Scaling and Dynamic Resource Reallocation in NFV: Challenges and Research Perspectives

Review Paper

Tung Thanh Hoang
Electric Power University, Faculty of Information Technology
Hanoi, Vietnam
tunght@epu.edu.vn

Linh Manh Pham*
VNU University of Engineering and Technology,
Faculty of Information Technology, Department of Network and Computer Communications
Cau Giay, Hanoi, Vietnam
linhmp@vnu.edu.vn

Hoai Son Nguyen
VNU University of Engineering and Technology,
Faculty of Information Technology, Department of Network and Computer Communications
Cau Giay, Hanoi, Vietnam
sonnh@vnu.edu.vn

*Corresponding author

Abstract *– Network Function Virtualization (NFV) has brought incredible experiences for Internet users and network operators. NFV enables the implementation of Virtualized Network Functions (VNFs) as software running in High Volume Servers (HVSs) to execute a Service Function Chain (SFC) to satisfy service demands of Internet users. During the execution of SFCs, VNFs and Virtual Links (VLs) tend to change their resource requirements due to the dynamic nature of the end user's demands. In this paper, we focus on dynamic resource allocation to the elements of SFC throughout the SFC process to adapt to the elasticity in demand from users by providing an overall picture of NFV and the scaling problem of SFC. We then review and analyze related studies on dynamic resource allocation of NFV systems during SFC operation and analyze the results of these projects. The most recent works are also classified based on several criteria to highlight their approaches, achievements, and also shortcomings. Finally, we introduce some research directions to deal with the scaling problem during SFC operation that needs more attention from researchers to inspire future work in the elastic operation of NFV-enabled systems.*

Keywords*: elasticity, network function virtualization, optimization, resource reallocation, scaling*

Received: May 25, 2024; Received in revised form: September 11, 2024; Accepted: September 11, 2024

1. INTRODUCTION

1.1. MOTIVATION

The Internet has achieved incredible development in recent years, the number of Internet users is constantly increasing to reach 5.3 billion users (approximately 66% of the global population) by 2023 [1]. Therefore, the network infrastructures are continuously improved to meet the increasing needs of users.

Traditional network systems are mainly built from dedicated hardware devices such as routers, load balancers, firewalls, etc. The need to continuously upgrade infrastructure to satisfy the demand of Internet users creates pressure on Capital Expenditure (CAPEX), such as buying equipment, space for placing devices, and Operation Expenditure (OPEX), such as electricity bills or labor expenses for Network Service Providers. Additionally, the operating of physical appliances is also a waste of physical resources, while this approach only enables a single user to use a device at a time instead of sharing resources to leverage idle resources for others. Network Function Virtualization was developed to overcome the limitations of traditional networks.

NFV was first introduced by the European Telecommunications Standards Institute (ETSI) [2], decoupling network functions (e.g., firewall, Intrusion Detection System (IDS), Network Address Translation (NAT), load

balancer, etc.) from their dedicated hardware by deploying these network functions as software on High Volume Servers (Fig. 1) and then providing them to tenants. These network functions are now called Virtual Network Functions.

The release of NFV comes with several benefits, including: *i) Reducing resource wastage*: The nature of NFV is virtualization. By virtualizing physical resources (i.e., compute, storage, network), controllers in NFV systems can flexibly allocate and reallocate resources provided to VNFs. As a result, idle physical resources should always be utilized; *ii) Elasticity*: User demands are dynamic and may cause variations in system resource consumption. Because the resource allocation in NFV is flexible, as mentioned above, changes in service requests from users can be easily satisfied by granting more or releasing resources to these service requests; *iii) Minimizing CAPEX and OPEX*: Using NFV, service providers can reduce investments when buying Commercial Off-the-Shelf (COST) servers instead of spending money on high-cost dedicated hardware to deploy their system. Next, most NFV platforms are open-source, which means they are free. Additionally, automation mechanisms in NFV also reduce human operational activities; *iv) Fast error remediation*: Because elements in NFV systems are 'soft', administrators can quickly fix errors when unexpected things occur. Furthermore, the system can be easily re-implemented in the worst cases.

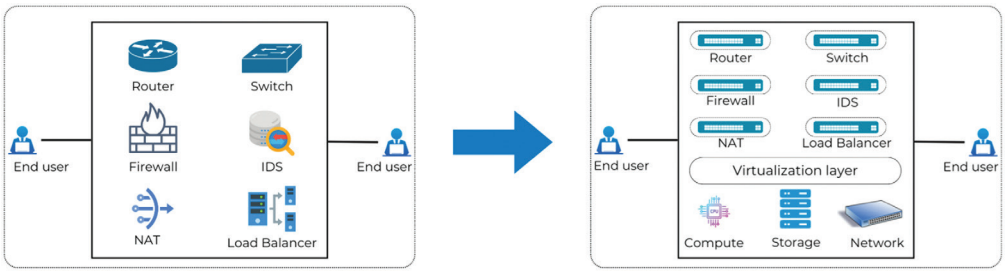

Fig. 1. The concept of virtualizing dedicated network devices to be software

1.2. RELATED WORK

As part of activities on the Internet, Internet users send data from and to the equipment. Traffic flows between these devices may need to pass through several network functions. In NFV-enabled systems, the service provider assigns VNFs to traffic flows to complete network services as expected by users. A combination of VNFs and possibly Physical Network Functions (PNFs) in a specific order can create a so-called Service Function Chain [3] or Network Service (NS) to satisfy the user's demand, as illustrated in Fig. 2. In this article, 'service function chain' and 'network service' are used equivalently.

Recent surveys focused on providing an overall picture of NFV. In the survey of Bo et al., the authors explained NFV concepts, terminologies, and architecture of NFV and introduced some projects that address hot topics in NFV such as VNF placement, scheduling, migration, chaining, and multicast [4]. Herrera et al. introduced a complete survey of the resource allocation in NFV within three stages: VNF Chain Composition, Embedding and Scheduling [5]. Yang et al. explored challenges and opportunities and offered some potential research directions in security issues for NFV [6]. The paper of Yanghao et al. [7] presented the variants of the resource allocation problem in NFV and provided a basic and standard mathematical model for the resource allocation problem for SFCs. The authors also offered some prominent research trends.

Other surveys [8], [9], [10], [11] placed emphasis on the placement of constituents (e.g., VNFs, CNFs (Container Network Functions), VMs (Virtual Machines), etc.) of NSs at the initialization of SFC. In which the authors attempted to explore solutions to answer the question: "What is the best strategy for the placement problem to get the highest system performance with the lowest cost (in terms of minimizing the volume of resources that servers and links provide to SFC's elements)?".

1.3. OUR CONTRIBUTIONS AND PAPER ORGANIZATION

In recent years, along with the elasticity in cloud computing, the issue of flexibility in resource allocation for the operation of SFCs has also been the subject of concern. Especially, in the context of SFCs, it is always necessary to adjust to changes in tenant requirements. However, whereas most surveys are paying attention to resource allocation at the SFC's initialization, there is no overall picture of resource reallocation during the SFC's operation due to the dynamicity of requests from users, although this problem has become a hot trend in the last several years, as we will point out in sub-section 2.3.

In this paper, we focus on studying aspects related to dynamic resource reallocation to elements of SFCs during the operation of SFCs to meet the flexibility of end-user needs.

The main contributions of this survey are summarized as follows:

- An overview of NFV-enabled systems is presented, as well as a clarification of the scale issue encountered when operating virtual functions.

- We review and analyze the most recent work in dealing with the SFC scaling problem based on four aspects: The problem that the project tries to solve, the proposed solutions of the authors, the measured parameters in the research, and the experimental results. We also point out several deficiencies of existing research.
- Finally, according to the analysis results, we summarize existing solutions using a comparison table. Since this comparison, we offer and explain in detail some potential research directions in this field.

The rest of this paper is organized as follows: We analyze the background of NFV in Section 2, including NFV architecture and the scaling problem in the operation of SFCs, and then we briefly describe most recent works in the field. Previous efforts are summarized based on specific criteria and recommend several promising avenues in the field in Section 3. Section 4 concludes our work.

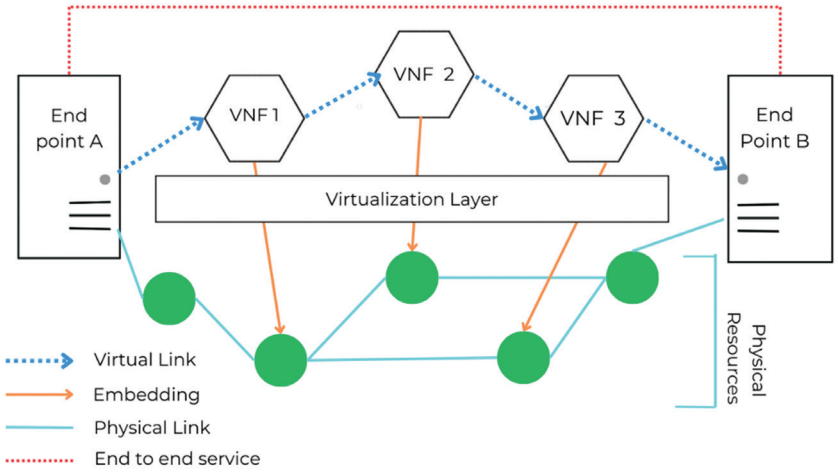

Fig. 2. An example of an SFC [12]

2. BACKGROUND AND RELATED WORK

2.1. NFV ARCHITECTURE

Network Function Virtualization is a network architecture where network functions are managed and deployed on hosts (physical computers, virtual machines, or containers) instead of traditional dedicated physical devices. Virtualization technology plays a key role in NFV, in which physical resources (e.g., compute, storage, network) are abstracted by hypervisors before allocated to upper layers. Figure 3 depicts the NFV architecture model consisting of the components as follows:

The NFV Infrastructure (NFVI) block consists of:

The *hardware and software infrastructure* that provides the platform for deploying VNFs. They are: i) servers that provide compute or storage capabilities [13], these servers can be either physical or virtual; ii) network facilities, including connection devices, transmission media, and network cards.

The virtualization layer is right above the hardware resources for the purpose of abstracting the underlying physical resources to create virtual resources. Therefore, this layer is also called a hypervisor. Currently, there are several well-known hypervisors on the market, such as KVM, Microsoft Hyper-V, ESXi, Xen, etc.

Virtualization infrastructure is virtualized resources that are abstracted by the hypervisors. It includes virtual compute (i.e., CPU), virtual storage (i.e., RAM), and virtual network (i.e., bandwidth). These resources mainly constitute a virtualization environment to implement VNFs.

VNF layer: This layer plays a vital role in the NFV system. In this layer, network functions are deployed on virtualized resource platforms as software. VNFs perform network functions such as NAT, firewall, load balancing, etc., replacing traditional physical devices in the network. Each VNF may consist of one or several VNF Components (VNFC), which are orchestrated by the corresponding Element Management (EM). EM collects information about the operation of VNFs to provide to the VNF manager (VNFM). The set of EMs will make up an Element Management System (EMS). The market for VNFs is highly tremendous, including some notable names such as Suricata for IDS, HAproxy for load balancers, Open vSwitch for switches, etc.

The Management and Orchestration (MANO) block is a constituent of three sub-blocks:

Virtualization Infrastructure Manager (VIM): VIM manages and coordinates the virtualized resources of the system.

VNF Manager (VNFM) is responsible for managing VNFs, including: i) VNF Lifecycle Management (LCM); ii) VNF configuration management of the configuration parameters of a VNF/VNFC; iii) VNF information management for the value changes of VNF-related indicators; iv) VNF Performance Management (PM); v) VNF Fault Management (FM). VNFM can be deployed to manage a single VNF or a group of VNFs.

NFV Orchestrator (NFVO) in-charges of: i) handling the lifecycle management of NSs and their constituents; ii) NS performance measurements and NS fault management; iii) onboarding and management of Network Service Descriptors (NSDs) (detailed in Section 2.3); iv) onboarding and management of PNF Descriptor archives; v) onboarding and management of VNF Packages; vi) management of software images.

Operation Support System/Business Support System (OSS/BSS) is a system that supports the operation of the NFV system by interacting with operators and customers.

2.2. SCALING IN NFV

To complete a service request from a client, VNFs are logically connected to form an SFC. For example, traffic in a video conference session may need several network functions such as load balancing, video encoding, HTTP services, etc. Fig. 2 depicts an example of an SFC. In this example, traffic from end point A to end point B must be handled by three network functions, VNF 1, VNF 2, and VNF 3. A combination of three VNFs in this order forms an SFC.

To implement an SFC, the controller needs to determine a forwarding graph, which is called VNF Forwarding Graph (VNFFG) based on the physical forwarding graph between physical nodes. The mapping of physical network forwarding graph and SFC is depicted in Fig. 4. In which, to form an SFC from user U1 to user U2 consisting of 3 VNFs in sequence: E ⟶ B ⟶ A, a virtual path will be constructed, starting from U1, pass through server 1, server 3, and server 4 in the cloud environment, before reaching U2 at the end of the path. Each server serves as a physical node for VNFs, and we must be aware that they can reside in different data centers and belong to multiple service providers. The path between the physical nodes is called a physical link, whereas the connection of ordered VNFs: E, B, and A is called a virtual link. An SFC is the constitution of VNFs and VLs.

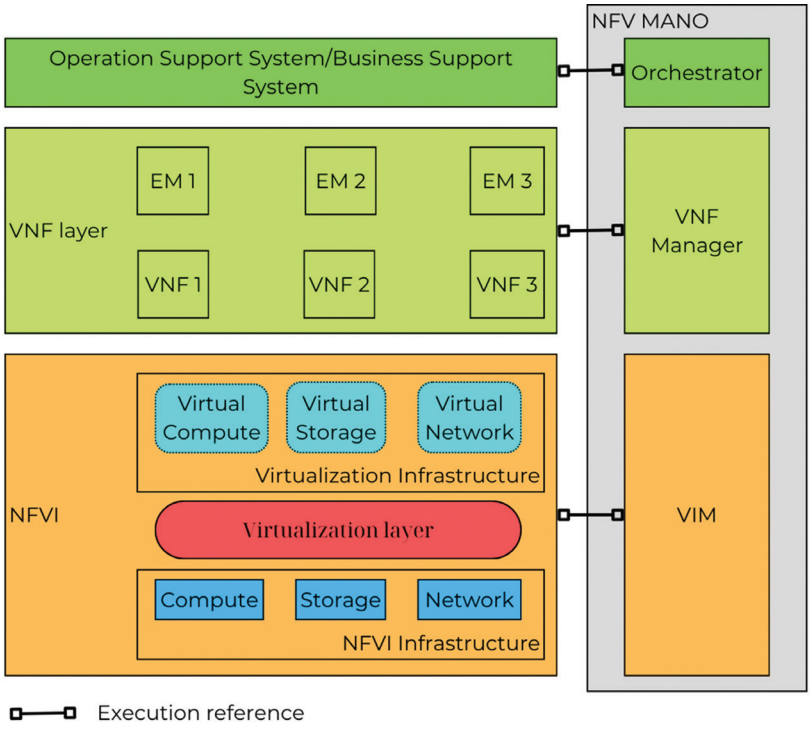

Fig. 3. NFV reference architecture [14]

Because user demands are constantly changing, tenants may have to increase or decrease their service requests by changing the volume of traffic or the quality of service while using the network service. Due to this variability, during the operation of the SFC, VNFs and VLs can be overloaded and need more resources or underloaded and need to be revoked resources to adapt to the change and to use efficiently the resources of servers and links. This leads to a phenomenon called 'scaling'.

Scaling in SFC is the term for VNFs and VLs that need to add/release resources (scaling up/down) [15] or need to add/delete the instances of VNFs (scaling out/in) [16], [17] as shown in Fig. 5. These concepts of scaling up/down (Fig. 5b) or scaling in/out (Fig. 5c) are also known as vertical scaling and horizontal scaling, respectively. For more detail, in Fig. 5b, VNF B requires more virtual resources (i.e., vCPU, vRAM) to handle larger volumes of traffic flows and the node hosting this VNF will grant more after checking its remaining capacity. Similarly, when the volume of data flow decreases, VNF B will return redundant resources to the physical node to enhance resource utilization. Horizontal scaling is depicted in Fig. 5c, which means that to deal with the increase in ingress data flow, other replicas of VNF B are deployed on other nodes and the traffic will be split

in a certain ratio to be transmitted to all instances of VNF B. Figure 5d illustrates migration. It is a term referring to the movement of VNFs from one server to another. In this situation, the current instance of VNF B is terminated and the virtual link from VNF A to VNF B will be rerouted to other instances of VNF B on other nodes.

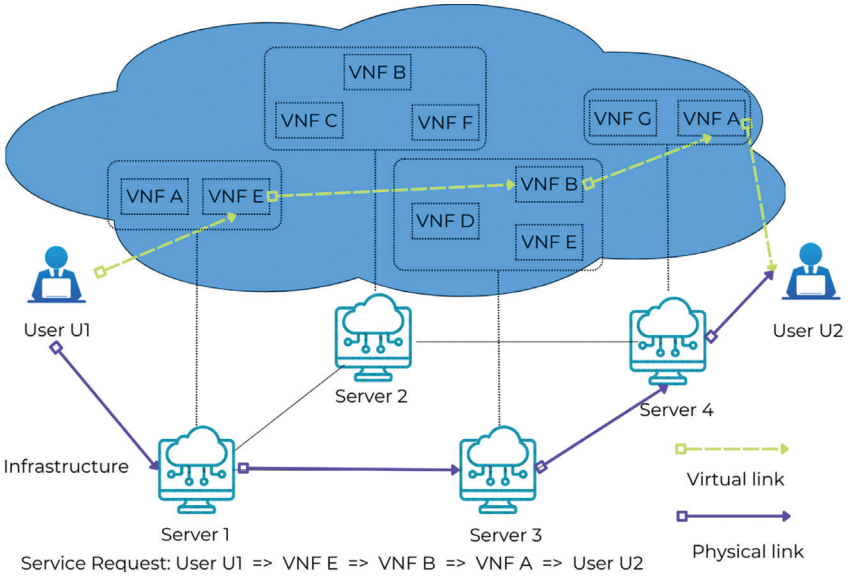

Fig. 4. A mapping of physical network and Service Function Chain [4]

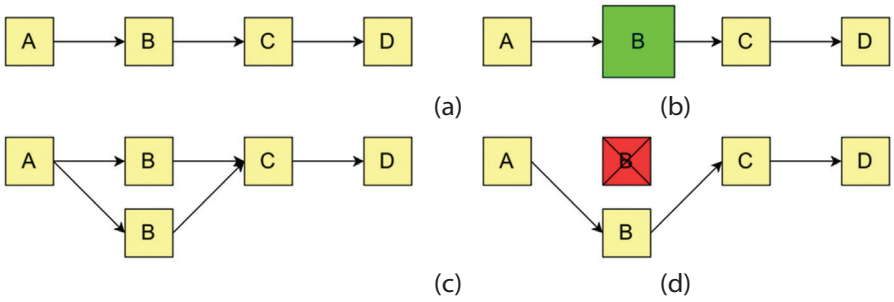

Fig. 5. VNF scaling models. (a) Service Funcion Chain, (b) Vertical Scaling, (c) Horizontal scaling, (d) Migration

To conclude, to guarantee the normal operation of NSs, controllers may need to: i) allocate more resources to VNFs participating in SFCs in case of scaling-up; ii) release resources granted to VNFs when scaling-down; iii) deploy more VNF instances on other servers and then split traffic flow with a determined proportion to pass through these instances in case of scaling-out; iv) delete VNF instances on idle servers to save resources in case of scaling-in; v) delete VNFs from resource-starving servers and deploy them on other servers that have enough capability, then re-direct the traffic flow to a new server in case of migration; and vi) no scale. Together with VNFs, VLs are also impacted. VLs that connect to and from the scaled VNF also need to be adjusted bandwidth to adapt with an increment or decrement of the VNF. Additionally, horizontal scaling and migration also occur when VLs between VNFs starve resources. In that case, VNFs related to those VLs also need to be deployed in other servers to reduce the amount of traffic across overloaded VLs.

In NFV systems, management operations take place in the MANO block. VIM is responsible for receiving virtual resources from hypervisors and granting them to VNFs, whereas VNFM performs activities to manage VNFs, including initializing and terminating VNFs. During the process of creating VNFs and constructing SFCs, the initial resource requirements of VNFs are declared in the VNF Descriptor (VNFD), as described in subsection 2.3. These initial resource indices, along with operational system metrics such as transmission latency, data traffic, etc., serve as input factors for scaling-related decisions such as triggering scaling events, requesting additional resources, etc. For example, on the OpenStack open-source platform, Tacker runs as MANO and is executed on controller nodes to deploy and coordinate NFV-related tasks such as VIM registration, VNF creation, SFC construction, and resource allocation/reallocation.

2.3. SFC SCALING SOLUTION

One of the objectives of organizations when implementing NFV-enabled networks is to optimize the resource utilization of the system. The resource allocation problem in NFV is divided into three stages during the cre-

ation of the network system [5], including: i) considering the constraint between VNFs in the SFC chain (VNF chain composition problem); ii) determining the best place for the deployment of VNFs on physical servers (VNF Embedding problem); and iii) scheduling VNFs operations (VNF Scheduling). Previous works emphasized these stages [18], [19], [20], [21], [22]. Some papers took into account the flexibility factor when allocating resources for VNFs during deployment to ensure a minimum amount of resources for the operation of VNFs [23], [24], [25].

The appropriate resource allocation solution must be determined during the system deployment and service chain initialization stages. However, resource reallocation during system operation and SFC implementation should also be considered. SFCs must be added/released resources for a variety of reasons, including failures, security concerns, and changes in user needs during operation. In case of force majeure, it may be necessary to move one or some VNFs to another server to avoid service disruptions. In recent years, many researchers have begun to pay attention to the problem of scalability of VNFs (as well as VLs) and flexible resource reallocation for the operation of the SFC.

Hui et al. proposed a model to increase the success rate of scaling by developing an algorithm called ElasticNFV based on two main ideas: i) allocating more vCPU and vMemory to VNFs when needed; ii) in case there are not enough resources to grant more, move the VNFs to other servers [26]. ElasticNFV provides a Two-Phase Minimal Migration (TPMM) algorithm to minimize the migration time and embedding cost of VNF replicas. The experimental results showed that the TPMM algorithm outperforms two previous solutions, Sandpiper [27] and Oktopus [28], in terms of migration time and cost. At the same time, in a small test bed, ElasticNFV also achieved better resource utilization than FreeFlow [17]. The proposed algorithm is fine-grained when combining vertical scaling and migration to have efficient utilization of resources, a short scaling period, and a fast response time. However, the work can be improved by leveraging horizontal scaling. In which the controller can replicate more VNF instances and then tear the traffic to pass through both the current VNF instance and its new replicas.

The article [29] offered a mechanism called ENVI (Elastic resource Flexing for Network function VIrtualization) that uses both features at the VNF level and infrastructure level to construct a machine-learning-based decision engine that can detect VNFs that need to be scaled. This mechanism continuously collects information about VNFs and their resource utilization, then it is fed to train a neural network. The evaluation shows that the neural network model outperforms other classification models such as decision tree, random forest, and logistic regression, with measures of accuracy, precision, recall, Receiver Operating Characteristic (ROC), and Area Under ROC curve (AUROC) and therefore can be a promising approach for scaling detection.

Zhao et al. [30] presented a model that considers the resource utilization of physical machines and the transmission delay of SFCs in response to the scaling out of SFCs. This model begins by continuously collecting information about the resource usage of the SFC, comparing it to a predetermined threshold, and deciding whether to scale in or out. VNFs in the scale-in list will be turned off to save resources, and the algorithm will prepare specifications for VNFs in the scale-out list to deploy them on other servers. The proposed algorithm proves that it brings about better physical machine resource utilization and transmission delay in comparison with traditional greedy algorithms. However, whereas the main idea of the algorithm is to migrate resource-starving VNFs to other nodes, scaling problems are affected by many complicated factors and can be sophisticatedly treated with other scaling models such as vertical scaling and horizontal scaling before migration.

The paper of Toosi et al. [31] dealt with techniques to solve resource utilization problems of SFCs using a resource threshold and the algorithm based on this threshold to decide whether the resources of the chain need to be augmented or reduced or not. To evaluate the performance of the solution, the authors defined two baselines: NoScale-Min represents the performance parameters of the system when the amount of resources is set as the initial value of the system, and NoScale-Max represents system parameters when providing maximum resources and assuming that Service-Level Agreement (SLA) violations never occur. What both baselines have in common is that they represent the performance statement of the system if the system is not scaled. Experimental results show that the ElasticSFC algorithm brings the SLA violation rate and the average response time of the requests close to that of NoScale-Max while saving resources by dynamically allocating resources based on workload. The migration phase is simply performed by finding the closest node to the traffic flow to deploy new VNF instances. This can be improved by implementing VNF replacement with an optimized VNF placement mechanism.

Dong et al. proposed a hybrid solution called HSM (Hybrid Scaling Method) [32]. The method allows to increase the success scaling rate of SFC by applying the IVS (Improve Vertical Scaling) algorithm to the server hosting that VNF and the virtual links connecting to that VNF when the required resource at a time exceeds the resource provided according to SLA. If this allocation fails, the IHS (Improve Horizontal Scaling) technique will be used to generate a new instance of that VNF and deploy it to another server. To lower scaling failure ratios, the IVS algorithm combines vertical scaling with traffic splitting, which supplies bandwidth for the increased bandwidth demand of virtual links by utilizing additional physical links. Experimental results showed that, compared to ElasticNFV [26] and ElasticSFC [31], the HSM method brings about superior success scaling rates with lower resources. However, in this article, the authors assumed

that the substrate system is secure without any attacks or failures of the network's elements. This may not reflect the real-world system. Additionally, the proposed horizontal scaling mechanism is an approach that is close to migration when creating new VNF instances and redirecting the traffic flow to those instances instead of replicating instances and splitting traffic to pass through both the current VNF and its new replicas. The HSM algorithm can be refined by tearing data flow to pass to new replicas of VNF before creating new VNF and rerouting the traffic flow as IHS does.

Cao et al. proposed a dynamic resource allocation mechanism that allows adjusting the operation of SFCs to guarantee the NS provision for end users in case the hardware infrastructure (node or link) fails [33]. This minimizes service interruptions in NFV-enabled vehicular networks. Nevertheless, although the project focuses on the flexible resource allocation problem, the authors did not mention the changes in resource demand for services. Therefore, they ignored vertical scaling and horizontal scaling.

In practice, network topology may change continuously over time because of adding or removing VNF instances. Eliminating changes in the network topology is a good way to reduce costs. Yifu et al. [34] proved that VNF scaling is an NP-hard problem. Then, the authors proposed an online algorithm to assist the VNF horizontal scaling problem, which includes two parts: The first part is a forecasting model based on Fourier series to mitigate frequent updates to the network topology, and the second is an algorithm to place the right VNF instance. The experimental results show that the approach can save 20% of costs while retaining performance parameters.

Rankothge et al. presented a framework based on metaheuristic Iterated Local Search (ILS) for autonomously reallocating resources to three scaling models (vertical scaling, horizontal scaling, and migration) [35]. The results of the experiment show that the proposed framework can return an optimal solution in several milliseconds, whereas Integer Linear Programming (ILP) might take some minutes to converge. The authors also explored how optimization is affected by the different scaling models and the optimization goals, then proved out that adopting only vertical scaling should be avoided and horizontal scaling is a method that trade-offs between CPU resources, system instability, and accepting more scaling requests.

The paper of Houidi et al. [36] focused on solving the VNFFG extension problem during SFC's operation. That is, tenants are likely to add more network functions or new forwarding paths into their services as demands arise and as their consumer base and profiles evolve. To maximize the number of extended requests while maintaining the stability of the original system and to avoid service disruption with a minimum execution time, the authors first addressed the problem through an ILP model, which can bring good performance indexes in reasonable problem sizes, as a baseline to evaluate proposed heuristic algorithms (e.g., a Steiner Tree-based algorithm and an Eigendecomposition based algorithm). The Steiner Tree is proven to be the best solution for the VNFFG extension problem as it archives high successful extension ratios with an acceptable execution time for large scales. The Eigendecomposition algorithm can bring smaller execution time in high connection environments.

To get close to the real world, the project [37] takes into account the concurrent operation of multiple SFCs. In which, a VNF instance can join more than one SFC simultaneously. When migration occurs, SFCs constituted by these VNF instances may be affected. With the objective of reducing end-to-end delays for all affected SFCs while guaranteeing network load balancing after migration happens, Li et al. first formalized the VNF instance migration and SFC reconfiguration problem using a mathematical model. Finally, the authors proposed a multi-stage heuristic algorithm based on optimal order to solve the problem. The heuristic algorithm has three stages: i) determining the order of VNF instances to migrate. In which, the VNF instances that less affect SFCs can be prioritized to migrate; ii) determining the candidate nodes to migrate to; iii) calculating the minimum influence requirement and making decisions on migration. The results show that the proposed algorithm can reduce the average delay of 16% to 25% for various scale networks while maintaining the balance of network load. In the same vein, the study [38] focuses on utilizing multipath routing to distribute network traffic more efficiently, thereby improving network performance and reliability. By implementing multipath routing in NFV, the authors aim to address congestion issues and optimize resource utilization across the network. The proposed solution demonstrates significant improvements in balancing the load across multiple network paths, leading to enhanced overall system performance.

While most studies cannot achieve optimization in both efficiency and scalability, Yu et al. [39] developed a hybrid technique to address vertical and horizontal scaling, with the goal of providing an optimum solution in large-scale systems. By determining use cases for a specific scaling approach, the study pointed out that the priority rules for scaling method selection can be based on the comparison from six aspects, the results are: Vertical scaling can bring more efficient *resource utilization* than horizontal scaling; the *scaling period* of vertical scaling is smaller than horizontal scaling; vertical scaling has faster *response time* than horizontal scaling; for *compatibility*, horizontal scaling has more advantages than vertical scaling because some VNFs cannot improve their performance by granting more resources; horizontal scaling has better *scalability* than vertical scaling because vertical scaling is limited to physical machine capacity; two scaling methods have similar performance in *robustness*. According to

the above comparison, the authors conclude that vertical scaling has a higher priority than horizontal scaling. The experimental results showed that the proposed approach has acceptance ratios and resource utilization better than FreeFlow [17] and ElasticNFV [26].

In large network systems, the paper [40] considered the flexibility of VNF deployment and SFC orchestration based on network conditions. Besides the dynamicity of user requests, VNFs themselves can also modify the traffic amount during their execution. To minimize resource costs while satisfying VNF dependency and traffic volume scaling, Zeng et al. proposed a heuristic approach named TAIVP (Traffic Aware and Interdependent VNF Placement) consisting of three components: i) the SFC construction component is used to construct SFC with the lowest network resource cost while ensuring the constraint of the VNF dependency; ii) the path planning function determines a shortest path from source to destination based on the A-star algorithm; iii) and the SFC embedding function places VNFs on nodes based on the order of VNFs in the SFC and the discovered shortest path. The results reveal that the TAIVP algorithm can reduce network costs by 10.2% and increase the acceptance ratio of service requests by 7.6% on average. However, there are still some limitations to the project. The authors did not consider the delay of the service requests, which is an important factor in NFV. Additionally, the heuristic algorithms cannot provide a solution that is close to the optimal one.

ETSI defined a framework, namely NSD (NS Descriptor) [41], that is integrated inside the NFV MANO block for automatic detection of resource requirement changes. The key concept is that developers will define a discrete set of Instantiation Levels (ILs) for NS (NS-ILs), which NSs can be resized to during their lifecycle. The similarity for VNF-ILs and VL-ILs are found in VNF Descriptor (VNFD) and VL Descriptor (VLD), respectively. This framework can reduce the work for scaling research when they do not need to care about how to detect scaling events but only need to focus on developing solutions to deal with them. Adamuz-Hinojosa et al. analyzed how ILs are designed in NSD [42]. The authors also figured out how the scaling requirement of NSs, VNFs, and VLs can be triggered automatically by using NS-ILs, VNF-ILs, and VL-ILs, respectively.

In QoS enhancement, guaranteeing end-to-end reliability is a crucial factor. NFV-enabled networks are vulnerable due to frequent hardware and software errors. These hazards can come from many reasons, such as server failures, broken links, software errors, cyberattacks, etc. There are a number of projects that pay attention to this issue.

In order to ameliorate system reliability when failures happen, the paper [43] introduced a novel redundancy scheme while considering the VNFFG structure to avoid over backup and the utilization reduction of the underlying resource. The key concept of the solution is to place backup VNFs on high-reliability nodes. From the simulations, the proposed mechanism can cut down the backup cost by up to 46% and keep high acceptance ratios with respect to the existing algorithms.

Liu et al. in the paper [44] proposed a Mixed Integer Linear Programming (MILP) to address the reliability-aware service chaining mapping problem and an online algorithm based on the joint protection redundancy model and backup selection scheme to improve the acceptance ratio of service requests while minimizing the consumption of physical resources. The main concept is to find an efficient mapping strategy for each SFC while maintaining constraints with two main steps for two mapping schemes: The primary scheme is the mapping of VNFs along the shortest path from ingress to egress nodes, the backup scheme is the mapping of redundant VNFs that can be used when any element in the SFC fails. The proposed novel online learning algorithm optimizes the management cost and service reliability while maintaining capacity and reliability constraints with the acceptability of delay.

Additionally, the Q-learning is adjusted to select backup VNFs in the chain. The results show that the proposed approach can significantly enhance the service request acceptance ratio while reducing resource consumption in comparison to two other backup algorithms.

To detect SFC failure in real-time, the paper [45] proposed a mechanism to jointly recover failures, prevent faults, and manage resources efficiently. In the article, the authors attempt to optimize the probability of failure in networking equipment in the case of changes in network topology. The issue is mathematically formulated as an optimization problem called the Optimal Fog-Supported Energy-Aware SFC rerouting algorithm (OFES). The proposed mechanism called Heuristic OFES (HFES) includes a near-optimal heuristic to solve the OFES problem in polynomial time by guaranteeing that the probability of fault is always less than a pre-defined threshold. The simulation results point out that the average failure probability of HFES is up to 40% higher than OFES.

In recent years, Artificial Intelligence (AI) has attracted a lot of attention from the public. Researchers have started to use machine learning algorithms to solve SFC scaling issues. Jing et al. [46] designed a Long Short-Term Memory (LSTM)-based algorithm for predicting user demands. Based on predicted results, the authors proposed a proactive method to deal with the vertical and horizontal scaling problems of VNFs. The project [15] also used an online machine learning algorithm to predict upcoming user traffic, then proactively assign a new instance of VNF and reroute the data flow with fewer resources. The research of Namjin et al. improved the Graph Neural Network (GNN) architecture and utilized a few techniques from other domains, such as image processing and natural language processing, to efficiently obtain a node representation of networking information for the VNF placement problem [47]. Therefore, the proposed method can be more effective in solving the VNF deployment problem for the scaling-in.

Table 1. Approaches and methods of existing works

Research	Approach	Scaling model	Migration	Failure
[15]	To use an online learning to proactively predict upcoming traffic demands. Then efficiently create new instances of VNFs and provide optimal route for service chain.	None	☑	☒
[17]	Splitting data flow to perform load sharing between VNF instances.	Horizontal scaling	☒	☒
[26]	To use existing Kernel-based Virtualization Machine (KVM) techniques to perform dynamic resource allocation and a TPMM algorithm for optimizing migration cost.	Vertical scaling	☑	☒
[30]	Turning off VNFs that do not use up resources and deploy VNFs that need to be scaled out on another server.	None	☑	☒
[31]	To release/grant more computing resources for VNFs, bandwidth resources for virtual links.	Both	☑	☒
[32]	Considering vertical scaling and horizontal scaling to achieve a higher success scaling rate. Split the data stream to share the load among VNF instances.	Vertical scaling	☑	☒
[33]	Reallocating the deployment location of the element (VNF or virtual link) on the faulty device.	None	☒	☑
[34]	Forecasting service request changes based on the Fourier Series to reduce the frequency of network topology changes.	Horizontal scaling	☒	☒
[35]	To use a framework based on metaheuristic Iterated Local Search (ILS) to automatically reallocate resources to three scaling models.	Both	☑	☒
[36]	Optimizing the number of extended requests with an acceptable execution time when there are changes in constituents of SFC by ILP model and two heuristic algorithms.	None	☒	☒
[37]	To use a multi-stage heuristic algorithm based on optimal order to handle migration problem with participating of a VNF in multiple SFCs simultaenously.	None	☑	☒
[39]	To determine the priority of scaling method in deadling with scaling events to have optimal solution in large scale networks.	Both	☑	☒
[40]	To use a heuristic approach to construct SFCs and place VNFs with considering the VNF can change volume of traffic flow itself.	None	☒	☒
[43]	Placing backup VNFs on high-reliability nodes.	None	☒	☑
[44]	Determining two VNFs mapping schemes for normal operation and for failure use cases.	None	☒	☑
[45]	To ensure that the fault probability is always less than a threshold.	None	☒	☑
[46]	Proposing an algorithm base on LSTM to predict user demands. Then solve VNFs vertical and horizontal scaling problem base on predicted results.	Both	☒	☒
[47]	To adapt the GNN architecture and use a few techniques to obtain a better node representation for the VNF deployment task. Therefore, proposed approach can help to solve scaling-in and out of VNFs.	Horizontal scaling	☒	☒
[48]	Defining a MILP model for the problem of resilient SFC to be able to recover from a failure.	None	☒	☑

3. DISCUSSION

3.1. SUMMARY

To provide an overview picture of dealing with the SFC scaling problem, in this section, we briefly summarize the existing studies based on the following criteria: Approach and scaling model. From the reviews in sub-section 2.3, we realize that most projects tend to ignore migration scaling, while this method has its own advantages. Therefore, we involve the problem of migrating SFC elements in this comparison for an adequate view. That is, we will examine whether the study considers the migration phase of the VNFs or not. We are also interested in failure situations. Did the research take into account the possibility of system failure? Because reliability plays a vital role in satisfying QoS, especially in the current context, where network compromises cannot be ignored [51].

In Table 1, we are involved in many articles coming from various purposes, although we want to focus on the scaling problem. This is because while reviewing current work, we realize that, aside from solving the SFC scaling issue, there are a number of studies that involve aspects that are close to the scaling. For example, the paper [36] considers the addition of VNFs into SFC instead of granting more resources or changing the embedding of VNFs. To have a deep view of guaranteeing the reliability of the NFV-enabled system, we examined the roles of the failure factor in the SFC operation, then we found that most studies treated failure-related factors at the initialization of SFC [44], [45]. The authors tried to enhance reliability at the VNF placement stage and ignored failures during the execution of SFCs. That means failures are underestimated in the SFC scaling problem. We can also see that, in terms of scaling strategies, the majority of the listed research only tackles the problem using a single scaling model.

3.2. EMERGING RESEARCH CHALLENGES

Thanks to the interest of researchers, in recent years, the issue of flexible resource reallocation for the SFC has achieved significant improvements. According to our investigation, various features of dynamic resource reallocation in NFV may need to be further exploited. This section covers potential future research directions that need to be explored in the development of NFV.

Taking advantage of the elasticity of the Cloud

Since virtualization technology, servers can be deployed as containers (e.g., Docker). In this case, applications can be deployed and destroyed in milliseconds [49].

In horizontal scaling or migration scaling, the controller needs to deploy new VNFs on other servers in the system to satisfy scaling demands. This action is close to the VNF placement problem [30], [47], while most of the existing works are trying to place VNFs on running servers, then chaining these VNFs to initialize SFC without considering the flexibility of the cloud environment. Because the time to deploy and destroy VNFs on servers can be very short, when there is an SFC that needs to be deployed or scaled, the controller will select the appropriate nodes, and then deploy the VNFs to those hosts instead of selecting existing VNFs on nodes. Simultaneously, every node that does not participate in any SFC will be turned into idle mode to save energy and maximize the remaining volume of resources. Nevertheless, in the case of multiple types of servers collaborating with each other in the NFV system, this approach has drawbacks. In this case, the physical servers may take time to boot up and initialize the VNF instances. Consequently, the total convergence time of the SFC construction process may become longer and may even create bottlenecks in the system.

Considering all scaling models simultaneously

As demonstrated in Table 1, previous studies have evaluated one or more models when scaling events occur. Nonetheless, the authors tend to deal with the problem by a single scaling method. For example, articles [17], [34], [47] only leverage horizontal scaling to deal with the increase in traffic volume. This is a coarse-grained approach that may lead to a decrease in resource utilization and successful scaling ratio. In the paper [39], Yu et al. examined three scaling models and concluded that to deal with scaling events, the priority of vertical scaling is highest, followed by migration and horizontal scaling. Therefore, involving various scaling models can bring about more fine-grained solutions and could make important contributions to resource optimization for NFV operations. However, incorporating many constraints and input factors into a single problem may cause an increase in the complexity of the algorithm, while the nature of the optimization problem is trading off objectives and dependencies such as maximizing system performances while minimizing resource consumption.

Using machine learning to solve SFC scaling

As mentioned in Section 2.3, in recent years, many researchers have paid attention to using machine learning algorithms to solve the SFC scaling problem. AI has gained many achievements in image processing, text processing, etc. In NFV, it has been adapted to solve the VNF placement problem [50], VNF forwarding graph embedding problem [22]. Deep learning and reinforcement learning algorithms also joined to handle scaling problems in SFC [15], [47], [46], [48] and got highlight results. Therefore, adjusting more advanced machine learning algorithms will be a future approach. However, the accuracy of machine learning models is determined by various factors, including the quality of the training data, the effectiveness of the algorithm utilized, and the dataset size. Whereas data flow and online behaviors are complex, diverse and unpredictable. As a result, these approaches may have a certain ratio of wrong decisions such as triggering scaling events at the inappropriate time.

Flexible resource allocation in the event of failures

As figured out in Table 1, most scaling-related studies did not take into account system failures, including software errors and infrastructure crashes, while the availability of NSs must be continuously guaranteed. Therefore, when errors occur, the system needs a mechanism to react to these failures. According to reviews in sub-section 2.3, there are a number of projects involving system failures in order to maintain reliability. However, most authors mitigate faults by implementing strategies at the SFC initialization. That means they ignore fault-related factors during SFC execution. Note that these failures can come from errors at many levels (e.g., in physical servers, in virtual servers, VNF errors, etc.) and for many reasons. Therefore, besides the approach of mitigating failures at the initializing SFC stage, considering failures when measuring SFC scaling problems can improve system reliability.

4. CONCLUSIONS

Network Function Virtualization is a promising field. NFV research has grown exponentially in recent years. In which, the problem of optimizing resources for NFV operation is the focus of attention. In this paper, we provide a picture of a narrow field in resource optimization. We cover the basics of NFV and scaling in the operation of SFCs. We also present a taxonomy of recent studies in the field of solving SFC scaling during its operation. Comparisons of existing works show that there were several inadequate aspects to consider in researching the scaling of the SFC. Finally, we offer some bright directions for the future to deal with resource reallocation in the operation of network services to adapt to the dynamic demands of users.

5. ACKNOWLEDGEMENT

This project is partially supported by the European Union's Horizon 2020 Research and Innovation Programme RISE under grant agreement no. 823759 (REMESH).

6. REFERENCES

[1] Cisco, "Cisco Annual Internet Report (2018–2023) White Paper", https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed: 2023)

[2] ETSI, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call

for action", SDN and OpenFlow World Congress, Darmstadt-Germany, 2012.

[3] J. Halpern, C. Pignataro, "Service function chaining (SFC) architecture", Technical Report RFC, 2015.

[4] Y. Bo, W. Xingwei, L. Keqin, D. K. Sajal, H. Min, "A comprehensive survey of Network Function Virtualization", Computer Networks, Vol. 133, 2018, pp. 212-262.

[5] J. G. Herrera, J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey", IEEE Transactions on Network and Service Management, Vol. 13, No. 3, 2016, pp. 518-532.

[6] W. Yang, C. Fung, "A survey on security in network functions virtualization", Proceedings of the IEEE NetSoft Conference and Workshops, Seoul, Korea, 6-10 June 2016, pp. 15-19.

[7] X. Yanghao, L. Zhixiang, W. Sheng, W. Yuxiu, "Service Function Chaining Resource Allocation: A Survey", arXiv:1608.00095, 2016.

[8] W. Attaoui, E. Sabir, H. Elbiaze, M. Guizani, "VNF and CNF Placement in 5G: Recent Advances and Future Trends", IEEE Transactions on Network and Service Management, Vol. 20, No. 4, 2023, pp. 4698-4733.

[9] A. Laghrissi, T. Taleb. "A Survey on the Placement of Virtual Resources and Virtual Network Functions", IEEE Communications Surveys & Tutorials, Vol. 21, No. 2, 2019, pp. 1409-1434.

[10] J. Sun, Y. Zhang, F. Liu, H. Wang, X. Xu, Y. Li, "A survey on the placement of virtual network functions", Journal of Network and Computer Applications, Vol. 202, 2022.

[11] X. Li, C. Qian, "A survey of network function placement", Proceedings of the 13th IEEE Annual Consumer Communications & Networking Conference, Las Vegas, NV, USA, 9-12 January 2016, pp. 948-953.

[12] ETSI GS NFV-IFA 006, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Vi-Vnfm Reference Point — Interface and Information Model Specification", 2018.

[13] ETSI GS NFV-IFA 001, "Network Functions Virtualisation (NFV); Infrastructure Overview", 2015.

[14] ETSI GS NFV-IFA 002, "Network Functions Virtualisation (NFV); Architectural Framework", 2014.

[15] X. Fei, F. Liu, H. Xu, H. Jin, "Adaptive VNF Scaling and Flow Routing with Proactive Demand Prediction", Proceedings of IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, USA, 16-19 April 2018, pp. 486-494.

[16] P. Shoumik, L. Chang, H. Sangjin, J. Keon, P. Aurojit, R. Sylvia, R. Luigi, S. Scott, "E2: A framework for nfv applications", Proceedings of the 25th Symposium on Operating Systems Principles, Monterey, CA, USA, 4-7 October 2015, pp. 121-136.

[17] R. Shriram, W. Dan, J. Hani, W. Andrew, "Split/merge: System support for elastic execution in virtual middleboxes", Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, Lombard, IL, USA, 2-5 April 2013, pp. 227–240.

[18] M. T. Beck, J. F. Botero, K. Samelin, "Resilient allocation of service Function chains", Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Networks, Palo Alto, CA, USA, 7-10 November 2016, pp. 128-133.

[19] S. Mehraghdam, M. Keller, H. Karl, "Specifying and placing chains of virtual network functions", Proceedings of the IEEE 3rd International Conference on Cloud Networking, Luxembourg, Luxembourg, 8-10 October 2014, pp. 7-13.

[20] M. Wang, B. Cheng, B. Li, J. Chen, "Service Function Chain Composition and Mapping in NFV-Enabled Networks", Proceedings of the IEEE World Congress on Services, Milan, Italy, 8-13 July 2019, pp. 331-334.

[21] M. T. Beck, J. F. Botero, "Coordinated Allocation of Service Function Chains", Proceedings of the IEEE Global Communications Conference, San Diego, CA, USA, 6-10 December 2015, pp. 1-6.

[22] P. T. A. Quang, Y. Hadjadj-Aoul, A. Outtagarts, "A Deep Reinforcement Learning Approach for VNF Forwarding Graph Embedding", IEEE Transactions on Network and Service Management, Vol. 16, No. 4, pp. 1318-1331.

[23] M. Karimzadeh-Farshbafan, V. Shah-Mansouri, D. Niyato, "A Dynamic Reliability-Aware Service

Placement for Network Function Virtualization (NFV)", IEEE Journal on Selected Areas in Communications, Vol. 38, No. 2, 2020, pp. 318-333.

[24] M. Mechtri, C. Ghribi, D. Zeghlache, "A Scalable Algorithm for the Placement of Service Function Chains", IEEE Transactions on Network and Service Management, Vol. 13, No. 3, 2016, pp. 533-546.

[25] Y. T. Woldeyohannes, A. Mohammadkhan, K. K. Ramakrishnan, Y. Jiang, "A scalable resource allocation scheme for NFV: Balancing utilization and path stretch", Proceedings of the 21st Conference on Innovation in Clouds, Internet and Networks and Workshops, Paris, France, 19-22 February 2018, pp. 1-8.

[26] H. Yu, J. Yang, C. Fung, "Fine-Grained Cloud Resource Provisioning for Virtual Network Function", IEEE Transactions on Network and Service Management, Vol. 17, No. 3, 2020, pp. 1363-1376.

[27] W. Timothy, S. Prashant, V. Arun, Y. Mazin, "Black-box and gray-box strategies for virtual machine migration", Proceedings of the 4th USENIX Symposium on Networked Systems Design & Implementation, Cambridge, MA, USA, 11-13 April 2007.

[28] B. Hitesh, C. Paolo, K. Thomas, R. Ant, "Towards predictable datacenter networks", ACM SIGCOMM, Vol. 41, No. 4, 2011.

[29] C. Lianjie, S. Puneet, F. Sonia, S. Vinay, "ENVI: Elastic resource flexing for Network function Virtualization", HotCloud'17: Proceedings of the 9th USENIX Conference on Hot Topics in Cloud Computing, Santa Clara, CA, USA, 10-11 July 2017.

[30] X. Zhao, X. Jia, Y. Hua, "An Efficient VNF Deployment Algorithm for SFC Scaling-out Based on the Proposed Scaling Management Mechanism", Proceedings of the Information Communication Technologies Conference, Nanjing, China, 29-31 May 2020, pp. 166-170.

[31] T. N. Jadel, S. Jungmin, C. Qinghua, B. Rajkumar, "ElasticSFC: Auto-Scaling Techniques for Elastic Service Function Chaining in Network Functions Virtualization-based Clouds", Journal of Systems and Software, Vol. 152, 2019, pp. 108-119.

[32] D. Zhai, X. Meng, Z. Yu, H. Hu, X. Han, "A fine-grained and dynamic scaling method for service function chains", Knowledge-Based Systems, Vol. 228, 2021.

[33] H. Cao, H. Zhao, D. X. Luo, N. Kumar, L. Yang, "Dynamic Virtual Resource Allocation Mechanism for Survivable Services in Emerging NFV-Enabled Vehicular Networks", IEEE Transactions on Intelligent Transportation Systems, Vol. 23, No. 11, 2022, pp. 22492-22504.

[34] Y. Yifu, G. Songtao, L. Pan, L. Guiyan, Z. Yue, "Forecasting Assisted VNF Scaling in NFV-Enabled Networks", Computer Networks, Vol. 168, 2019.

[35] W. Rankothge, H. Ramalhinho, J. Lobo, "On the Scaling of Virtualized Network Functions", Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management, Arlington, VA, USA, 8-12 April 2019, pp. 125-133.

[36] O. Houidi, O. Soualah, W. Louati, D. Zeghlache, "Dynamic VNF Forwarding Graph Extension Algorithms", IEEE Transactions on Network and Service Management, Vol. 17, No. 3, 2020, pp. 1389-1402.

[37] B. Li, B. Cheng, J. Chen, "A Multi-Stage Approach for Virtual Network Function Migration and Service Function Chain Reconfiguration in NFV-enabled Networks", Proceedings of the IEEE International Conference on Web Services, Beijing, China, 19-23 October 2020, pp. 207-215.

[38] T.-M. Pham, L. M. Pham, "Load balancing using multipath routing in network functions virtualization", Proceedings of the IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, Hanoi, Vietnam, 7-9 November 2016, pp. 85-90.

[39] H. Yu, J. Yang, C. Fung, R. Boutaba, Y. Zhuang, "ENSC: Multi-Resource Hybrid Scaling for Elastic Network Service Chain in Clouds", Proceedings of the IEEE 24th International Conference on Parallel and Distributed Systems, Singapore, 11-13 December 2018, pp. 34-41.

[40] Z. Zeng, Z. Xia, X. Zhang, Y. He, "SFC Design and VNF Placement Based on Traffic Volume Scaling and VNF Dependency in 5G Networks", Computer Modeling in Engineering & Sciences, Vol. 134, No. 3, 2023, pp. 1791-1814.

[41] ETSI GS NFV-IFA 014, "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Network Service Templates Specification", 2019.

[42] O. Adamuz-Hinojosa, J. Ordonez-Lucena, P. Ameigeiras, J. J. Ramos-Munoz, D. Lopez, J. Folgueira, "Automated Network Service Scaling in NFV: Concepts, Mechanisms and Scaling Workflow", IEEE Communications Magazine, Vol. 56, No. 7, 2018, pp. 162-169.

[43] W. Ding, H. Yu, S. Luo, "Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme", Proceedings of the IEEE International Conference on Communications, Paris, France, 21-25 May 2017, pp. 1-6

[44] Y. Liu, Y. Lu, W. Qiao, X. Chen, "Reliability-aware service chaining mapping in NFV-enabled networks", ETRI Journal, Vol. 41, No. 2, 2019, pp. 207-223.

[45] M. M. Tajiki, M. Shojafar, B. Akbari, S. Salsano, M. Conti, M. Singhal, "Joint failure recovery, fault prevention, and energy-efficient resource management for real-time SFC in fog-supported SDN", Computer Networks, Vol. 162, 2019, p. 106850.

[46] T. Jing, L. Z. Jia, C. Yan, W. J. Wei, Y. Peng, L. C. Hao, "Adaptive VNF Scaling Approach with Proactive Traffic Prediction in NFV-enabled Clouds", ACM TURC '21: Proceedings of the ACM Turing Award Celebration Conference, Hefei, China, 30 July - 1 August 2021, pp. 166-172.

[47] S. Namjin, H. D. Nyeong, C. Heeyoul, "Advanced Scaling Methods for VNF deployment with Reinforcement Learning", arXiv:2301.08325, 2023.

[48] P. Tuan-Minh, N. Thi-Minh, N. Xuan-Tuan-Trung, C. Hoai-Nam, S. H. Ngo, "Fast Resource Allocation for Resilient Service Coordination in an NFV-Enabled Internet-of-Things System", REV Journal on Electronics and Communications, Vol. 12, 2022.

[49] M. P. Amit, G. D. Narayan, K. Shivaraj, M. M. Mohammed, "Performance Evaluation of Docker Container and Virtual Machine", Procedia Computer Science, Vol. 171, 2020, pp. 1419-1428.

[50] O. Houidi, O. Soualah, W. Louati, D. Zeghlache, "An Enhanced Reinforcement Learning Approach for Dynamic Placement of Virtual Network Functions", Proceedings of the IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, 31 August - 3 September 2020, pp. 1-7.

[51] T. -T. -L. Nguyen, T. -M. Pham, L. M. Pham, "Efficient Redundancy Allocation for Reliable Service Function Chains in Edge Computing", Journal of Network and Systems Management, Vol. 31, No. 1, 2023.