use only two methods to compare with our model;

- **Perpetual license:** In this system, the user has to pay upfront; then, the user has the right to use the software. Depending upon the license entitlement, user can upgrade and updates their software. Since it is a one-time payment, generally, it is costly. The dimension parameters used for a perpetual license are:
  - $LC_{prl}$ : One-time cost, upfront payment for the license.
  - $RC_{prl}$ : One-time cost, upfront payment for resources.
  - $TC_{prl}$  : Total cost for perpetual license.

$$TC_{prl} = LC_{prl} + RC_{prl} \quad (4.1)$$

- **PAYG:** In this model, end users must pay according to their consumption capacity increment. It can be used based on resources, services, or others. There are many pay-as-you methods, such as pay-as-you-use, pay-as-you-eat, and pay-as-you-go, but we use it as pay-as-you-grow and grow here in terms of SAU/BW. The dimension parameters used for the PAYG license model are:

- $nb_{SAU}$ : Number of SAU at a time.
- $CS_{PAYG}$ : Unit cost per SAU license.
- $CSr_{PAYG}$ : Unit cost for Resource.
- $LC_{sPAYG}$ : License cost for SAU.
- $RC_{sPAYG}$ : Resource cost for SAU.
- $TC_{sPAYG}$ : Total cost for PAYG.

$$LC_{sPAYG} = CS_{PAYG} \times nb_{SAU} \quad (4.2)$$

$$RC_{sPAYG} = CSr_{PAYG} \times nb_{SAU} \quad (4.3)$$

$$TC_{sPAYG} = LC_{sPAYG} + RC_{sPAYG} \quad (4.4)$$

Now for the BW, dimension parameters are;

- $nb_{BW}$ : number of SAU at a time.
- $CB_{PAYG}$ : Unit cost per SAU license.
- $CBr_{PAYG}$ : Unit cost for Resource.
- $LCB_{PAYG}$ : License cost for SAU.
- $RCB_{PAYG}$ : Resource cost for SAU.
- $TCB_{PAYG}$ : Total cost for PAYG.

$$LCB_{PAYG} = CB_{PAYG} \times nb_{SAU} \quad (4.5)$$

$$RCB_{PAYG} = CBr_{PAYG} \times nb_{SAU} \quad (4.6)$$

$$TCB_{PAYG} = LCB_{PAYG} + RCB_{PAYG} \quad (4.7)$$

- **Subscription:** In this model, licensing is done for a certain period (monthly or yearly) of subscription time based on the number of SAU or BW. For example, 1000 SAU within six months or 2 Gbps per month.

One of the crucial points here not to forget is that these are business models, not actual licensing models. They are only used as license models due to the lack of the standard license model. To fill this gap, we conducted this research and proposed license models that ultimately help construct business models in a virtual environment, VNF.

### 4.3.2 Cost model for Virtual Network Function

Our crucial challenge is to construct reliable and authentic methods for estimating license and its associated costs for the VNF, which would ultimately help to optimize the total cost of VNF. We used two important license metrics to address this task: SAU and BW.

- **SAU:** SAU means simultaneous active users connected with VNF who are consuming some resources and using services provided by VNF.
- **BW:** It is related to the amount of bandwidth-consuming/consumed by SAU.
- **License Reference (LR):** It helps estimate the number of licenses required for the VNF system for a certain period and duration.

Since these parameters define the usage of VNF, these parameters fulfilled the preliminary requirement to be license metrics and are also regarded as simple, scalable, and measurable parameters. Using these two metrics, we created two license references as;  $LR_{SAU}$  Equation (4.8) and  $LR_{BW}$  Equation (4.9) respectively for SAU and bandwidth, they can be formulated as follows;

$$LR_{SAU} = \max \left( \sum_{\forall i \in V} average(SAU_i) \right) \quad (4.8)$$

Similarly for BW,

$$LR_{BW} = \max \left( \sum_{\forall i \in V} average(BW_i) \right) \quad (4.9)$$

where,  $V = (1, 2, \dots, v)$  is a set of all concerned VNF (can be same or different type) in node. We define H as a set of hours (1, 2, 3, ..., h), D as a set of days (1, 2, 3, ..., d) and R as a set of License Reference (LR). LR corresponds to  $LR_{SAU}$  or  $LR_{BW}$  which we get from the equations (4.8) and (4.9). Our assumption for this research was that all VNFs were properly deployed in their respective places and functioning accurately at full capacity. These license references helped to estimate an optimized total cost model, which includes license and resource costs calculated as follows;

$$TC_j = RC_j + LC_j \quad (4.10)$$

Now,  $LC_j$  and  $RC_j$  can be calculated for capacity model for a day,  $j \in D$  and  $r \in R$ , as:

$$LC_j^{ca} = \phi_r + \sigma_r \alpha_r \tau_r, \quad (4.11)$$

$$RC_j^{ca} = \theta_r + \delta_r \tau_r \beta_r \quad (4.12)$$

$LC_j$  and  $RC_j$  can be calculated for consumption model for a day,  $j \in D$ , as:

$$LC_j^{cp} = \gamma_j \times \alpha_r, \quad (4.13)$$

$$RC_j^{cp} = \gamma_j \times \beta_r, \quad (4.14)$$

where,

Table 4.1: Key Notation

Notation	Description
$r$	A License Reference. $r \in R$ ,
$\phi_r$	Pre-paid amount for License Reference (€),
$\tau_r$	Surpass or exceed License Reference $r$ ,
$\alpha_r$	Unit cost of license for License Reference $r$ (€/SAU or Mbps),
$\sigma_r$	License factor for License Reference $r$ ,
$\theta$	Prepaid amount for resources (€),
$\delta_r$	Resources factor for License Reference $r$ ,
$\beta_r$	Unit cost of resources for License Reference $r$ ,
$\gamma_j$	License Reference for a day, $j \in D$ ,
$LC^{ca}$	License cost for capacity,
$RC^{ca}$	Resources cost for capacity,
$LC^{cp}$	License cost for consumption,
$RC^{cp}$	Resources cost for consumption.

Now, after formulation these formula we implement these equations (4.8), (4.9) and (4.10) in different scenario, such as *i) VNF instances scenario ii) Users dependent iii) Using flavor in nodes*. These scenarios depend on users, usages, and nodes.

## 4.4 Use cases with different scenarios

This section presented different use cases in VNFs and showed how users could optimize license and resource costs.

#### 4.4.1 Scenario 1: VNF instances

For this scenario, we presented different available clients' usages like; Web, VoIP, and Online Game. We modified the table of [27] to adjust it to our model as shown in Table 4.4 for BW and SAU. For this scenario, users need to be aware of their requirements based on SAU or BW. Using SAU and BW the flavor base table is proposed on Table 4.2 and Table 4.3 which is supposed to meet the user's requirements. A Python-based simulator was built for this scenario where the user can input their requirement. The simulator will provide the best available options from tables according to the client's needs. Importantly, license cost and resource cost mentioned in our tables are  $LC^{ca}$  and  $RC^{ca}$ , i.e., license capacity and resources capacity cost.

Table 4.2: BW Flavor Table for Scenario 1 (VNF instances)

Flavor	BW (Mbps)	LC (K€)	vStorage (TB)	vRAM	vCPU	Redundancy	Resources cost (K€)
A	15	200	2	2 GB	2	1	170
B	20	250	3	2 GB	3	2	200
C	30	355	4	3 GB	4	2	285
D	50	435	4	3 GB	4	2	338
E	65	549	5	4 GB	4	2	420
F	Customize your needs						

Table 4.3: SAU Flavor Table for Scenario 1 (VNF instances)

Flavor	SAU	LC (K€)	vStorage (TB)	vRAM	vCPU	Redundancy	Resources cost (K€)
A	100	250	4	2 GB	2	1	100
B	150	350	4	3 GB	3	2	150
C	200	450	6	4 GB	4	2	215
D	250	550	7	5 GB	5	2	275
E	350	650	8	5 GB	5	2	325
F	Customize your needs						

Now, for example, let us consider first Bandwidth (BW) is the client requirement if the traffic requirement is BW (65Mbps) for web services from Table 4.4 then for this service simulator proposed flavor E along with RC and LC from Table 4.2. If the use case is VoIP, then the simulator will propose flavor D along with other parameters from Table 4.2. Whenever exact traffic range requirement is not input by the client for such case simulator suggested higher value flavor, and for the third use case, online game since users requirement is higher than the available option. So users either can be satisfied with flavor D or the best option would be to customize their requirements with the services provider, i.e., flavor E. In such a situation sim-

Table 4.4: Service chain of different client usages with BW

Client usage	Service Chain	Minimum Traffic Required (BW)	Minimum Traffic Required (SAU)
Web Services	NAT-FW-WOC-IDPS	65Mbps	165
VoIP	NAT-FW-TM-FW-NAT	35Mbps	200
Online Game	NAT-FW-VOC-WOC-IDPS	150Mbps	300

*NAT: Network Address Translator, FW: Firewall, TM: Traffic Monitor, WOC: WAN Optimization Controller, IDPS: Intrusion Detection Prevention System, VOC: Video Optimization Controller*

ulator proposed flavor D along with a note that the client can go for customization since their given range of BW is beyond the option available in the table.

A similar phenomenon goes for SAU as well. If the use case is web services (165 SAU) from Table 4.4 usage, our simulator proposed flavor C from Table 4.3 because the exact range is not available, and it proposed a higher range flavor. Similarly, for VoIP, flavor C perfectly matched the requirement of SAU, and for the online game, flavor E, covered the user's need; in this case, the simulator proposed a slightly higher range flavor.

Thus, using these flavors tables based on SAU and BW, the end-user can estimate license and resource costs regarding these they can optimize the total cost.

#### 4.4.2 Scenario 2: Users

Our second scenario depends on the nature of the users. Nature of users means the users have exact or some knowledge regarding the resources in their system or not. In this research, we categorized users into two types, ❶ Resources Know Users and ❷ Resources Unknown Users.

❶ **Resources Known Users (RKU)**: These users have tentative knowledge about the number of resources in terms of SAU/BW required for their system. To estimate the need for resources, clients might have to use paid or unpaid software such as OpTISAM (from Orange free software)<sup>3</sup>, Cisco Asset Management (Cisco, paid)<sup>4</sup>, and Ericsson adaptive inventory (Ericsson, paid)<sup>5</sup>. OpTISAM is a free software assets management developed by Orange. For this scenario, a flavor table was created based on SAU, BW like in previous Table 4.2 and Table 4.3, also in these tables, the total cost was introduced so that depending upon the user's budget they can choose flavors too.

<sup>3</sup><https://opensource.orange.com/fr/actualites/optisam-un-outil-optimise-pour-la-gestion-dactifs-logiciels/>

<sup>4</sup><https://software.cisco.com/>

<sup>5</sup><https://www.ericsson.com/en/press-releases/2015/6/ericsson-adaptive-inventory-redefines-network-and-service-resource-management>

For this evaluation, a second simulator has been created named flavor selector, where users can input the range of SAU, BW, and total cost depending upon their requirement then our simulator will propose the approximate flavor. For example, if the SAU range given by the client is 100-120, our selector suggests flavor A from Table 4.5. Another case is a range between 130-160 then our selector suggests flavor B. Also, whenever the user inputs a higher value that is not in the available range value, it will suggest customizing flavor, i.e., flavor F or the highest flavor from the table, that is, flavor E. A similar goes for BW; for example, if the client provides a range of bandwidth between 18-20 Mbps, the selector will propose flavor B from Table 4.6 since it is in the range (20 Mbps)

Another interesting case is using total cost. For the total cost, we added the value of the Resources Cost ( $RC^{ca}$ ) and License Cost ( $LC^{ca}$ ) with the help of which we created the last column (9<sup>th</sup>) in Table 4.5 and Table 4.6. Hence, depending on the customer's budget simulator proposed a flavor between SAU and BW. For example: if the client's budget range is from 800-900k€, the simulator provides flavor D from Table 4.5. Furthermore, if the range is from 600-700k€, it could be from flavor C from the SAU or BW table. So, to avoid the confusion of choosing between SAU and BW, the simulator asks about the client's preference between SAU and BW in the beginning. When the client chooses, their requirement is the total cost. Thus, depending upon the client's needs, the simulator provides the result either from BW, SAU, or total cost.

Thus, using these two metrics and the total cost will give liberty to the client to choose between different available options depending upon their needs, usages, and situation, which helps to estimate and optimize VNF cost.

🔴 **Resources Unknown Users (RUU):** Now, for the second type of users. These users don't have the estimated knowledge about their system's resource requirements (SAU, BW). So for these types of users, a third simulator was created to provide the users with several choices. At first, users need to provide their range (between SAU, BW, and total cost), after which the simulator will propose the least SAU value from the Table 4.5 along with other associate parameters like storage, vRAM, vCPU, and cost. If the user is not satisfied with that proposal, then they can process further simulator will propose from BW Table 4.6, least range from BW.

If this range is not satisfactory to the client's requirement, then the simulator proposed the mean value from the SAU flavor table. If this also fails to meet the user's needs, the simulator suggests the mean value from the BW flavor table. After this, the simulator proposed the highest value of SAU and BW from SAU and BW flavor table, respectively. So the simulator proposed from least to maximum flavor value from tables based on SAU, BW, and total cost. Thus, our aim here is to provide the user with as many options as possible. Additionally, the offer can be made concerning the total cost as performed in RKU. Figure 4.1 summarize the overall users types.

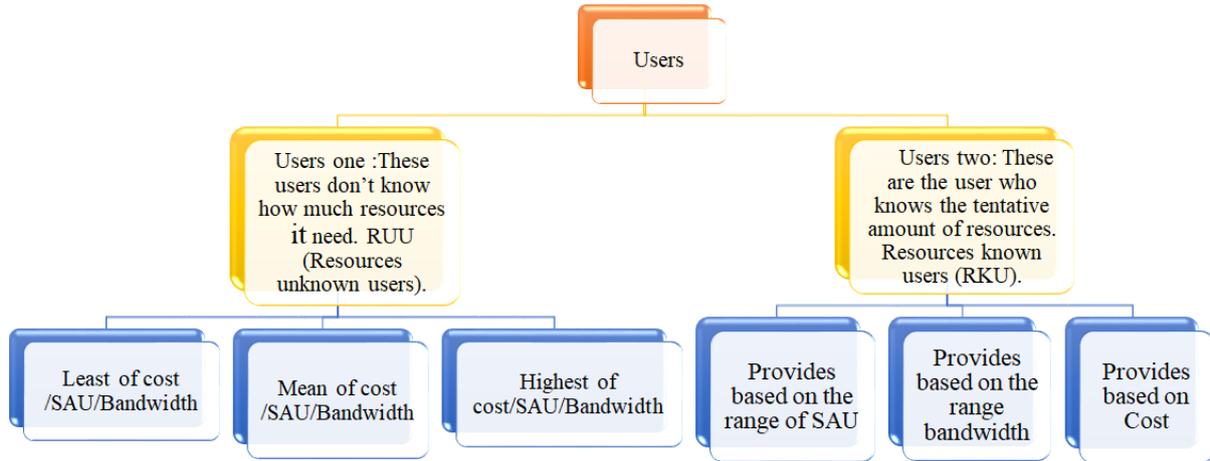


Figure 4.1: Taxonomy of Users Scenario

Table 4.5: SAU Flavor Table for Scenario 2

Flavor	SAU	LC (K€)	vStorage (TB)	vRAM	vCPU	Redundancy	Resources cost (K€)	Total cost (K€)
A	100	250	4	2 GB	2	1	100	350
B	150	350	4	3 GB	3	2	150	500
C	200	450	6	4 GB	4	2	215	665
D	250	550	7	5 GB	5	2	275	825
E	350	650	8	5 GB	5	2	325	975
F	Customize your needs							

### 4.4.3 Scenario 3: Nodes

We used the two techno-economic-friendly models for this scenario to estimate the LC and RC. They are capacity and consumption. To adapt these models from a business point of view, we have considered some thresholds and constraints related to licensing and resources like license threshold, resources threshold, and license factors, as mentioned below;

- Capacity:** Capacity means the amount of SAU or BW that can serve by the system (VNF) at a specific time. The capacity analysis is similar to prepaid service, where a certain amount of cost is paid upfront to a specific capacity (license reference for our research) of VNF. When it surpasses the threshold, extra costs will be incurred. The threshold can be a license or resource, or both. In this research, both LC and RC were estimated using unit cost and license reference using Equations (4.11) and (4.13). In this mode, once the capacity is increased, it cannot be reversed even if the consumption (SAU/BW) is lower than the threshold.

Table 4.6: BW Flavor Table for Scenario 2

Flavor	BW (Mbps)	LC (K€)	vStorage (TB)	vRAM	vCPU	Redundancy	Resources cost (K€)	Total cost (K€)
A	15	200	2	2 GB	2	1	170	370
B	20	250	3	2 GB	3	2	200	450
C	30	355	4	3 GB	4	2	285	640
D	50	435	4	3 GB	4	2	338	773
E	65	549	5	4 GB	4	2	420	969
F	Customize your needs							

- **Consumption:** It means amount of SAU or BW consumed at a time. Clients will pay for the resources consumed during a specific time. Consequently, there is no contractual threshold limiting the user's ability to consume resources (SAU, BW). It was calculated using Equations (4.13) and (4.14).
- **LC threshold:** This is the threshold for calculating license costs. LC threshold was implemented for  $LC^{ca}$ . It is defined at the contract's negotiation time based on estimated needs. If the uses exceed the threshold, the license cost is increased by 1.5, 2, 3, ..., n, known as license factor  $\sigma$ . The license threshold is based on LR.
- **RC threshold:** It is the threshold in the resources. Whenever a threshold is exceeded, it requires a careful evaluation to understand whether the exceeded threshold can be covered by a single resource or more. If it can be covered with one resource, then our research will use resources factor ( $\delta$ )=1, and if it requires more than one, it will be from 1.5, 2, 3, 4, 5, ..., n. This threshold is also dependent upon the LR. Similar to the LC threshold, it can be negotiable between the VNF provider and the SP.
- **LC factor ( $\sigma$ ):** It is a multiplicative factor after exceeding threshold,  $\sigma = 1.5, 2, \dots, n$ . The license factor and resources factor were introduced here to create a proper business model because whenever the threshold is exceeded in the capacity model, the service provider will charge some extra amount.
- **RC factor ( $\delta$ ):** It is similar to the LC factor but in the resources aspect.  $\delta = 1.5, 2, 3, \dots, n$ .

So, now using all these metrics and equations (4.14) and (4.13) the total cost in *consumption model* becomes:

$$\max_{\forall j \in D} \sum (\alpha_r \gamma_j + \beta_r \gamma_j) \quad (4.15)$$

Similarly, from equations (4.12) and (4.11) total cost for *capacity model* became:

$$\max_{\forall j \in D} \sum (\phi_r + \sigma_r \tau_r \alpha_r + \theta_r + \delta_r \tau_r \beta_r) \quad (4.16)$$

A presumption was made that it will meet the QoS threshold,  $TH_f$ , i.e  $\min TC \preceq TH_f$ .  $TH_f$  is not a numerical value but a condition.

#### 4.4.4 Node scenario

The first analysis was performed in a single node and the second with multiple nodes because we wanted to prove that our models could be adapted to almost any scenario. Also, in a single node focus was on the nature of LR, and in multiple nodes, the focus was on two metrics, SAU and BW, using two usages aspects of VNF, which are capacity and consumption.

##### 4.4.4.1 Single Node

In this case, two different License References (LR) were generated as in Figure 4.2. One of them is increasing steadily while another is increasing rapidly, and our models (Capacity and Consumption) were applied to these data. For this case simulation adopted parameters are given in Table 4.7.

Table 4.7: Simulation parameters for a single node

Parameters	Value
License threshold for capacity model	500
Resource threshold for capacity model	300
$\sigma$ for both capacity and consumption models	1.5
$\delta$ for both capacity and consumption models	1.5
$\theta$	300€
$\phi$ for capacity model	500€
$\alpha$ for LR and LR1	10€/LR
$\beta$ for LR and LR1	10€/resource

##### 4.4.4.2 Result evaluation for single node

Figure 4.2 shows the two types of License References (LR and LR1 (License Reference) that were generated in a single node. LR1 is steadily increasing, whereas LR is rapidly growing as days pass. When the research model was implemented in these two datasets, we obtained a license cost as in Figure 4.3. LR and LR1 license costs for the capacity and consumption model started and ended almost at the exact cost. The overall consumption model estimated the highest cost. Until the first 5 days, LR1 consumption cost was constant. However, after day 5 the

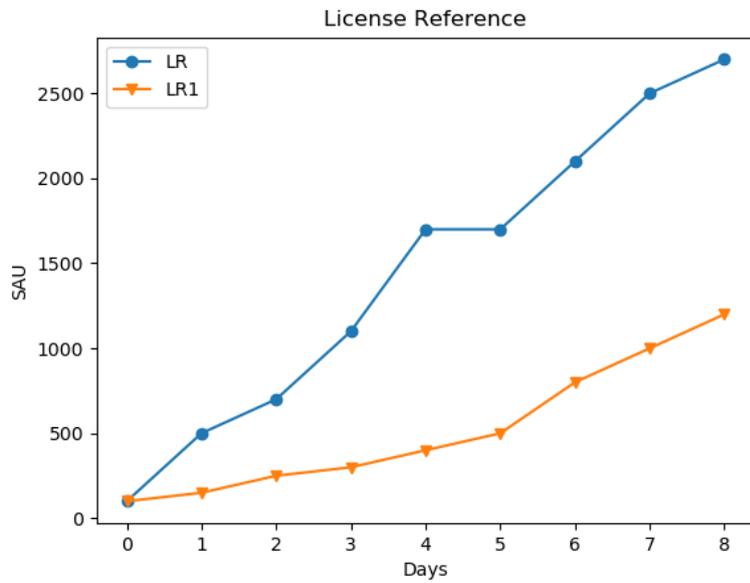


Figure 4.2: Daily License Reference (LR1, LR)

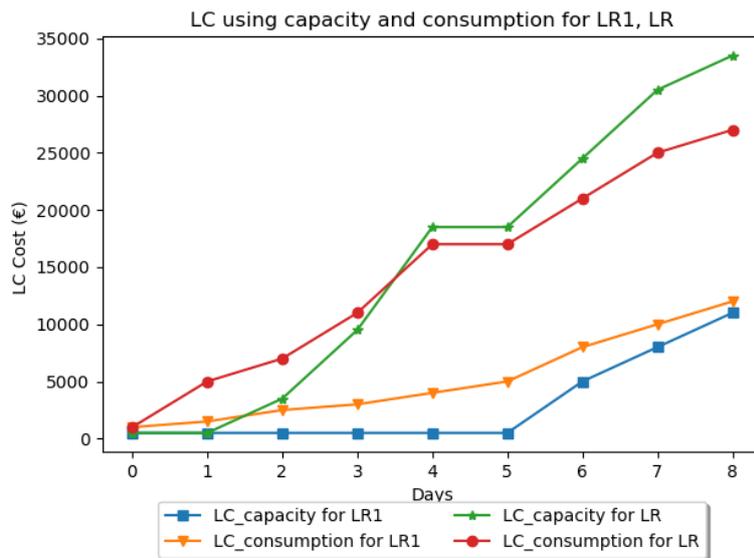


Figure 4.3: Daily license cost for both license reference (LR, LR1)

cost started to increase slowly and ended at the exact cost as the consumption. LR capacity cost started from the same cost as LR consumption, but after day 3 it overtook and surpassed the LR consumption cost. Figure 4.4 and Figure 4.5 show the resource cost and summation of LC and RC, respectively. They were similar in license cost. It can be seen for LR, the capacity model generated less cost in the beginning days, but as days passed, the capacity model cost overtook the consumption model cost. When data is slowly (LR1) increasing, the capacity model generates less cost (license, resources, summation) than consumption at the beginning but ends at a

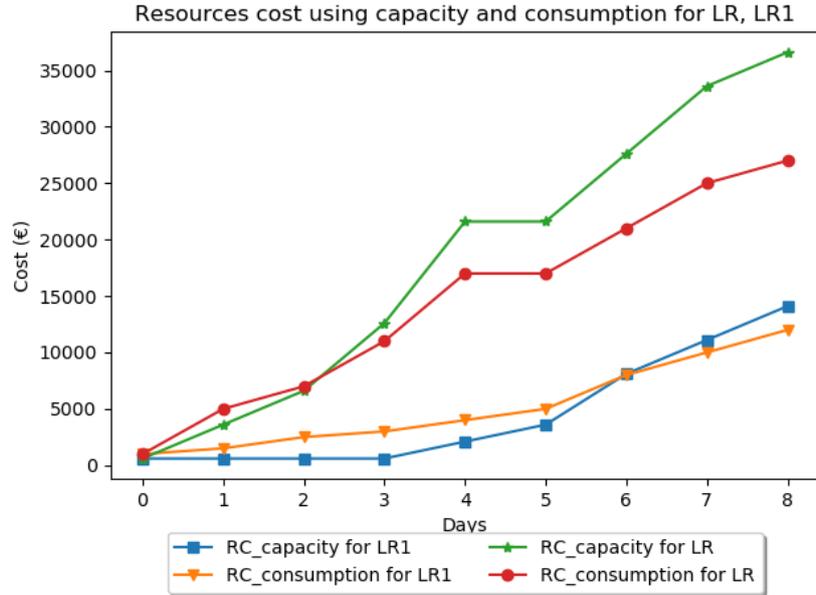


Figure 4.4: Daily resource cost for both license reference (LR, LR1)

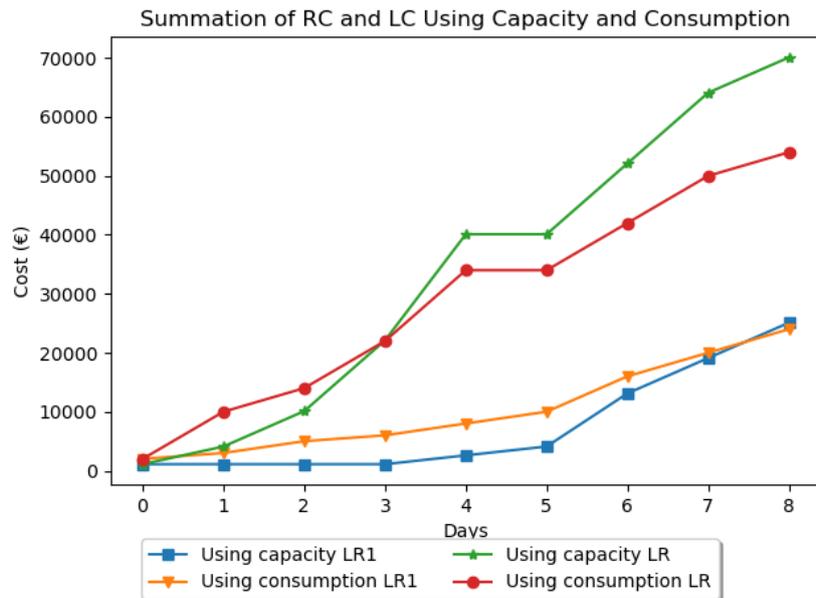


Figure 4.5: Daily summation of RC and LC (Daily Total cost) for both license reference (LR, LR1)

slightly higher cost value than consumption. Whereas, when LR is rapidly increasing (LR), the capacity model generates less cost only for a few days and quickly surpasses the consumption model cost. Cumulative total cost was calculated as shown in Figure 4.6 from which it can be interpreted that LR1 and LR consumption model estimated less cost than capacity. Thus, this process offers the choice of models, either consumption or capacity, for users. That signifi-

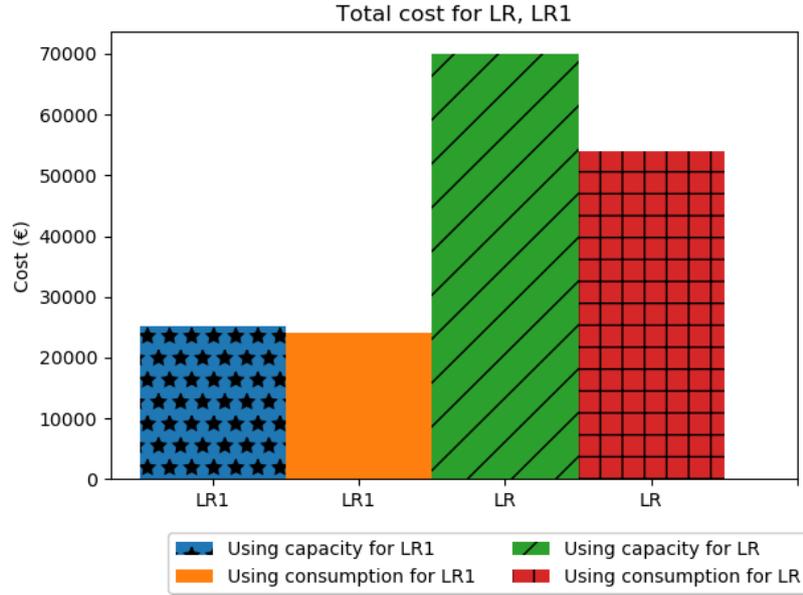


Figure 4.6: Total cost for both license reference (LR, LR1)

cantly impacted the total cost, and the choice of models depended on the nature of the license reference.

The point here is that we want to prove that depending upon the nature of license reference, different models give different outcomes. LR's nature depends upon what metrics have been used to estimate LR. It can be hourly average SAU or BW or other metrics.

#### 4.4.4.3 Multiple Nodes

We considered a similar scenario for this situation as in [33] i.e., Intersection Collision Avoidance (ICA). ICA is used in today's cars. ICA issues the alert signal if any pair are about to collide. All the parameters which were adapted to replicate the business models are given in Table 5.5 and Table 4.9 for SAU and bandwidth. For the SAU, we choose a similar threshold for license and resources for the convenience of the analysis; all the values for SAU or BW are considered either from Request For Proposal (RFP), which the telco company received (in our case Orange) or hit and trial base. This simulation is executed on the Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz 2.81 GHz, 16GB RAM, Windows 10. For the sake of simplicity, we assume that resources can be scaled easily. We generated SAU and BW randomly on each virtual node, also known as Virtual Evolve Packets (vEPC) such as Mobile Management Entity (MME), Service Gateway (SGW), Packet Gateway (PGW). After the SAU and BW were generated in each vEPC we implemented our license reference model as in Equations (4.8) and (4.9) and obtain the Figures 4.8 and 4.12 for SAU and BW. After the estimation of license reference, we estimated the license cost using the Equations (4.11) and (4.13), and its result is

shown in the Figures 5.5 and 4.13. After successfully evaluating the LC, we analyze the RC using the Equations (4.12) and (4.14) with the help of the license reference as shown in Figures 5.6 and 4.14. Further, we estimated the total cost using Equations (4.15) and (4.16) which were shown in Figures 4.11 and 4.15. The experiment is carried out three times with three different random values for thirty days, and its cumulative average results are presented in all figures.

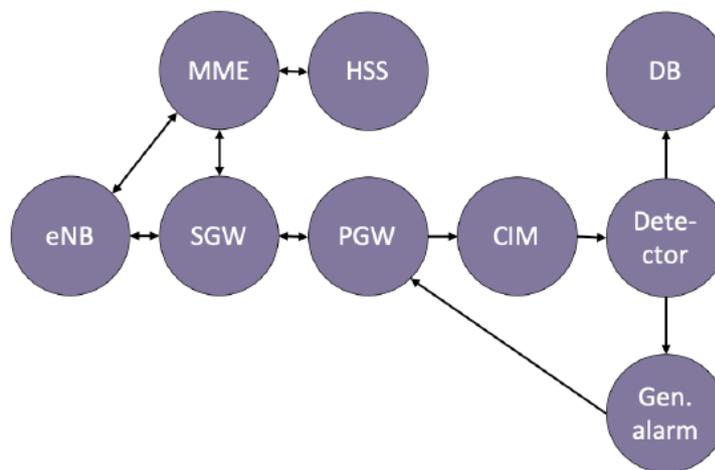


Figure 4.7: VNF graph of the ICA service.

Table 4.8: Simulation parameter for SAU.

Parameters	Value
License threshold for capacity	4725
Resource threshold for capacity	4725
$\sigma$ for both capacity for SAU	1.5 for LR less then 4725, 2 for LR > 4725
$\delta$	1.5 for SAU < 4725 and 2 for LR > 4725
$\theta$	33000 (€)
$\phi$	40000 (€)
SAU	500 add random (0,50), random (0,100) and random (10,1000)
$\alpha$ for SAU	10€ per LR for License
$\beta$ for SAU	6€ per LR for Resource
$CS_{PAYG}$	0.01(€) per SAU
$CSr_{PAYG}$	0.04 (€) per SAU
$LC_{prl}$	500 (€)
$RC_{prl}$	800 (€)

#### 4.4.4.4 Result Evaluation for multiple nodes

The main reason for undertaking this project is to develop a proper model that calculates the license and cost model, which helps optimize the total cost of VNF software, which is validated

Table 4.9: Simulation parameter for BW.

Parameters	Value
License threshold for capacity	65
Resource threshold for capacity	70
$\sigma$ for both capacity for BW	1.5 for LR less then 65, 2 for LR > 65
$\delta$	1.5 for SAU < 70 and 2 for LR > 70
$\theta$	700 (€)
$\phi$	600 (€)
BW	random (5,10), random (10,20) and random (30,40)
$\alpha$ for BW	10€ per LR
$\beta$ for BW	6€ per Resource
$CB_{PAYG}$	0.01 (€) per BW
$CB_{rPAYG}$	0.04 (€) per BW
$LC_{prl}$	500 (€)
$RC_{prl}$	800 (€)

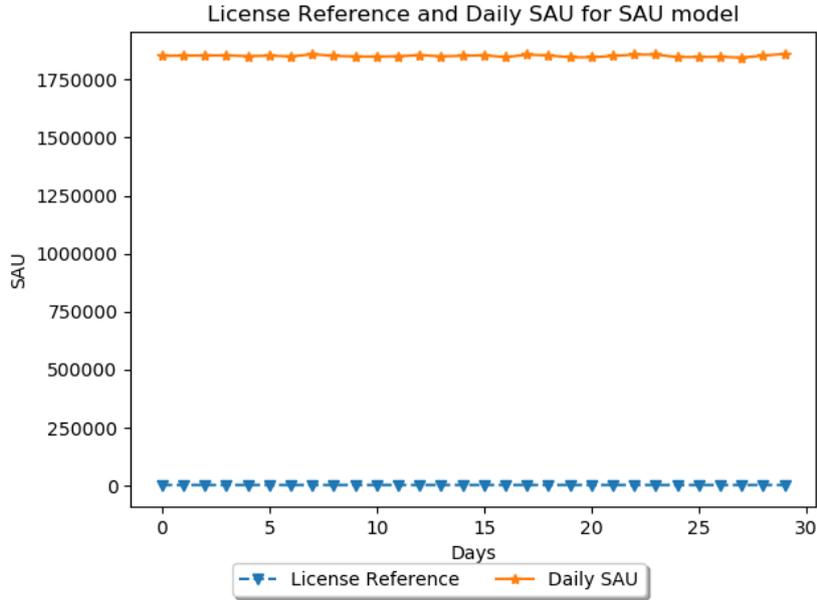


Figure 4.8: License Reference for SAU

by illustrating different outcomes. Figure 4.8 and 4.12 show the estimated license reference based on randomly generated SAU/BW. Randomly generated SAU/BW is also shown in the figures, with the help of which pay-as-you-grow and perpetual licenses and their related cost were estimated. We can see on both figures that daily SAU/BW is higher than the license reference. It is due to licensing reference being based on hourly average, maximum over a day from all concerned VNF from Equations (4.8) and (4.9). Not to be confused that daily SAU and BW shown in the figures are 24 hours consumption by ICA service.

Figures 5.5 and 4.13, which are the cumulative license cost for 30 days, we can see that

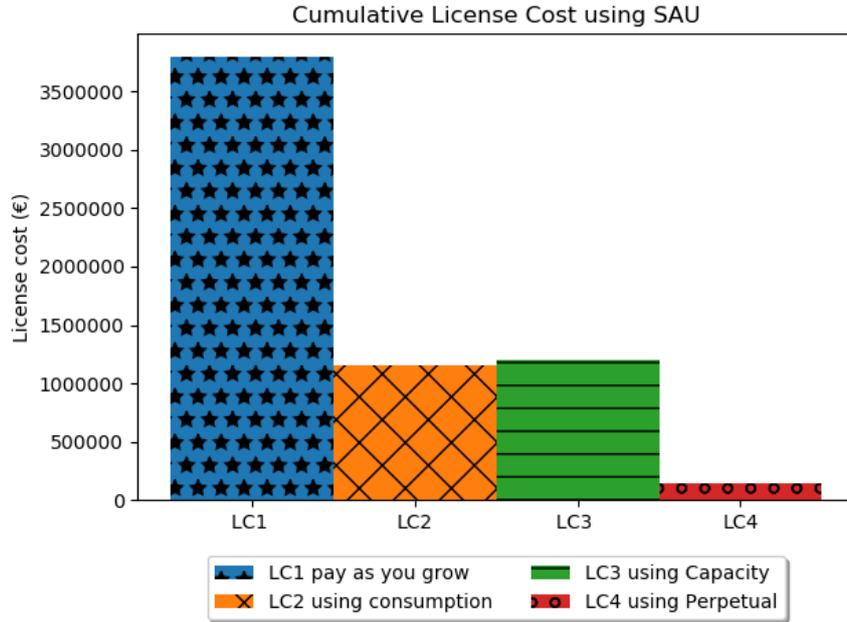


Figure 4.9: Cumulative License cost using SAU

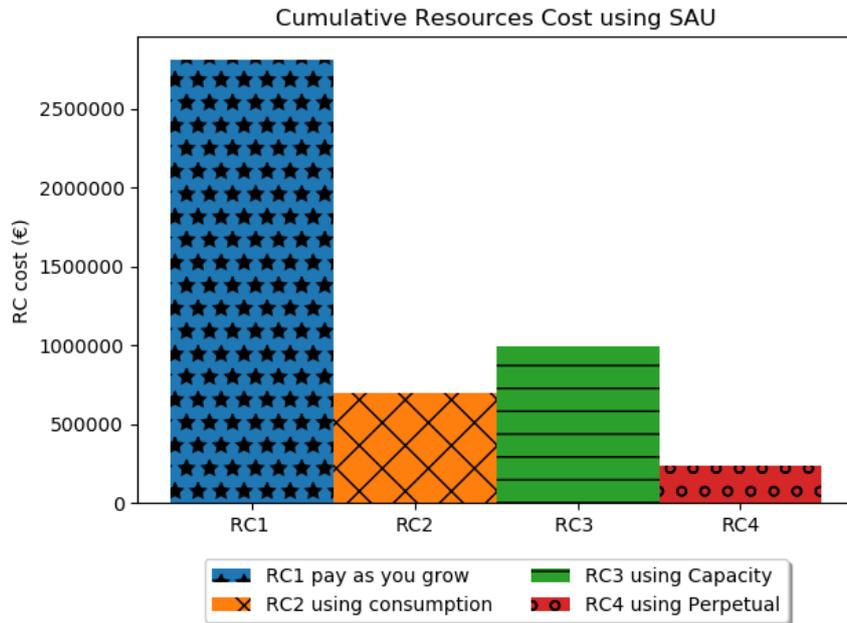


Figure 4.10: Cumulative Resource cost using SAU

license cost using perpetual is lower, and license cost using pay as you grow higher. An interesting case here is license cost using consumption and capacity methods. These are lower than pay as you grow; among consumption and capacity, consumption has a lower cost than capacity. One can argue that since the perpetual cost is lower, why not choose it? However, it is not beneficiary for the VNF services provider. Because the perpetual model did not consider

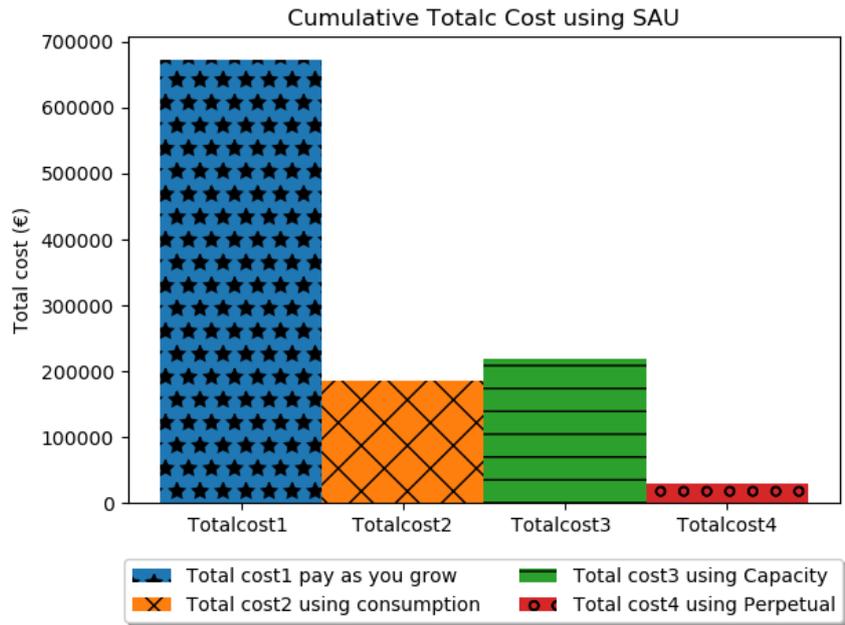


Figure 4.11: Cumulative Total cost using SAU

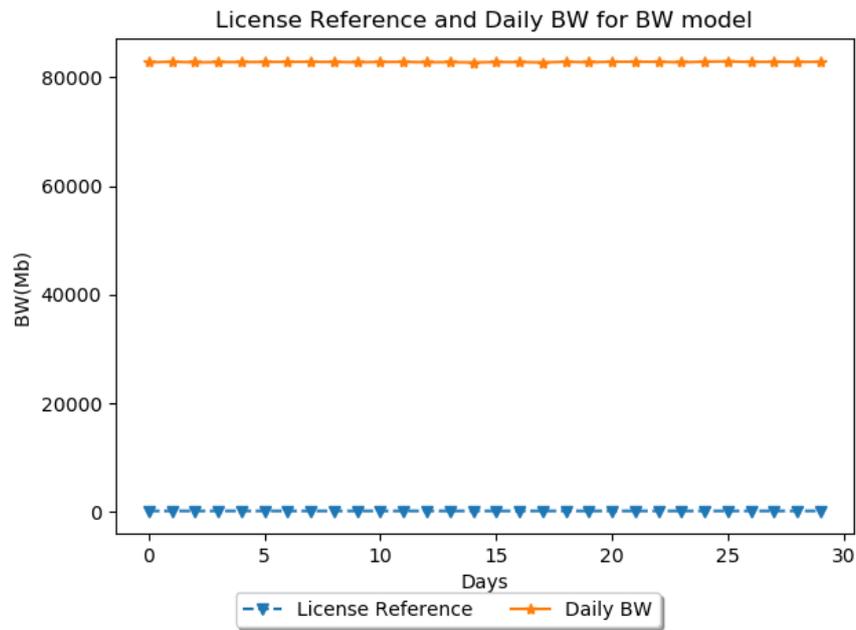


Figure 4.12: License Reference for BW

usage or resource consumption, it is not fair for the VNF services provider. Now, returning to our figures in contrast with the LC of SAU, BW (Bandwidth) LC for consumption is higher, and capacity is lower. Figures 5.6 and 4.14 show that the consumption model estimated the lower resources cost than the capacity in both SAU and BW. Figures 4.11 and 4.15 show the estimated total cost. Figure 4.11 is the total cost for the SAU here; we can see that the consump-

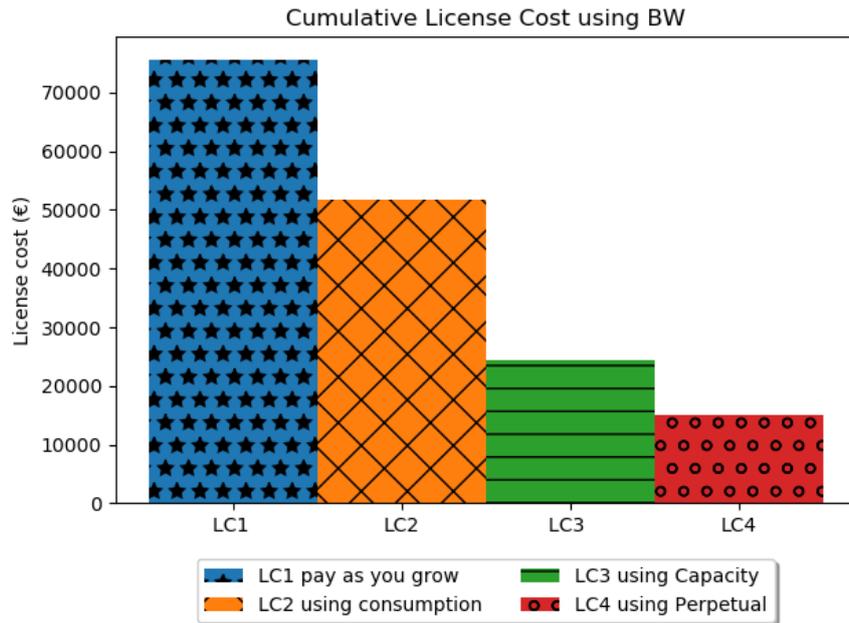


Figure 4.13: Cumulative License cost using BW

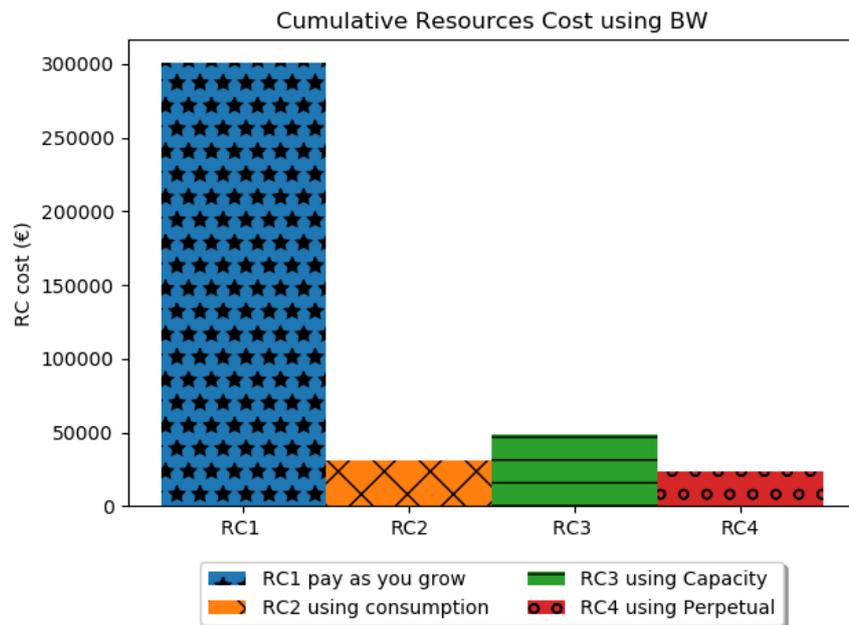


Figure 4.14: Cumulative Resource cost using BW

tion model estimated a lower total cost than capacity. While the BW capacity model estimated a lower cost than consumption, as shown in Figure 4.15.

One of the interesting points we can depict from these figures is that using SAU as a metric with our model’s consumption model estimates lower costs. Nevertheless, in contrast, the capacity model estimates lower costs when we use BW as a metric. This is because SAU and

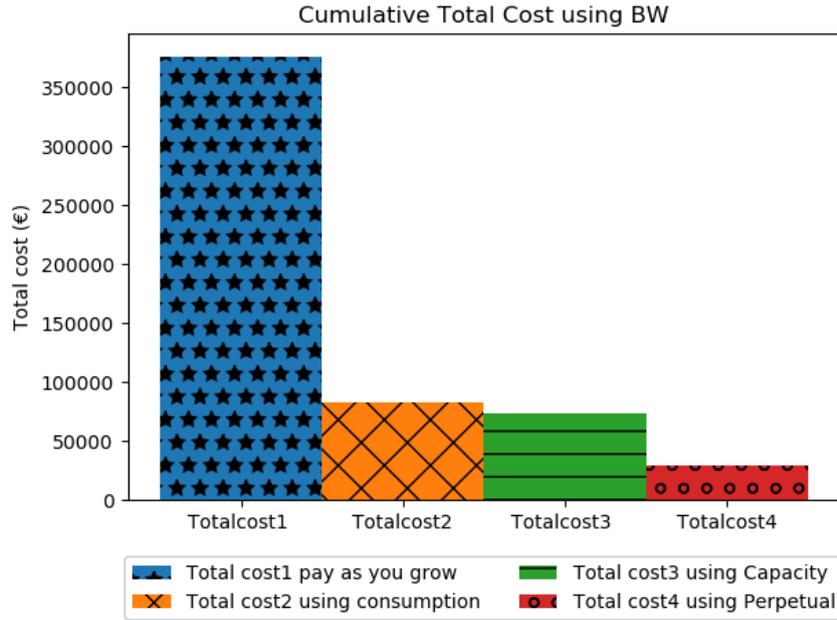


Figure 4.15: Cumulative Total cost using BW

BW cannot be compared to each other as they are two different parameters; one is related to the data plane (SAU), and another is the control plane (BW). Even though the SAU license reference is higher than BW, we got BW's total cost higher, which is because of the high unit cost in BW (refer to the Table 4.9). For this topology, this study does not recommend either SAU or BW is the best model but tries to present different available options to clients. In turn, clients can choose any model according to their needs and circumstances. For example, if in VNF1 an active user is 1 and the user is consuming bandwidth 1Gbps. In VNF2, 10 users consume 100 Mbps. If the SAU model is implemented in both VNFs, then VNF1's total cost will be far less than VNF2's, even though the user in VNF1 consumes more bandwidth. Thus, depending on the situation, it might be wise to use the bandwidth model in VNF1 and SAU model in VNF2. Thus, as shown in Figure 4.11, if clients wish to choose the SAU model, then for them to get optimum total cost, they can select the consumption process, and for BW, as shown in Figure 4.15, they can choose the capacity method instead of pay-as-you-grow or perpetual.

## 4.5 Conclusion

In this chapter, research was performed to analyze the drawbacks of existing VNF licensing and propose a new idea to formulate a licensing and cost model that would eventually optimize the software's total cost. Based on the need of users, the study has tried to suggest several models relevant to various scenarios. We have considered two categories of usage: theoretical usage

(capacity) and observed usage (consumption), which are calculated with the help of SAU and BW. Depending on the scenarios, different results were acquired using appropriate models that would help to generate licenses reference and estimate optimized total cost. Also, these models are applied in one network topology, which estimates the minimized total cost based on usage.

Furthermore, instead of proposing a single solution in this research, we have tried to propose several options that fit the complex licensing models in the industries. However, we also know that these models are non-exhaustive as they are still under consideration by standards and suppliers. So, licensing is a complex task that depends not only on one factor or metrics but also on several metrics, users, services, and many others. Therefore, we tried to include potential metrics and constructed a novel model. Thus, we assume that our SAU and BW models are not just limited to one scenario but could be implemented in different circumstances and topologies and help to estimate the optimized total cost.

Our chapter 5 work will enhance this model using Deployment Flavor (DF) and green energy and implement it in the more complex VNF scenarios.

# VNF Software Cost Modeling Based On Telecommunication Network

*"The fire that warms us can also consume us; it is not the fault of the fire."*

— Vivekananda

## Contents

---

5.1	Context . . . . .	61
5.2	Introduction . . . . .	62
5.3	Related Work . . . . .	63
5.4	Proposed model . . . . .	69
5.5	Conclusion . . . . .	83

---

## 5.1 Context

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, licenses, and cost management. Thus, to achieve this trade-off, this chapter first module the TCO of software cost of VNF based on three crucial costs, LC, RC, and EC. 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 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 thanks to the live migration feature of NFV.

## 5.2 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. Virtualization appeared to be not only an IT topic but also a networking opportunity. NFV help in the emergence of vEPC such as MME, Authentication Server Function (AUSF), Access and Mobility Management Function (AMF), Cloud-RAN (C-RAN), and other components, which are commonly referred to as Virtual Network Functions (VNF). NFV not only paves the way for SDN but also works as a complementary entity to SDN and network slicing. Undoubtedly, it will be a milestone for many upcoming generations [25]. Due to the virtualization of the networks, the concept of the Network as a Service (NaaS) increased network services commercialization [34].

We know that the use of NFV will increase due to its advancement and the different kinds of survey statistics we presented in Chapters 1 and 2, such as IDC. VNF is a virtual component of the NFV that can operate most telco functional entities. For example, Dynamic Host Configuration Protocol (DHCP), Domain Name Service (DNS), and Intrusion Detection Algorithm (IDA) 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, just as in software, TCO of software cost and management is necessary to achieve resources and cost optimization so as in VNF.

One of the common approaches to managing the software life-cycle management (deployment, changing, removing, updating, upgrading) and costs in IT is using SAM (Software Assets Management) that is further expanding to telco and other network sectors. Although VNF act as software, very few studies have been performed in terms of software cost [4, 35], etc.

Software cost is not only the purchase price of software to be used. It includes various other costs such as licenses, resources, software updates, upgrades, and maintenance. The license gives the right to use the software. Hence it is an unavoidable constraint for software. Licenses help frame CSPs' rights and obligations 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, 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 [36].

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 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 PoP or datacenter to another. The decision for VNF sharing and consolidating VNF is based on the requirement of resources, license, and both at SaaS and NaaS levels, which eventually help minimize the TCO with optimum resource utilization. This research fundamental goal is to provide a cost model to estimate the TCO of software for telecom (5G) architecture. After, we reduce the TCO using different VNF sharing and consolidating techniques. Finally, this chapter 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 report is organized as follows: Section 5.3 present related work. Section 5.4 present our proposed models and Section 5.4.4 simulation and Section 5.5 concludes our chapter.

## 5.3 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:*

Resources allocation is one of the tedious problems in VNF. Article [37] categorizes the VNF allocation problem in 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, [38] and [39], they presented resource allocation based on VNF forwarding graph embedding (VNF-FG). These articles are similar to those previously shown; they focus on VNF mapping and network traffic. The only difference between these two is that research [39] focused on 5G and beyond; still, they didn't consider the software

cost of VNF. In article [40] they studied the scheduling of VNF; this research helped us to know the relation between users and VNF, but they didn't consider VNF cost. Research [41] 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.

Thus, there are many types of research such as [24, 36, 42, 43] that present various ways for VNF placement latency, energy consumption, and optimization of resources; thus, these show the importance of VNF resources. But they didn't consider software cost. Some of the research [34] 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. But, all this past research focuses much more on network traffic, traffic steering, VNF placement, energy consumption, and load balancing than on the software cost of VNF.

b) *VNF cost in general:*

In [44] article was based on QoS of the non-cooperative 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 [33] provided the FlexShare algorithm, which gives near-optimal deployment cost, but it did not consider the software cost perspective. In research [45] 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 [46–48] that presented the operational cost, which deals with cost, and network. They provide 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 research.

### **5.3.1 System Model**

This section provided a system model for the VNF software cost and sharing. Resources sharing and consolidation are well-known strategies for optimum use of resources, but using these strategies for software sharing is a less explored area. Let us see the different parameters of software cost adapted in this research.

#### **5.3.1.1 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, 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 (5.1);

$$LR_{SAU} = \max_{\forall p \in D} \left( \sum_{\forall p \in V} average(SAU, p) \right) \quad (5.1)$$

where,  $V=(1,2,3,\dots,v)$  is a set of all concerned VNFs (VNFs can be the same or different types) in the node. For example,  $D$  is a set of days (1,2,3,...,  $D$ ). SAU are users who are active in VNF, using services provided by VNF. Of course, there are other metrics for estimating LR, such as transmission per second, bandwidth, and subscriber. However, 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 in two ways based on usage: capacity and consumption. We have discussed this in chapters 3, 4, and article [35] in detail. 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 research, we use the capacity module. We have the initial cost  $\rho$  up to the LR (threshold). The extra cost will be incurred if the license's demand exceeds the LR. Equation (5.2), (5.3) show our LC.

if LRQ (License Required) is  $\leq$  LR

$$LC_i = \sum_{i=1}^n \rho_i \quad (5.2)$$

if LRQ (License Required) is  $>$  LR

$$LC_i = \sum_{i=1}^n (\rho_i + \beta_i \times \omega_i) \quad (5.3)$$

2. **Resource Cost (RC):** For this study, resources include the virtual resources that operate VNF. Our virtual resources are mainly vCPU and vRAM, but they are not limited to this. It includes data centers, Virtual Machines (VM), Virtual Network (VN) resources, etc. We consider the total resources of VNF with 100% (this is not an assumption; it is well practices in telecom industries like orange) 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 Figure 5.1. 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%. The percentage use over here can be different; it is not fixed to 40 or 60; it can be different depending upon the VNF provider. With the help of Orange, France, we use these values. Resource costs can be expressed as follows;

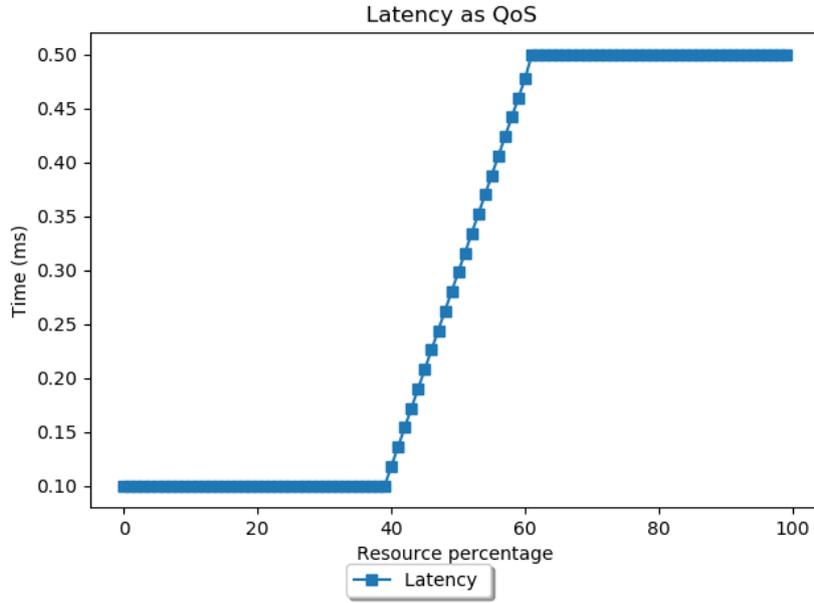


Figure 5.1: Latency (QoS) of a VNF

if LRQ (License Required) is  $\leq$  LR

$$RC_i = \sum_{i=1}^n \eta_i \quad (5.4)$$

if LRQ (License Required) is  $>$  LR

$$RC_i = \sum_{i=1}^n (\eta_i + \alpha_i \times \gamma_i) \quad (5.5)$$

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 [49]. It can be formulated as Equation (5.6) depending upon power consumption and unit energy cost. It is measured in Joule or kWh.

$$EC_i = \sum_{i=1}^n (\phi_i \times \delta_i) \quad (5.6)$$

4. **Total Cost of Ownership (TCO):** The total cost of ownership for software includes indirect and direct costs incurred by the software, which plays a critical part in ROI. As for now, in this research, we only considered RC, LC, and EC for TCO because they are three which have a significant impact on TCO. Licenses encompass the rights to use the software. Resources determine the capacity of software performance, and energy consumption depends on resources. From this, we can easily see that TCO 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:

$$TCO = \sum_{i=1}^n (RC_i + LC_i + EC_i) \quad (5.7)$$

and our objective is to minimize this TCO providing the QoS  $TH_f \preceq D_f$

$$\min TCO = \min \sum_{i=1}^n (RC_i + LC_i + EC_i) \quad (5.8)$$

$D_f$  = Throughput, SAU, latency or any KIPs of clients needs

5. **Power consumption of VNF:** Estimating the energy consumption of the VNF, power consumption of each VNF is very important. We follow a similar paradigm presented by ETSI [49] 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 research, the power of VNF is set following the guideline given by ETSI. The power module of VNF proposed by ETSI and given an arbitrary value to each VNF. VNF Energy Efficiency (EE) is estimated as in Equation (5.9) which is related to power.

$$P = P_{load} - P_{idle} \quad (5.9)$$

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 energy conversion rate.

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;

$$VNF_{EER} = \frac{Useful_{output}}{Power_{consumption}} \quad (5.10)$$

Usefuloutput can be expressed as Packets/s, subscribers, and transmission per second. In our work,  $Useful_{output}$  is the SAU server by VNF, and the power consumption of VNF ( $Power_{consumption}$ ) is for the moment we use an arbitrary value.

### 5.3.2 Sharing and Consolidation Algorithm

VNF sharing and VNF consolidation are similar to many extents, but there are significant differences;

- *VNF sharing*: It shares 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 consumption. In our research, we share the resources after 60% of resources have been consumed by implementing our heuristic approach.
- *Consolidation*: Consolidating 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 when VNF can share resources up to or more than 40%. In such a 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.

We classified software sharing and consolidation into two types:

- **Vertical Sharing**: It shares resources, 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 where a replica of VNF is used for load balancing and redundancy. Similarly, horizontal consolidation is also the process of moving the VNF within the same network, Point-Of-Presence.

## 5.4 Proposed model

This section presents three types of sharing algorithms and one classic model for comparison. We modified these algorithms to fit our models.

- **Resources Methods for Share (RMS)** : It is the method based on resources for consolidation. 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, generated random users in each VNF, then calculated LR. Then in the second step, again, users were generated; this time, we associated 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 line 6, conditions are checked if VNF can share resources; if  $VNF_i$  can share a resource with another VNF,  $VNF_l$ . The resource requirement of  $VNF_l$  must be lesser than the other VNF in the network that requires resources. If the condition is satisfied, line 8 checks if VNF can share resources more than or equal to 40%; if yes, then in such case, we consolidate the VNF to the other PoPs where the RC requirement is less among others. Whereas if no, only resources will be shared as in line 11. Another resource share is happening at line 13; line 13 is the summation of all resources available for sharing by each VNF. Thus, sharing is taken place in two steps. Lines 6 and 13 One is with each VNF, and the other is the summation of all VNFs.

This can be clarified with one example if the VNF3 resources requirement is 10% and VNF1 and VNF2 can share 5% each, VNF1 and VNF2 individually 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 (5.3) to Equation (5.7).

- **License Method for Share (LMS)**: 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 of license requirement. In this research, each user needed to be licensed, and each user consumed some amount of resources. Thus there is a

---

**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
8         in the system, such as  $RN(t)_l < RN(t)_j, RN(t)_k$ .
9       if yes check if resources can be shared  $\geq 40\%$ 
10      if yes consolidate VNF, and then share
11      Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
12      if no share resources only Resources are shared with VNF
13       $RN(x)_l \leftarrow Rs(x)_i$ ;
14       $TCO_{RMS} = RC_{RMS} + LC_{RMS} + EC_{RMS}$ ;
15    else if  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  then
16      Check if this VNF has the least resources or not with all the VNF in the
17      system, such as  $RN(t)_l < RN(t)_j, RN(t)_k$ .
18      if yes and Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
19       $TCO_{RMS} = RC_{RMS} + LC_{RMS} + EC_{RMS}$ ;
20    else
21      Resource cannot be shared need of extra resources
```

---

relationship 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 requirement in the VNF. Only line 6 and 13 is changed.

For example, if the license requirement in VNF1 is 10 and VNF2 is 5, resources will be in VNF2 if they need resources because the license requirement is less.

- **License and Resource Methods For Sharing (LRMS):** It is the combination of the methods mentioned above; resources and licenses. Only lines 6 and 13 are changed. 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

---

**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 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 license required or not with all the
8       VNF in a system such as  $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$ 
9       if yes check if resources can be share  $\geq 40\%$ 
10      if yes consolidate VNF and then share
11      Resources are share with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
12      if no share resources only Resources are share with VNF
13       $RN(x)_l \leftarrow Rs(x)_i$ ;
14       $TCO_{LMS} = RC_{LMS} + LC_{LMS} + EC_{LMS}$ ;
15    else if  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  then
16      Check if this VNF has least license required or not with all the VNF in a
17      system such as  $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$ .
18      if yes and Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
19       $TCO_{LMS} = RC_{LMS} + LC_{LMS} + EC_{LMS}$ ;
20    else
21      Resource cannot be shared
```

---

resources to form VNF, which is sharing. A pseudo-code is presented in Algorithm 3.

- **Classic Method (CLM):** In this method, resources are shared without any constraint; if VNF can share its resources, it will simply share with other VNFs as 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. All these modules and algorithms are used in Section 5.4.4.

### 5.4.1 Deployment Flavor (DF)

In this study, we use the concept of flavor for the VNF. The flavor is based on VNFC (Virtual Network Function Components) vCPU, vRAM, and Virtual Storage (vStorage). For this study, we constructed the flavor based on vCPU and vRAM as shown in Table 5.1. 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

---

**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
8       with all the VNF in a system such as  $LN(t)_i < LN(t)_j$  and
9        $LN(t)_i < LN(t)_k$  and  $RN(t)_i < RN(t)_j, RN(t)_j$  and soon.
10      if yes check if resources share  $\geq 40\%$ 
11      if yes consolidate VNF and then share
12      Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
13      if no share resources only Resources are shared with VNF
14       $RN(x)_l \leftarrow Rs(x)_i$ ;
15       $TCO_{LRMS} = RC_{LRMS} + LC_{LRMS} + EC_{LRMS}$ ;
16    else if  $\sum_{i=1}^S RS(t)_i > RN(t)_l$  then
17      Check if this VNF has the least resources and license required or not with
18      all the VNF in a system such as  $LN(t)_i < LN(t)_j$  and  $LN(t)_i < LN(t)_k$ 
19      and  $RN(t)_l < RN(t)_j, RN(t)_k$ .
20      if yes and Resources are shared with VNF  $RN(x)_l \leftarrow Rs(x)_i$ ;
21       $TCO_{LRMS} = RC_{LRMS} + LC_{LRMS} + EC_{LRMS}$ ;
22    else
23      Resource cannot be shared, add extra resources
```

---

affects software TCO. Thus, we have used DF along with resource sharing in our simulation and presented its impacts.

## 5.4.2 Use case

The primary purpose of presenting this use case is that we want to show how the users can optimize energy costs and their impact on TCO and the environment. We adopted the 5G Gaming use case from article [50]. 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.  $C_{O_2}$  emission per Kw is 0.709kg, and electric cost per Kw is 0.108€. Thus, we formed the TCO for a day as shown in Table 5.3.

---

**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
```

---

Table 5.1: 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

So from this case, we can see that different VNF provider provides the same services with different TCO and carbon footprints. We can see from Table 5.3 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 offers good performance with a bandwidth of 1 Gbps, but it comes with huge costs, and its carbon footprint is higher than any other VNF. So it is up to the client what they want. If they are not willing to contribute to the planet and only need the best performance, they can go for company C VNF. Still, 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 needs and systems.

Table 5.2: Key Notation

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

Table 5.3: Different VNF providers

(NAT: Network Address Translator, FW: Firewall, TM: Traffic Monitor, WOC: WAN Optimization Controller, IDPS: Intrusion Detection Prevention System, VOC: Video Optimization Controller)

VNF Providers	Type	SFC 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 5.4: VNF cost based on different methods

VNF	Instances	Unit cost (€)	Cost/day	Cost/month
VNF method 1	50	0.1	5	150 €
VNF method 2	50 (12 days), 10 (18 days)	0.1	5 (12days), 2 (18days)	96 €

### 5.4.3 Utility Based

It is another method for estimating the cost. Where the customer is charged according to the time product is used, pay per use. This technique can be very much helpful in various scenarios. For example, let us take a football stadium; in a football stadium, generally, the game is on Friday, Saturday, and Sunday. It means the stadium needs more quality communication network services on these three days. The number of users on different days is shown in Figure 5.2. Here we can see that traffic is heavy on weekends. Now, let us consider that on the weekend, the stadium needs 50 VNF instances (1 VNF instance = 1vCPU,1GB vRAM), and on the other days, they require 20. Each instance costs 0.1€. Then we can see that considering the maximum instance (50), it costs  $50 \times 0.1=5$  €. The total cost for a month will be  $30 \times 5=150$ €. However, if we use utility-based methods, we use 20 VNF instances for weekdays and 50 for the weekend. Thus, it takes only extra 30 (50-20) instances only for the weekend. If we take 4 weekends in a month, that means 12 ( $4 \times 3$ ) days with 50 instances which will cost  $12 \times 50 \times 0.1=60$ , and the remaining 18 days will be  $20 \times 0.1 \times 18=36$ €. The total cost will be  $36+60=96$ . This can be seen in table 5.4.

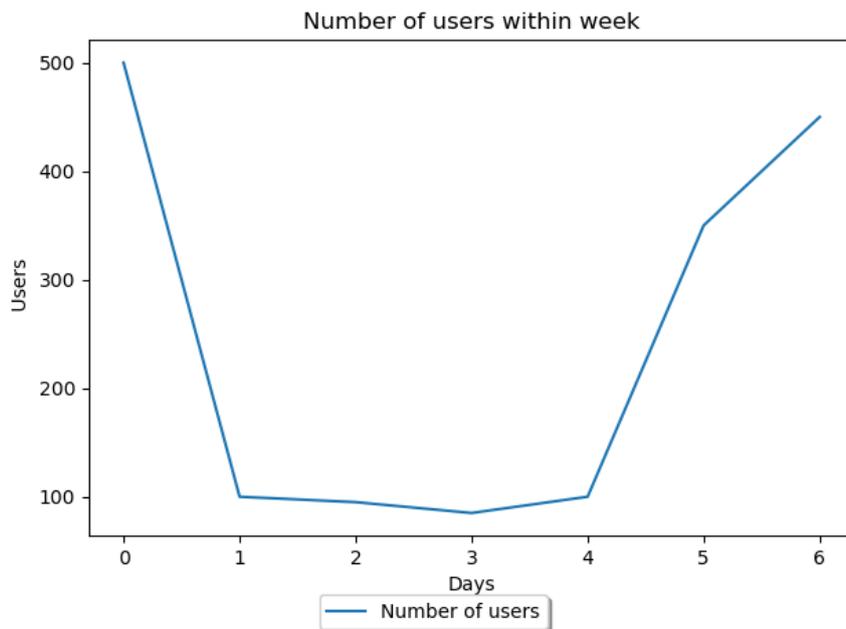


Figure 5.2: Traffic evolution in stadium with amount of users

VNF method 1 does not use a utility-based process, and VNF method 2 uses a utility-based method. We see the difference in the prices is a huge difference. This will help clients or VNF providers to reduce cost expenses. When the stadium is not used, other clients can use those resources, which means there is no need for extra resources to be added. This decreases the energy consumption in data centers and will help reduce  $C_{o_2}$  emission. This method not only helps to reduce the cost of VNF users; it also assists in contributing to the green planet.

#### 5.4.4 Simulation setup and evaluation

For our studies, we used the Mobile data core (EPC+5G Core) as shown in Figure 5.3. The Mobile data core is the heart of the mobile network. Figure 5.3 shows the different generation mobile data core. It provides solutions for voice and data services: voice over LTE/WiFi, M2M/IoT/B2B, Enhanced Mobile Broadband (emBB), Massive Machine Type Communications (MMTC), Ultra Reliable Low Latency Communications (URLLC), QoS differentiation and any IP application depending upon the generation (3G, 4G, 5G) it belong. In 5G Mobile, the data core is divided into several NFs; each NF handles subscribers, sessions, or transactions. For the VNF, each VNF consumes NFVI resources which are vCPU and vRAM.

Among different generations. We used the 5G architecture as shown in Figure 5.4 because VNF is used in the 5G generation. Which we got from the telco industries, orange, France. Our focus is on the 5GC (core) network that is supported by VNFs. Which are Policy Control Function (PCF), Unified Data Management (UDM), AUSF, AMF, and Session Management Function (SMF). For each VNF, we calculate the LR using Equation (5.1). After estimating LR in these VNFs, we considered that each user is consuming some % of resources, basically from (0.1- 0.5)% and (1- 10)%. All the parameters used in this execution are presented in the Table 5.5. All these values are closely related to the value we obtain from a different vendor in our industry (Orange, France). Then associating resources to each VNF, we checked whether the user (SAU) and resources exceeded the threshold. 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 Figure 5.5 to 5.10. 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. For estimating the resources, license, energy, and TCO. We use Equations (5.3) to (5.7). For the EE and latency, we consider that when the resources needed by VNF are fulfilled by sharing between the VNF, then latency and EE will not degrade. DF used in the

Table 5.5: 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 [50]
Initial LC for CLM	1200 €
Initial RC for CLM	3500 €
Latency without sharing	0.5
$\rho$	2000
$\eta$	1000

simulation corresponds to vCPU and vRAM. DF1 is with 2 vCPU and 2 GB vRAM in which each user consumes resources between (0.1-5), and DF2 is with 2vCPU and 2 GB vRAM in which resources consumption is between (1-10)% by each user. DF3 is with 4GB vCPU and 4 GB 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. We use (1-10)% resources in all three DF (DF2, DF3, DF4) for the simplicity of calculation only. We found that when we keep changing resources % in CLM methods, cost increases significantly, and in other methods, cost increases slowly.

#### 5.4.4.1 Evaluation of Results

1. **License cost.** Our methods play a significant role in LC reduction. For example, Figure 5.5 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 a higher cost than others but less than classic. This is because, at that instant in DF2, VNF required extra resources and licenses, which was not fulfilled by sharing. Thus, VNF sharing in these cases didn't happen. But CLM has a high LC.

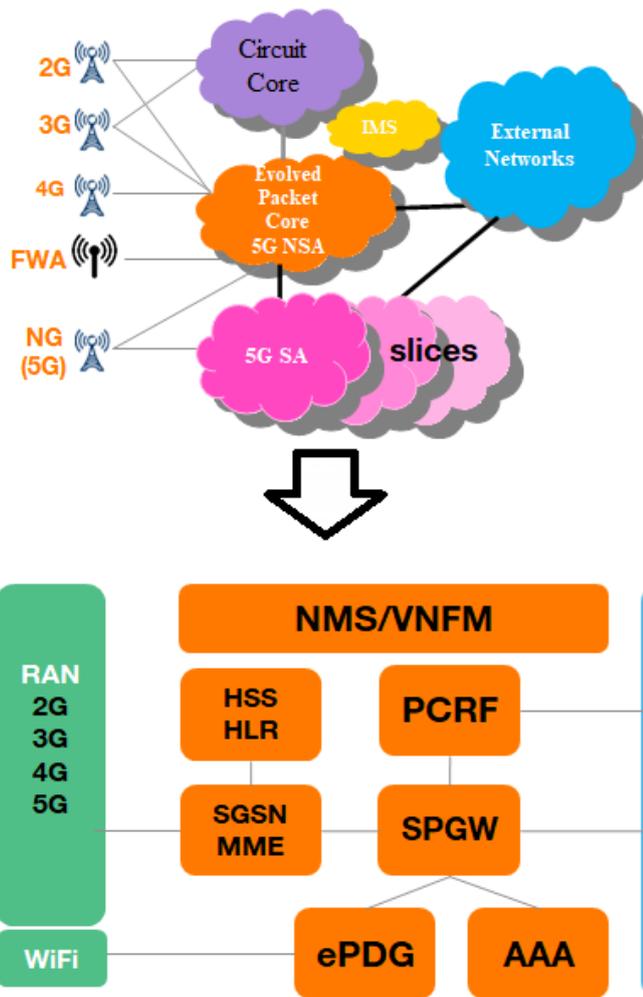


Figure 5.3: Mobile data core

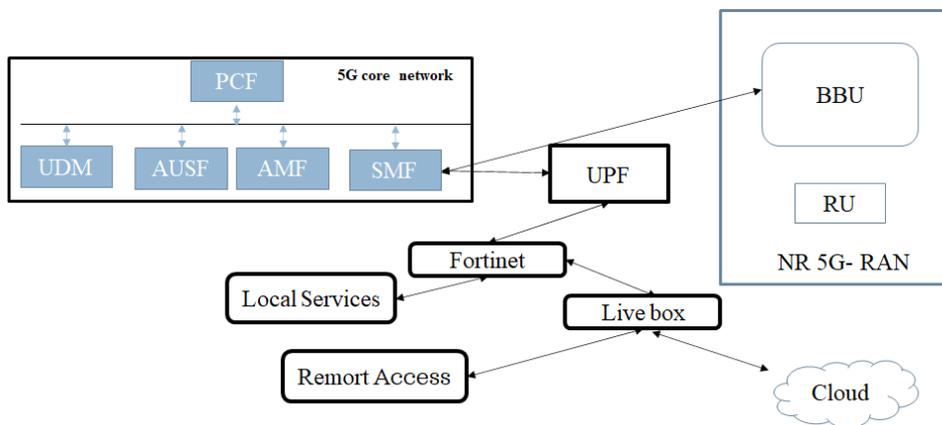


Figure 5.4: 5G Network Architecture.

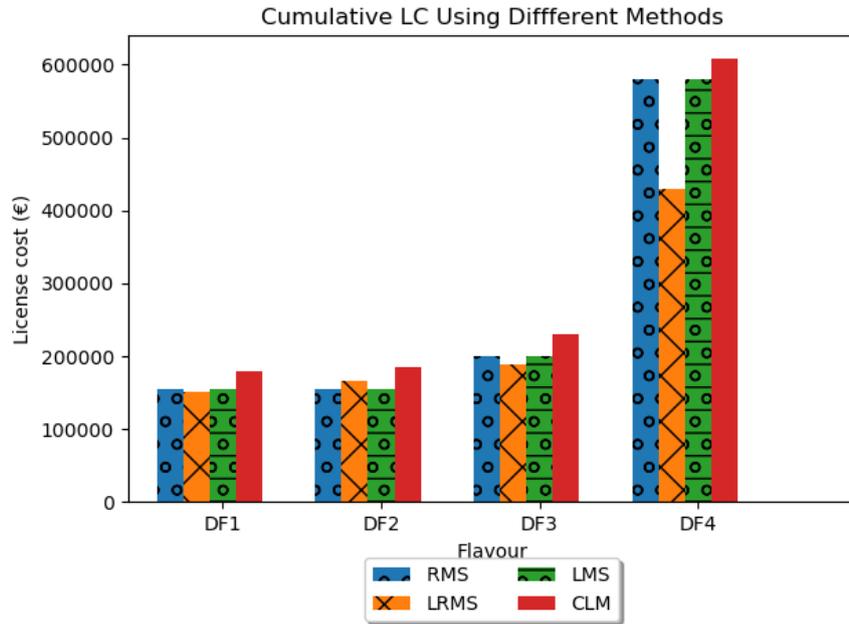


Figure 5.5: License Cost Using Different Methods

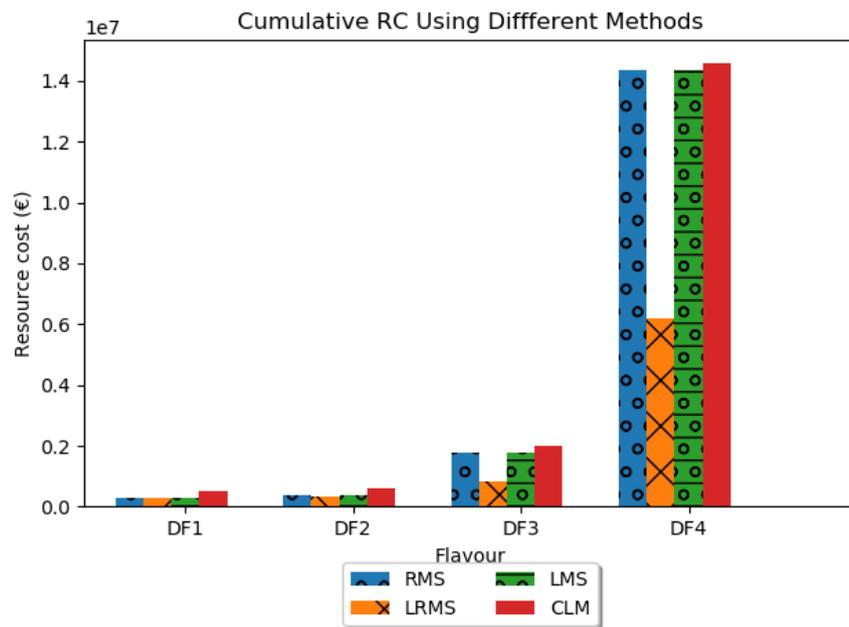


Figure 5.6: Resource Cost Using Different Methods

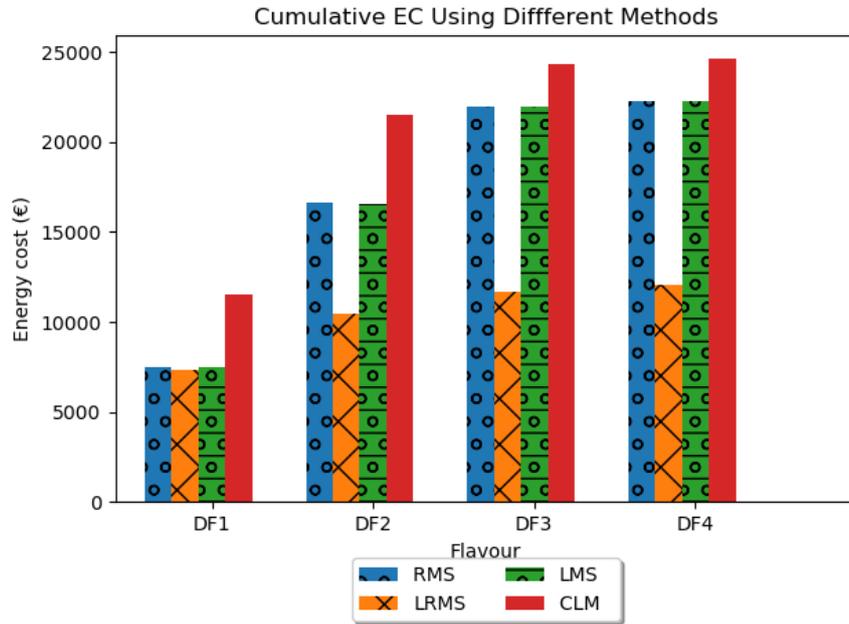


Figure 5.7: Energy Cost Using Different Methods

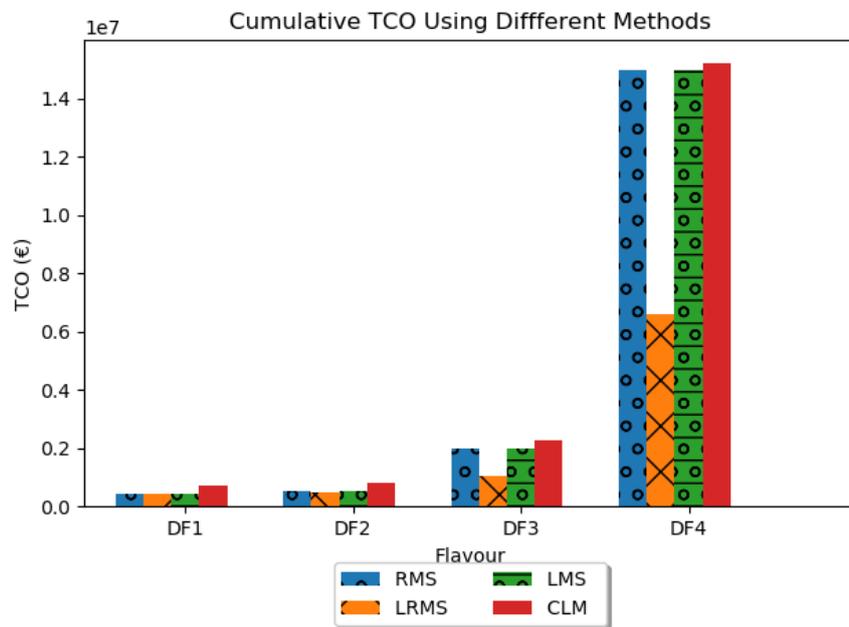


Figure 5.8: TCO Using Different Methods

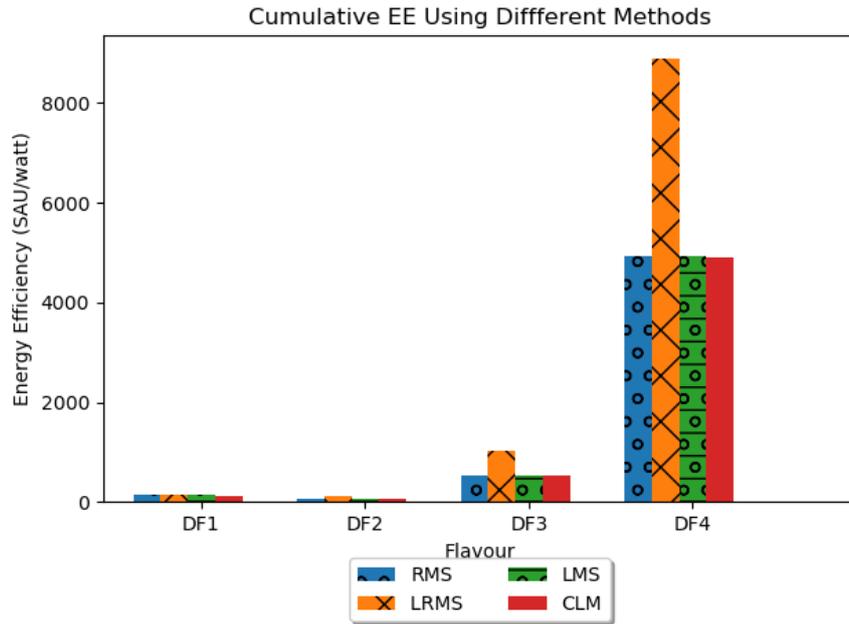


Figure 5.9: Energy Efficiency(SAU/watt) Using different methods

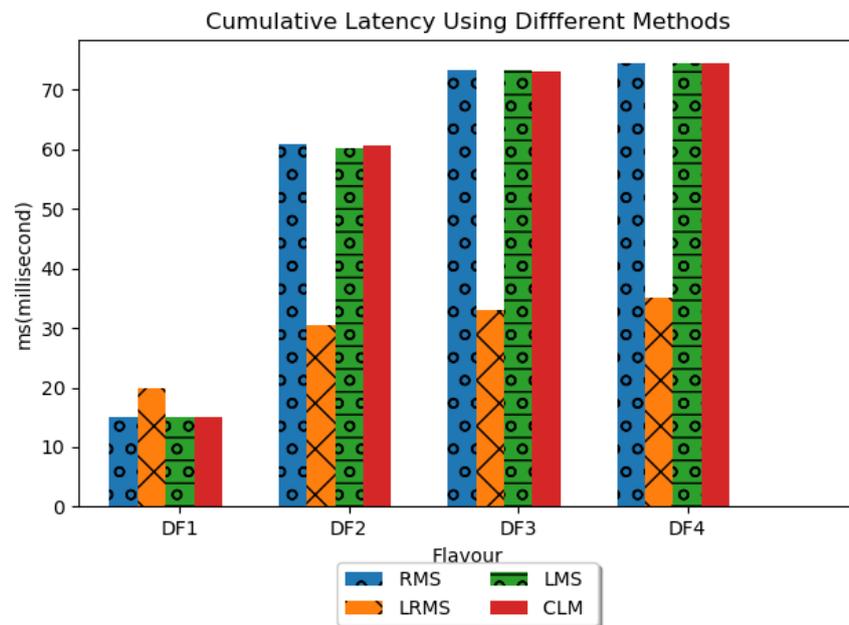


Figure 5.10: Latency (ms) Using Different Methods

2. **Resource cost.** RC is another parameter where our methods play a vital role in optimum use and reducing RC. Figure 5.6 shows that LRMS estimated lower cost in almost all DF except in DF1 and DF2; this is equal to the other two methods (LMS, RMS). Although it is equal, it still estimates the lowest cost than CLM. Because, in DF1 and DF2, user and resource consumption are both less; they did not surpass the VNF license or resource usage at the same time, or even if they required the resources, it is not so much huge. Even if they require resources, it is less. However, for the LMS and RMS methods, resources cost is highest 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.
3. **Energy cost.** Our energy cost is related to resources, so its nature in Figure 5.7 is similar to RC. LRMS outperformed the other methods. This means fewer extra resources are added, and more resource sharing is performed in each case for LRMS. Energy cost is less when resources share, or consolidation has taken place. In LMS and RMS, energy cost is similar because the license and resources required are either compensated by sharing or adding resources simultaneously. However, CLM estimated the highest cost due lack of the proper implementation of resource sharing.
4. **Total Cost Of Ownership.** This is our primary objective to module the software TCO and minimizes it. Figure 5.8 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 techniques. This is because more VNFs required licenses and resources at the same time. Whereas in LRMS resources are share properly. Also, we can see that as DF changes, LRMS shows significant results because as DF changes resource capacity, the number of users and quantity to share or need for resources is changed. Finally, we can see that both other resource (RMS) and license (LMS) methods are similar. This happened because each user must be licensed and consumes some resources. Due to this, we say that licenses and resources are interdependent and cannot be neglected.
5. **Energy Efficiency.** It is the amount of SAU served per watt by VNF. From Figure 5.9 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 resources share available and possible. Therefore, more numbers of SAU are being served by consuming less energy. Also, as DF changes from DF1 to DF4 capacity of VNF increases. Thus more resources can serve more SAU and also sharing.
6. **Latency.** Figure 5.10 shows 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

except in DF1. It is also due to the efficiently sharing of resources. In DF1, LRMS is the worst among other methods because the license and resources required in VNF are not fulfilled by sharing; thus, extra resources are added. As we mentioned earlier, our latency depends upon the sharing. If sharing is possible, the cost will not augment if sharing is not possible, it will increase. Because when sharing is not possible, they need extra resources, so they may have to wait longer.

Thus, we can see that LMRS outperforms every method in flavor and cost. This is because SAU and resource consumption is related; license and resources are interdependent. Therefore, these two parameters should not be neglected. Also, as the DF changes, capacity in terms of resources and licenses increases. Thus, it can share more or less VNF breach the threshold (SAU, Resource).

## 5.5 Conclusion

In this chapter, we presented the study on VNF software cost based on 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, 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 use these results among complex and commercial 5G deployments.

# Conclusion

*"Nature does not hurry, yet everything is accomplished."*

— Lao Tzu.

## Contents

---

6.1	Summary . . . . .	84
6.2	Contributions . . . . .	85
6.3	Future perspectives . . . . .	87

---

## 6.1 Summary

The evolution of softwarization and cloudification rapidly shifts current technologies towards virtualized ones. Observing the current trends, we can now say that most technologies will be virtual in the future. Thus, we need to be prepared for this shift, not only in terms of infrastructure. But also in terms of economic aspects. We need to be prepared in terms of the tools and methods needed to generate maximum profit or revenue from them. As we already present in this report, among many aspects of revenue, enterprise software cost has a significant role among them. In the future, it will be much more, too, as we are shifting toward the *softwarize* world. For example, software related to engineering are expensive. Licenses for typical engineering applications cost from 1,500 - 4,000 dollars per license per years [51]. Thus, there is the possibility of overspending if users are not on the top level of license management.

Among many virtual technologies, NFV is one of the current virtual technologies which help to virtualize the network function. With the help of its virtual component, VNF. NFV virtualizes the network function like Firewall, proxy, and NAT (Network Address Translation).

Thus, VNF behaves as software, and like software, VNF needs to be licensed too. Licensing in VNF is challenging due to the lack of proper models and metrics. Here in our research, we address these challenges by focusing on the following topic;

- **Estimating license with feasible metrics:** In our research, after analyzing the present situation of the software cost. We found that metrics are essential for estimating the usage of VNF. Thus, we used metrics SAU and BW. Using these metrics of VNF, we estimated the license required for VNF under time consideration.
- **Software cost:** Traditionally and still today, software cost estimation is significant for software users and provider. Because in an organization, roughly 40-50% investment is on software, and as we know, in the future, it will increase more because of the replacement of hardware with more complex software. It is crucial to estimate and minimize the software cost so that companies will not suffer from a loss of revenue. Hence, in this research we formulate the total software cost of VNF.
- **License:** Licensing is critical to track the usage of any software. The service provider gives the VNF user the right to use the VNF in the agreed terms and conditions. i.e., license. License is crucial to check the compliance of software. Thus, the license is vital for VNF. Furthermore, proper estimation of the license is essential to get control over the software cost.
- **Resource of VNF:** The resource consumption of VNF by the user helps to determine how the user needs to be licensed and charged. The resource is needed to run the services or functions. Resource consumption depends on how many users are using the VNF services. Resources consumption in terms of vCPU and vRAM. The resource is interdependent with the license, significantly impacting software cost.

## 6.2 Contributions

In this research, our primary contribution is the techno-economic analysis of VNF. We can summarize our contribution in the following ways;

1. **Licensing module for VNF.** In our research, we use two simple but very efficient metrics. First, we construct the model for licensing VNF. Two metrics are SAU and BW. These are two separate metrics that help to track the usage of VNF. With the help of these two metrics, first, we construct the license reference module based on the hourly average maximum over a day (BW/SAU). After estimating license references, we estimate license and resource costs, which ultimately form our software cost. This way, we show how software costs can be constructed using license and resources cost.

2. **Implementing deployment flavor with license.** In our research, we used the popular concept of the industrial field, a flavor named Deployment Flavour (DF). In our context, DF depends on the virtual resources, vCPU, and vRAM. vCPU and vRAM determine the number of Simultaneous Active Users (SAU) that can operate and the amount of bandwidth (BW) it can serve to SAU.

In various use cases from Chapters 3 to 5, we implement the DF and show how license cost and resources cost can be reduced using it. First, in chapter 3, we use DF alongside SAU and different VNF providers to offer how a client can reduce software costs. Then, in Chapter 4, using DF, we constructed the flavor table for SAU and BW and proposed the various options to clients depending on their needs and scenarios. Finally, in chapter 5, we implement DF with VNF sharing and show the process of reducing the TCO using various heuristic methods.

3. **VNF sharing** We implement VNF sharing in our research with different use cases and scenarios. In Chapter 5, we implement the VNF sharing alongside VNF consolidation. We used the VNF sharing with a heuristic algorithm based on resource, license, or both. This helps to reduce the license and resource cost of VNF compared to classic methods. We also show how VNF sharing helps reduce energy consumption. Thus, we use the concept of VNF sharing with licensing resources to reduce the TCO of VNF.

4. **License with QoS** It is harder for most people to believe that QoS impact the license, but it has. Our research's main objective is the minimization of TCO providing QoS. Although, we did not explore QoS much in this research because it is not our scope area. QoS is necessary for software licensing when licensing is done with a floating license. A floating license is sharing a limited number of licenses throughout the organization. A floating license is cost-effective if it is appropriately managed. When the user runs the application license manager provides the license as long as the pool is available. When the user can not get a license, then the user receives a 'denial.' This means that the service level will be down. For example, if the user asks for a license 20 times and gets it, then we can say the QoS level is 100%, but if the user request is denied 10 times, QoS drops to 90 %. Also, QoS information provides good information for clients or organizations regarding licenses such like;

- Whether to renew a license or not, maintenance, procure an additional license
- How many licenses have been used at the same time? Furthermore, what is the percentage of their usage time?

Thus, QoS help to understand whether the organization is under license or over the license, how many licenses are required to meet the desired QoS, or how many licenses

they can get rid of to be in the desired QoS.

## 6.3 Future perspectives

Our work opens several perspectives regarding NFV and virtual software license management. This section will discuss how our work opens new research areas.

- **Different metrics for different types of Virtual Network Function**

Our study uses SAU and BW to estimate license requirements in VNFs. These metrics work perfectly for us, but for future research, we believe that to be more precise and updated with fast-growing virtualization technologies. It may be better if different metrics for different VNFs depending upon the function they performed, were considered. For examples for vEPC domains, Simultaneous Active User (SAU) for **MME/SPGW**, Active Users (AU) for **PCRF and ePDG**, vCPU/Instance for **VNF Manager**, this list is non-exhausted.

Depending upon the evolution of technologies, metrics will be changed. We worked with SAU and BW and got promising results. A more precise result may be gained by using another metrics. Also, the formula we used for estimating the license reference, which is hourly average, maximum over a day based on the SAU or BW, can be modified or revised too. Therefore, future researchers should not limit themselves to these formulas only.

- **Deployment Flavour (DF) of VNFs**

In our study, we use DF based on resources, vCPU, and vRAM but the future researcher should not limit themselves to this. They can modify or use another Deployment Flavor. DF can be based on the metrics, client's needs, throughput, bandwidth, memory, and types of VNFs.

- **Green energy trade-off**

Green energy is an essential and well-known topic at present. In Chapter 5, we consider the energy aspect of VNF and show how it impacts the Total Cost Of Ownership (TCO) and its relation with VNF resources and licenses. We have initiated considering VNF energy to establish its link with resources and license in the context of vEPC, but there is still much room for exploration. We only consider the impact of resources and license with energy, and other parameters can be considered. Even the scenario may be changed; it can be slicing, SDN, or others. Thus, future researchers should consider different metrics and scenarios along with energy.

- **Machine learning and Artificial intelligence with flavor of VNFs**

The present world will not be complete without mentioning ML and AI. In our study, we did not use ML or AI because there is a lack of datasets in the context of the metrics we are considering: SAU and BW. We face a shortage of datasets because as much as NFV has progressed in academics, it is ironic that it did not progress much in the industries. Most projects are in the pilot phase, and some are just beginning. Nevertheless, we believe future researchers will have adequate datasets in domains like NFV, SDN, and other virtualized software. They can use these datasets and use it for future prediction. One of the possible approaches we can suggest now is considering DF and enterprise. The future researcher can use ML to classify different kinds of enterprises based on DF and use reinforcement learning to provide feedback and suggestions for the future to their clients.

Thus, many doors are open for the future researcher. Throughout this research, we observe the importance of software cost, license, and resources of the VNF. Licensing and these costs are not considered seriously. We show they are as engineering problems as others (traffic, load balancing, etc.). Technologies are shifting rapidly from hardware to softwarization and cloudification. Thus, we believe this is the correct time to initiate the thinking about licensing and software costs of virtual technologies. If we do not commence this attitude toward software cost, then we will not be able to yield the benefits promised by NFV, SDN, and other virtual technologies.

# Bibliography

- [1] N. ETSI, “Network functions virtualisation NFV release 3; licensing management; report on license management for nfv,” 2017. [https://www.etsi.org/deliver/etsi\\_gr/NFV-EVE/001\\_099/010/03.01.01\\_60/gr\\_nfv-eve010v030101p.pdf](https://www.etsi.org/deliver/etsi_gr/NFV-EVE/001_099/010/03.01.01_60/gr_nfv-eve010v030101p.pdf).
- [2] N. F. V. ETSI, “Network functions virtualisation NFV,” *Management and Orchestration*, vol. 1, p. V1, 2014. [https://www.etsi.org/deliver/etsi\\_gs/nfv/001\\_099/002/01.02.01\\_60/g\\_s\\_nfv002v010201p.pdf](https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.02.01_60/g_s_nfv002v010201p.pdf).
- [3] N. F. V. NFV, “Etsi gs NFV 001 v1. 1.1 (2013-10),” 2013. [https://www.etsi.org/deliver/etsi\\_gs/nfv/001\\_099/001/01.01.01\\_60/g\\_s\\_nfv001v010101p.pdf](https://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/g_s_nfv001v010101p.pdf).
- [4] 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)*, pp. 1–9, IEEE, 2017.
- [5] “NFV FAQ,” 2020. [https://nfvwiki.etsi.org/index.php?title=NFV\\_FAQ](https://nfvwiki.etsi.org/index.php?title=NFV_FAQ).
- [6] C. Jones, *Estimating software costs: Bringing realism to estimating*. McGraw-Hill Education, 2007.
- [7] M. Cossio, “Use total cost of ownership to optimize costs and increase savings.” <https://www.gartner.com/doc/reprints?id=1-2ADE1KAZ&ct=220622&st=sb#10>, 2022.
- [8] M. Christiano, “A brief history of software license management,” Jan. 2018. <http://reprisesoftware.com/blog/2007/01/a-brief-history-of-software-license-management-2>.
- [9] N. F. Holsing and D. Yen, “Software asset management: Analysis, development and implementation,” *Information Resources Management Journal (IRMJ)*, vol. 12, no. 3, pp. 14–26, 1999.

- [10] D. Clarke, “NFV license management: The missing piece of the puzzle,” May 2022. <https://www.cablelabs.com/blog/nfv-license-management-the-missing-piece-of-the-puzzle>.
- [11] A.-L. Vion, *Software asset management and cloud computing*. PhD thesis, Université Grenoble Alpes, 2018.
- [12] 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)*, pp. 293–297, IEEE, 2018.
- [13] 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)*, pp. 107–113, IEEE, 2016.
- [14] P. Vizarrreta, 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)*, pp. 1–7, IEEE, 2017.
- [15] 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.
- [16] 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.
- [17] 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)*, pp. 1–6, IEEE, 2019.
- [18] 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.
- [19] 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)*, pp. 5290–5295, IEEE, 2015.
- [20] F. Magnanini, L. Ferretti, and M. Colajanni, “Efficient license management based on smart contracts between software vendors and service providers,” in *2019 IEEE 18th In-*

- ternational Symposium on Network Computing and Applications (NCA)*, pp. 1–6, IEEE, 2019.
- [21] 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)*, pp. 226–233, IEEE, 2016.
- [22] Cisco, “How to select the right network function virtualization solution,” 2019. <https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/network-functions-virtualization-nfv-infrastructure/maximize-profit-potential-with-nfvi.pdf>.
- [23] A. Chevalier, *Optimization of software license placement in the Cloud for economical and efficient deployment*. PhD thesis, Université de Lyon, 2020.
- [24] A. Mouaci, É. Gourdin, I. Ljubić, and N. Perrot, “Virtual network functions placement and routing problem: Path formulation,” in *2020 IFIP Networking Conference (Networking)*, pp. 55–63, IEEE, 2020.
- [25] 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)*, pp. 1–6, IEEE, 2020.
- [26] Y. Liu, J. Pei, P. Hong, and D. Li, “Cost-efficient virtual network function placement and traffic steering,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2019.
- [27] C. Pham, N. H. Tran, S. Ren, W. Saad, and C. S. Hong, “Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach,” *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 172–185, 2017.
- [28] R. Shi, J. Zhang, W. Chu, Q. Bao, X. Jin, C. Gong, Q. Zhu, C. Yu, and S. Rosenberg, “Mdp and machine learning-based cost-optimization of dynamic resource allocation for network function virtualization,” in *2015 IEEE International Conference on Services Computing*, pp. 65–73, IEEE, 2015.
- [29] 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)*, pp. 1–7, IEEE, 2017.
- [30] 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 IN-*

- FOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, pp. 886–891, IEEE, 2019.
- [31] S. Agarwal, F. Malandrino, C. F. Chiasserini, and S. De, “VNF placement and resource allocation for the support of vertical services in 5g networks,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 433–446, 2019.
- [32] 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.
- [33] 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.
- [34] 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)*, pp. 1–9, IEEE, 2021.
- [35] 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.
- [36] 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)*, pp. 579–588, IEEE, 2017.
- [37] F. Carpio, S. Dhahri, and A. Jukan, “VNF placement with replication for load balancing in nfv networks,” in *2017 IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, 2017.
- [38] 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.
- [39] B. Zhang, Q. Fan, X. Zhang, Z. Fu, S. Wang, J. Li, and Q. Xiong, “A survey of vnf forwarding graph embedding in 5g/6g networks,” *Wireless Networks*, pp. 1–24, 2021.
- [40] 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)*, pp. 1–4, IEEE, 2019.

- [41] F. Carpio, S. Dhahri, and A. Jukan, “VNF placement with replication for load balancing in nfv networks,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.
- [42] 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)*, pp. 1–7, IEEE, 2017.
- [43] 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)*, pp. 472–479, IEEE, 2019.
- [44] 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)*, pp. 886–891, IEEE, 2019.
- [45] A. Leivadreas, G. Kesidis, M. Ibnkahla, and I. Lambadaris, “VNF placement optimization at the edge and cloud,” *Future Internet*, vol. 11, no. 3, p. 69, 2019.
- [46] 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*, pp. 1–8, IEEE, 2009.
- [47] 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*, pp. 8–13, IEEE, 2013.
- [48] C. C. T. Mark, D. Niyato, and T. Chen-Khong, “Evolutionary optimal virtual machine placement and demand forecaster for cloud computing,” in *2011 IEEE International Conference on Advanced Information Networking and Applications*, pp. 348–355, IEEE, 2011.
- [49] ETSI, “Environmental engineering (EE), measurement method for energy efficiency of network functions virtualisation NFV in laboratory environment.”
- [50] 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*, pp. 2449–2457, IEEE, 2019.
- [51] A. R. Gibbons, “New savings with engineering software,” 2020. <https://www.itassetmanagement.net/2020/10/27/new-savings-with-engineering-software/>.

# Glossary

---

<b>NFV</b> Network Function Virtualization . . . . .	1
<b>VNF</b> Virtual Network Function . . . . .	3
<b>SDN</b> Software Define Networking . . . . .	1
<b>ETSI</b> European Telecommunication Standards Institute . . . . .	1
<b>CAPEX</b> Capital Expenditures . . . . .	1
<b>OPEX</b> Operational Expenditures . . . . .	1
<b>IDS</b> Intrusion Detection System . . . . .	2
<b>NAT</b> Network Address Translation . . . . .	2
<b>QoS</b> Quality Of Services . . . . .	4
<b>MANO</b> Management and Orchestration . . . . .	2
<b>NFVI</b> NFV Infrastructure . . . . .	2

<b>IPR</b> Intellectual Property Right . . . . .	3
<b>CSP</b> Communication Service Providers . . . . .	3
<b>LC</b> License Cost . . . . .	3
<b>RC</b> Resource Cost . . . . .	3
<b>SC</b> Scientific Contributions . . . . .	5
<b>EC</b> Energy Cost . . . . .	6
<b>NFs</b> Network Functions . . . . .	8
<b>COTS</b> Commercial-Off- The Shelf . . . . .	9
<b>NFVO</b> NFV Orchestrtrion . . . . .	11
<b>VNFC</b> VNF Components . . . . .	11
<b>DF</b> Deployment Flavor . . . . .	11
<b>vCPU</b> Virtual CPU . . . . .	11
<b>vRAM</b> Virtual RAM . . . . .	11
<b>VLD</b> Virtual Links Descriptor . . . . .	11
<b>SAM</b> Software Assets Management . . . . .	3

<b>ROI</b> Return On Investment . . . . .	16
<b>PAYG</b> Pay As You Grow . . . . .	21
<b>OSS</b> Operation Support System . . . . .	21
<b>BSS</b> Business Support System . . . . .	21
<b>LM</b> License Management Entity . . . . .	23
<b>POPs</b> Points of Present . . . . .	23
<b>VNF-SC</b> Virtualize Network Function Software Cost . . . . .	27
<b>NSV</b> Network Software Vendor . . . . .	30
<b>SAU</b> Simultaneous Active Users . . . . .	32
<b>SW</b> Software . . . . .	39
<b>BW</b> Bandwidth . . . . .	45
<b>LR</b> License Reference . . . . .	43
<b>RKU</b> Resources Known Users . . . . .	46
<b>RUU</b> Resources Unknown Users . . . . .	47
<b>ICA</b> Intersection Collision Avoidance . . . . .	53

<b>RFP</b> Request For Proposal . . . . .	53
<b>vEPC</b> Virtual Evolve Packets . . . . .	53
<b>SGW</b> Service Gateway . . . . .	53
<b>PGW</b> Packet Gateway . . . . .	53
<b>MME</b> Mobile Management Entity . . . . .	53
<b>AMF</b> Access and Mobility Management Function . . . . .	62
<b>C-RAN</b> Cloud-RAN . . . . .	62
<b>DHCP</b> Dynamic Host Configuration Protocol . . . . .	62
<b>DNS</b> Domain Name Service . . . . .	62
<b>IDA</b> Intrusion Detection Algorithm . . . . .	62
<b>VM</b> Virtual Machines . . . . .	65
<b>VN</b> Virtual Network . . . . .	65
<b>EER</b> Energy Efficiency Ratio . . . . .	67
<b>EE</b> Energy Efficiency . . . . .	67
<b>LMS</b> License Method for Share . . . . .	69

**LRMS** License and Resource Methods For Sharing . . . . . 70

**CLM** Classic Method . . . . . 71

**vStorage** Virtual Storage . . . . . 71

**emBB** Enhanced Mobile Broadband . . . . . 76

**MMTC** Massive Machine Type Communications . . . . . 76

**URLLC** Ultra Reliable Low Latency Communications . . . . . 76

**PCF** Policy Control Function . . . . . 76

**UDM** Unified Data Management . . . . . 76

**AUSF** Authentication Server Function . . . . . 62

**AMF** Access and Mobility Management Function . . . . . 62

**SMF** Session Management Function . . . . . 76

# List of Figures

1.1	High level NFV framework . . . . .	2
1.2	Chapters Arrangement . . . . .	6
2.1	Network Function Virtualization . . . . .	9
2.2	NFV Reference Architecture Framework . . . . .	10
2.3	VNF Descriptor . . . . .	12
2.4	Traditional to NFV . . . . .	12
2.5	SDN and NFV . . . . .	13
2.6	Software Cost Classification . . . . .	14
2.7	License Cost and its hidden impacts. . . . .	16
2.8	OPEX and CAPEX General Classification . . . . .	17
2.9	License Management Actors and Different Roles. . . . .	19
2.10	Overall classification of software license . . . . .	20
2.11	Information flow between NFV- MANO and License Manager. . . . .	23
2.12	Impacts in SAM due Cloudification and Softwarization. . . . .	25
3.1	Taxonomy of research in VNF . . . . .	31
3.2	SAU Example on the vEPC . . . . .	32
3.3	Implementation of different VNF using different VNF provider Case I. . . . .	33
3.4	Different types of VNF and compute using third party in Cisco . . . . .	34
3.5	VNF implementation using different flavor. . . . .	35
4.1	Taxonomy of Users Scenario . . . . .	48
4.2	Daily License Reference (LR1, LR) . . . . .	51
4.3	Daily license cost for both license reference (LR, LR1) . . . . .	51
4.4	Daily resource cost for both license reference (LR, LR1) . . . . .	52
4.5	Daily summation of RC and LC (Daily Total cost) for both license reference (LR, LR1) . . . . .	52
4.6	Total cost for both license reference (LR, LR1) . . . . .	53
4.7	VNF graph of the ICA service. . . . .	54
4.8	License Reference for SAU . . . . .	55

4.9	Cumulative License cost using SAU . . . . .	56
4.10	Cumulative Resource cost using SAU . . . . .	56
4.11	Cumulative Total cost using SAU . . . . .	57
4.12	License Reference for BW . . . . .	57
4.13	Cumulative License cost using BW . . . . .	58
4.14	Cumulative Resource cost using BW . . . . .	58
4.15	Cumulative Total cost using BW . . . . .	59
5.1	Latency (QoS) of a VNF . . . . .	66
5.2	Traffic evolution in stadium with amount of users . . . . .	75
5.3	Mobile data core . . . . .	78
5.4	5G Network Architecture. . . . .	78
5.5	License Cost Using Different Methods . . . . .	79
5.6	Resource Cost Using Different Methods . . . . .	79
5.7	Energy Cost Using Different Methods . . . . .	80
5.8	TCO Using Different Methods . . . . .	80
5.9	Energy Efficiency(SAU/watt) Using different methods . . . . .	81
5.10	Latency (ms) Using Different Methods . . . . .	81

# List of Tables

3.1	Cost of VNF function according to different VNF providers. . . . .	34
3.2	Flavor classification. . . . .	35
3.3	Relation of flavor with SAU. . . . .	35
3.4	Relation of flavor with SAU. . . . .	36
3.5	List of cost with SAU and flavour. . . . .	36
4.1	Key Notation . . . . .	44
4.2	BW Flavor Table for Scenario 1 (VNF instances) . . . . .	45
4.3	SAU Flavor Table for Scenario 1 (VNF instances) . . . . .	45
4.4	Service chain of different client usages with BW . . . . .	46
4.5	SAU Flavor Table for Scenario 2 . . . . .	48
4.6	BW Flavor Table for Scenario 2 . . . . .	49
4.7	Simulation parameters for a single node . . . . .	50
4.8	Simulation parameter for SAU. . . . .	54
4.9	Simulation parameter for BW. . . . .	55
5.1	DF table . . . . .	73
5.2	Key Notation . . . . .	74
5.3	Different VNF providers . . . . .	74
5.4	VNF cost based on different methods . . . . .	75
5.5	Simulation parameter. . . . .	77

# List of Algorithms

- 1 An algorithm for RMS . . . . . 70
- 2 An algorithm for LMS . . . . . 71
- 3 An algorithm for LRMS . . . . . 72
- 4 An algorithm with Classic . . . . . 73

