Recent Advances of Resource Allocation in Network Function Virtualization

Song Yang, *Member, IEEE,* Fan Li, *Member, IEEE,* Stojan Trajanovski, *Member, IEEE,* Ramin Yahyapour, Xiaoming Fu, *Senior Member, IEEE*

Abstract—Network Function Virtualization (NFV) has been emerging as an appealing solution that transforms complex network functions from dedicated hardware implementations to software instances running in a virtualized environment. Due to the numerous advantages such as flexibility, efficiency, scalability, short deployment cycles and service upgrade, NFV has been widely recognized as the next-generation network service provisioning paradigm. In NFV, the requested service is implemented by a sequence of Virtual Network Functions (VNF) that can run on generic servers by leveraging the virtualization technology. These VNFs are pitched with a predefined order through which data flows traverse, and it is also known as the Service Function Chaining (SFC). In this survey, we provide an overview of recent advances of resource allocation in NFV. We generalize and analyze four representative resource allocation problems, namely, (1) the VNF Placement and Traffic Routing problem, (2) VNF Placement problem, (3) Traffic Routing problem in NFV, and (4) the VNF Redeployment and Consolidation problem. After that, we study the delay calculation models and VNF protection (availability) models in NFV resource allocation, which are two important Quality of Service (QoS) parameters. Subsequently, we classify and summarize the representative work for solving the generalized problems by considering various QoS parameters (e.g., cost, delay, reliability and energy) and different scenarios (e.g., edge cloud, online provisioning and distributed provisioning). Finally, we conclude our survey with a short discussion on the state-of-the-art and emerging topics in the related fields, and highlight areas where we expect high potential for future research.

Index Terms—Network Function Virtualization, Service Function Chaining, Resource Allocation, QoS, Placement, Routing.

1 Introduction

With the continuous emergence of new service patterns (e.g., Cloud Computing or Virtual Reality) and stringent demanding Quality of Service (QoS) to network services (e.g. High Dimensional Video), the IP traffic across the network increases exponentially. According to the forecast from Cisco [1], the global IP traffic will reach 3.3 ZB by 2021, with a Compound Annual Growth Rate (CAGR) of 24 percent since 2016. In the traditional network services provisioning paradigm, network functions (e.g., firewall or load balancer) which are also called middleboxes are usually implemented by the dedicated hardware appliances. Deploying hardware middleboxes is costly due to their high cost for design and production and also these middleboxes need to be configured and managed manually, which further increases the costs of service providers. Hence, the traditional network service paradigm fails to keep pace with satisfying the ever-increasing users' QoS requirements from the perspective of CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX), which poses a big challenge to network service providers.

- *S. Yang and F. Li are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: {S.Yang, fli}@bit.edu.cn*
- *S. Trajanovski is with Microsoft, W2 6BD London, United Kingdom. E-mail: sttrajan@microsoft.com*
- *R. Yahyapour is with Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) and Institute of Computer Science, University of Göttingen, 37077 Göttingen, Germany. E-mail: Ramin.Yahyapour@gwdg.de*
- *X. Fu is with Institute of Computer Science, University of Göttingen, 37077 Göttingen, Germany. E-mail: Fu@cs.uni-goettingen.de*

Network Function Virtualization (NFV) which is first proposed by European Telecommunications Standards Institute (ETSI) [2] in 2012 has emerged as an appealing solution, since it enables to replace dedicated hardware implementations with software instances running in a virtualized environment. In NFV, the requested service is implemented by a sequence of Virtual Network Functions (VNF) that can run on generic servers by leveraging the virtualization technology. These VNFs are pitched with a predefined order through which data flows traverse, and it is also known as the Service Function Chaining (SFC). Benefiting from virtualization technology, the SFC can be established by placing the requested VNFs on a network in a very efficient and agile manner. Moreover, one or more VNFs can be dynamically added or deleted with a very small cost and high efficiency to cope with the case when the requested SFCs have been changed. NFV allows to allocate network resources in a more scalable and elastic manner, offer a more efficient and agile management and operation mechanism for network functions and hence can lead to a significant reduction in CAPEX and OPEX for network service providers.

In line with the benefits that NFV brings, one important question is how can we efficiently allocate network resources to establish requested SFC(s). This primarily deals with the (fundamental) VNF Placement and Traffic Routing (VPTR) problem and its variants, which is to place each user's requested VNFs on networks and find routes among each adjacent VNF pair without violating the node capacity and link bandwidth. Due to the inherent features and designing principles of NFV, the VPTR problem is different from the existing e.g., service placement and routing prob-

lem [3] and virtual machine placement and routing problem [4], [5]. For instance, when the requested VNFs are placed on networks, the route should traverse each located VNF one by one with a predefined order. In addition, we should also take the QoS parameters into account in the VPTR problem. For instance, in an SFC the end-to-end delay of the packets which traverse each VNF in this chaining is an important QoS to measure the performance of the NFV. To guarantee a satisfying network service, service providers need to promise a delay-sensitive performance and service to the customers. In this sense, the service providers may get revenue loss if the promised total delay is violated. Resilience is another important QoS parameter of NFV that defines the level the provided service can survive in the face of failures. Since, once a node or a link in an SFC fails, the whole SFC cannot operate and hence the provided service has to be stopped. Needless to say, the VPTR problem and its variants become more complicated when these QoS parameters are considered. Therefore, it is important to get an insight into the resource allocation problem in NFV from different dimensions, which is the main focus of this survey.

A comprehensive survey about NFV can be found in [6], [7]. The authors in [6], [7] provide a broad view of NFV in terms of NFV history and standardization, ongoing NFV supporting projects, NFV architectures and implementations, resource allocation in NFV, and its application in other fields such as Cloud Computing or Software Defined Networking. Instead of providing a broad view in a high level about NFV as in the work [6], [7], we target a very dedicated topic to review that is the state-of-the-art work about resource allocation in NFV. Herrera and Botero [8] provide a survey about resource allocation in NFV. More specifically, Herrera and Botero [8] further divide this problem into the VNF-Chain Composition problem, the VNF-Forwarding Graph Embedding problem and the VNF-Scheduling problem. For brevity, we will not individually address the VNF-Chain Composition problem and the VNF-Forwarding Graph Embedding problem, and instead, we regard the VPTR problem as a joint problem of these two problems in this survey, as most of the existing literature do. Please refer to [8] for more details for the VNF-Chain Composition problem. Similar to [8], Mirjalily and Luo [9] also present a similar survey about resource allocation in NFV. However, [9] just gives a brief overview of it and does not expand the content in more details (e.g., model, adopted approaches, etc.). Bhamare *et al.* [10] also present a survey about resource allocation in NFV. They additionally address the aspects of various QoS parameters such as delay and energy in the VPTR problem. Xie *et al.* [11] provide a survey of the VNF placement problem and its application to different fields such as mobile networks and datacenter networks. An overview of placement of VNF in SDN is provided in [12]. Laghrissi and Taleb [13] present a survey of the Virtual Machine placement and VNF placement. Nevertheless, traffic routing as well as the other problem variants such as VNF redeployment and consolidation problem are not covered in [10], [11], [12], [13]. We have reviewed most of the recent work in the related fields. Most of these reviewed works in this survey have not been addressed in the previous survey work [8], [9], [10], [11], [12], [13], and hence this survey can be regarded as an extension and development from them. More than that, the related problem definition and analysis, its respective solution with applied technique, as well mainly adopted QoS model are also covered in this survey. We aim to provide a detailed review for summarizing the recent development and interesting breakthroughs for solving the related resource allocation problem in NFV.

The remainder of this survey is organized as follows: We start by generalizing and summarizing four representative resource allocation problems in NFV and analyze their features and complexities. Section 3 presents the QoS models regarding delay and reliability parameters in resource allocation in NFV. In particular, we propose a general availability calculation model for quantitatively calculating the availability of SFC protection. Section 4 summarizes the existing literatures classified by various QoS parameters (e.g., cost, delay, reliability, energy) and different scenarios (such as edge cloud, online provisioning, distributed provisioning). Section 5 discusses some emerging topics and points some possible directions. Finally, Section 6 concludes the survey.

2 PRELIMINARY

2.1 Basic Problem Definition and Analysis

A network is represented by $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ denotes a set of N nodes and $\mathcal{L}$ stands for a set of L links. Each link $l \in \mathcal{L}$ is associated with capacity $c(l)$. In this paper, the network is referred to a substrate network unless otherwise stated. We use R to represent the set of total requests and for a request $r(\alpha, F, \vec{w}) \in R$, α denotes the requested data rate (required bandwidth), F indicates the set of requested VNFs with orders, and $\vec{w} = [w_1, w_2, \ldots, w_m]$ is a requirement vector with m requirements (e.g., cost, delay, availability, energy, etc.). The VNF $f \in F$ on node $n \in \mathcal{N}$ requires processing time of Ψ_n^f, which means the processing delay of packets arriving at the function f on node n.

Definition 1. In a given network $\mathcal{G}(\mathcal{N}, \mathcal{L})$ and for each request $r(\alpha, F, \vec{w}) \in R$, the VNF Placement and Traffic Routing (VPTR) problem is to place its requested VNFs on $\mathcal{N}$ and find routes among each adjacent VNF pair without violating the node capacity and link bandwidth such that the requirement vector $\vec{w}$ is satisfied.

Below are the VPTR problem variants.

The VNF Placement (VNFP) problem, without considering the traffic routing (sub)problem, is defined as follows:

Definition 2. For a network $\mathcal{G}(\mathcal{N}, \mathcal{L})$ and a set of requests R, suppose that the path between any node pair in the network is known/given, and each adjacent VNF pair chooses the given path to route the traffic. For each request $r(\alpha, F, \vec{w}) \in R$, the VNF Placement (VNFP) problem is to place its requested VNFs on $\mathcal{N}$ without violating the node capacity such that $\vec{w}$ is satisfied.

The VPTR problem turns into the TRaffic Routing (TRR) problem, when the requested VNFs are already placed on the network, and is defined as follows:

Definition 3. In a given network $\mathcal{G}(\mathcal{N}, \mathcal{L})$ and for a set of requests R, suppose that the requested VNFs for each r have already been placed on network nodes. For each request $r(\alpha, F, \vec{w}) \in R$, the TRaffic Routing

(TRR) problem is to route the traffic between each VNF pair without violating the link capacity such that $\vec{w}$ is satisfied.

Another problem variant, called VNF Redeployment and Consolidation (VRC) problem, is defined as follows:

Definition 4. In a given network $\mathcal{G}(\mathcal{N}, \mathcal{L})$ where a set of existing SFCs are already deployed in the network, the VNF Redeployment and Consolidation (VRC) problem is to redeploy the already existing SFCs such that the requirement vector $\vec{w}$ is satisfied.

It is worthwhile to mention that in the Virtual Network Embedding (VNE) problem [14], a Virtual Network Request (VNR) consists of a set of required virtual nodes and virtual links. The VNE problem is to map each VNR to the substrate (physical) network such that each virtual node is placed on a different substrate node with sufficient capacity and each virtual link is allocated a path with sufficient bandwidth. Although the VNE problem shares some similarities with the VPTR problem, they have the following differences:

- In the VPTR problem, the required VNFs in an SFC have a predefined order and the traffic has to traverse each placed VNF under this order. On the contrary, there is no required order between the requested virtual nodes in the VNE problem.
- One or more required VNFs can be placed in one node in the VPTR problem, however, each requested virtual node in the VNE problem has to be mapped on different physical nodes.
- After the traffic goes through one VNF for processing, its rate may get changed (increased or decreased) in the VPTR problem. However, the traffic rate mostly keeps unchanged among different virtual node pairs in the VNE problem.

Online vs. offline Provisioning: Typically, in the offline provisioning scheme, R is given in advance, and the goal is to design the optimal solution based on R. On the contrary, in the online provisioning scheme, it is assumed that the traffic requests in R arrive at the network in an online fashion, and we have to provision the request one-by-one, since we do not know the future coming requests. Unless otherwise stated, we assume an offline provisioning scheme in this survey. In Section 4.6 we will discuss the resource allocation problem in NFV for the online provisioning scheme.

2.2 Examples

Let us use an example in Fig. 1 to describe the VPTR problem[1]. Consider a network with 5 nodes (including ingress node and egress node) in Fig. 1, and the capacity of nodes a, b and c are 8, 11 and 15, respectively. The link capacities are shown above the link. Suppose a request r asks for an SFC with an order $f_1 - f_2 - f_3 - f_4$ consisting of 4 VNFs, whose resource requirements are 8, 6, 14 and 5. We have to place these VNFs on the nodes as shown in the network without violating the node capacity. Let $\alpha = 5$ Mb/s for r, then the whole route in this SFC is ingress$\rightarrow a \rightarrow b \rightarrow c \rightarrow b \rightarrow$egress without violating each traversed link capacity. We notice that this route is not a simple path since it contains a loop. From this example, we see that the route traversing an entire SFC is not necessarily a simple path [15], which makes the VPTR problem more complex.

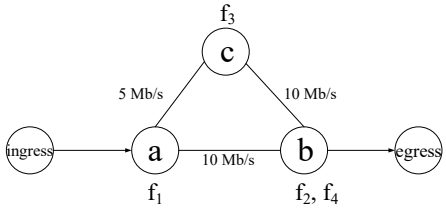
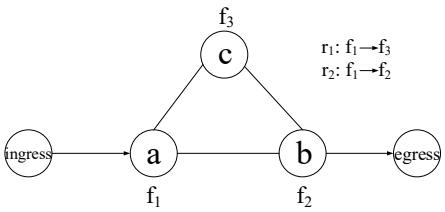
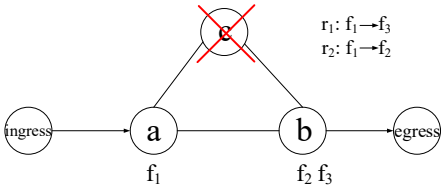


Fig. 1: An example of describing the VPTR problem.

Next, we start with an example to illustrate the VRC problem. Suppose there are 2 requests, where r_1 asks for an SFC of $f_1 - f_3$ and r_2 requests an SFC of $f_1 - f_2$. In Fig. 2(a) f_1, f_2 and f_3 are originally placed on a, b and c to serve these 2 requests. After consolidation and redeployment, f_3 is migrated from c to b and we switch off c for e.g., energy saving.



(a) Before migration and consolidation.



(b) After migration and consolidation.

Fig. 2: An example of describing the VRC problem.

2.3 Problem Goals

So far, we have formally defined the VPTR problem and its variants in Section 2.1 and described these problems by using examples in Section 2.2. However, we have not mentioned problem goals for these problems. For instance, in addition to finding a feasible VNF placement and routing solution for the VPTR problem, what is the problem goal to achieve? We could understand the problem goal as the objective function of the optimization problem. Below are frequently used problem goals:

- **Maximum link utilization (MLU)**: For a link l, its link utilization (denoted by u_l) is defined as $\frac{v(l)}{c(l)}$,

1. The ingress node and egress node represent the entry and exit of a request, respectively. However, they are usually not considered in the problem and solution. For completeness, we put them in all the illustrating figures in this survey.

where $v(l)$ denotes the actual traversing flow on l, and $c(l)$ indicates the total capacity of l. The maximum link utilization of a network is the maximum of the link utilization over all its links. The MLU is to be minimized. Sometimes, the reciprocal of the MLU is also used, which is called *network throughput*. In these cases, maximizing the network throughput is equivalent to minimizing MLU.

- **QoS parameters**: The main QoS parameters include service delay, availability, energy consumption, and so on. These parameters can quantitatively reflect how well an NFV service provides to customers.
- **Costs and Revenue**: Mostly used in the context with network operators, the network cost represents the operational expenditure or financial aspect of deploying VNFs on the nodes and routing traffic among used links. Meanwhile, revenue, in particular, refers to the net value earned by serving traffic demands, optimized under capacity constraints.

2.4 Problem Formulation

In this subsection, we formulate the (basic) VPTR problem[2] in the fully ordered SFC as an Integer Nonlinear Programming (INLP) [16] . We start by some necessary notations and variables:

R: A request set and for each request $r(\alpha, F, \vec{w}) \in R$, α denotes the requested data rate (required bandwidth), F indicates the set of requested VNFs with orders, and $\vec{w} = [w_1, w_2, \ldots, w_m]$ is a requirement vector with m requirements (e.g., cost, delay, availability, energy, etc.).

f_i and f_j: Two consecutive VNFs in the required SFC.

$X_n^{f,r}$: A boolean variable which returns 1 if the VNF f requested by r is placed on node $n \in \mathcal{N}$, and 0 otherwise.

$Y_{f_i,f_j,r}^{(u,v)}$: A boolean variable which returns 1 if the flows between VNFs f_i and f_j of r traverse link (u, v), and 0 otherwise.

Routing constraints:

$$\sum_{(u,v)\in\mathcal{L}} Y_{f_i,f_j,r}^{(u,v)} = 1 \;\; u \in \mathcal{N} : X_u^{f_i,r} = 1 \&\& X_u^{f_j,r} \neq 1, r \in R \tag{1}$$

$$\sum_{(u,v)\in\mathcal{L}} Y_{f_i,f_j,r}^{(u,v)} = 1 \;\; v \in \mathcal{N} : X_v^{f_j,r} = 1 \&\& X_v^{f_i,r} \neq 1, r \in R \tag{2}$$

$$\sum_{(u,v)\in\mathcal{L}} Y_{f_i,f_j,r}^{(u,v)} = \sum_{(v,w)\in\mathcal{L}} Y_{f_i,f_j,r}^{(v,w)} \quad v \in \mathcal{N} : X_v^{f_i,r} \neq 1 \&\& X_v^{f_j,r} \neq 1, r \in R \tag{3}$$

Placement constraint:

$$\sum_{n\in\mathcal{N}} X_n^{f,r} = 1 \;\; \forall r \in R, f \in r \tag{4}$$

Link capacity constraint:

$$\sum_{r(\alpha,F,\vec{w})\in R,(f_i,f_j)\in r} Y_{f_i,f_j,r}^{(u,v)} \cdot \alpha \leq c(u,v) \;\; \forall (u,v) \in \mathcal{L} \tag{5}$$

Node processing capacity constraint:

$$\sum_{r(\alpha,F,\vec{w})\in R,f\in r} X_n^{f,r} \cdot \eta(f) \leq \pi(n) \;\; \forall n \in \mathcal{N} \tag{6}$$

There is no objective (needed) in this formulation, but we can add the minimization of e.g., number of nodes or links, total costs, etc. which depends on specific network necessities. Eqs. (1)-(3) are the flow conservation constraints [17], [18] that account for the routing from a source to a destination and ensures to find a path from u to v. More specially, Eq. (1) ensures that if f_i is placed on node u and f_j is not placed on node u, then the sum of outgoing traffic from u is equal to 1. Eq. (2) ensures that if f_j is placed on node v and f_i is not placed on node v, then the sum of incoming traffic to v is equal to 1. Eq. (3) ensures that if f_i and f_j are not placed on node v, then the sum of incoming traffic to v is equal to its outgoing traffic. Eq. (4) ensures that each required VNF f must be placed on one node in the network. Eq. (5) ensures that the total allocated bandwidth on each link does not exceed its maximum capacity, which is denoted by $c(u, v)$. Eq. (6) ensures that the total assigned processing capacity on each node does not exceed its maximum processing capacity, where $\eta(f)$ and $\pi(n)$ represent the required resource of f and total available resource of node n, respectively.

When $X_n^{f,r}$ or $Y_{f_i,f_j,r}^{(u,v)}$ is known as an input, the above INLP accordingly changes to solve the TRR problem or VNFP problem. To solve the VRC problem, it is set in R that (some of) requests have already been provisioned by placing the required VNFs on nodes and selecting appropriate paths. On the basis of that, we can run the INLP in Eqs. (1)-(6) to solve it. That is to say, Eqs. (1)-(6) are quite general. It is also worthwhile to mention that Eqs. (1)-(6) solve the basic VPTR problem in generic networks without considering QoS requirements such as cost, delay, availability, energy. However, we can extend this formulation to solve the VPTR problem by taking into account QoS requirements in different network architectures. For brevity, please refer to [16] for the formulation of VPTR problem for delay-awareness and availability-awareness, [19] for the formulation of VPTR problem for energy-awareness. Moreover, the formulation of the VPTR problem is not unique as we state above and also a slight change of the problem input will result in different formulation. For instance, [20] assumes that a set of paths between each node pair in the network is given, then the VPTR problem formulation becomes different from Eqs. (1)-(3) since the multi-community routing constraints do not exist in this scenario. Nevertheless, since the INLP has exponential running time, it cannot return a solution in a reasonable time especially when the problem size is large, which calls for polynomial running time approximation algorithms or efficient heuristics to solve this problem.

2. On basis of the basic VPTR problem, the availability/resilience-aware VPTR problem only additionally considers the reliability constraint on basis of the basic VPTR problem. It follows similarly for the delay-aware VPTR problem, energy-aware VPTR problem, and so on.

2.5 Mainly Adopted Approaches

As we summarize in this survey, the most frequent approaches to deal with NFV resource allocation include combinatorial optimization theory (e.g., randomized/LP rounding, primal-dual approximation), Deep Reinforcement Learning, Game theory, etc. Below, we will briefly provide the representative approaches and their descriptions.

1) Randomized/LP rounding: It first formulates the problem to be solved as an integer linear program (ILP), and then computes an optimal fractional solution to the linear programming relaxation (LP) of the ILP. After that, it rounds the fractional solution of the LP to an integer solution of the ILP (original problem) within a gap to the optimal solution.
2) Primal-Dual approximation: *The primal-dual method (or primal-dual schema) is another means of solving linear programs. The basic idea of this method is to start from a feasible solution y to the dual program, then attempt to find a feasible solution x to the primal program that satisfies the complementary slackness conditions. If such an x cannot be found, it turns out that we cannot find a better y in terms of its objective value. Then, another iteration is started. An approximate solution to the primal IP and a feasible solution to the dual LP can be constructed simultaneously and improved step by step. In the end, the approximate solution can be compared with the dual feasible solution to estimate the approximation ratio* [21].
3) Markov Approximation: *It first derives log-sum-exp approximation of the optimal value of a combinatorial problem, and this leads to a solution that can be realized by time-reversible Markov chains. Certain carefully designed Markov chains among this class can yield distributed algorithms for solving the network optimization problem approximately* [22].
4) Local Search: *Local search moves from solution to solution in the space of candidate solutions (the search space) by applying local changes, until a solution deemed optimal is found or a time bound is elapsed* [23].
5) Alternating Direction Method of Multipliers (ADMM) [24]: It first separates the objective and variables into two parts, and then alternatively optimizes one set of variables that accounts for one part of the objective to iteratively reach the optimum.
6) Column Generation [25]: It begins with a small part of a problem as a restricted master problem (RMP), and then solves that part by analyzing that partial solution to discover the next part of the problem. After that, one or more variables are added to the model, and then resolves the enlarged model. This process repeats until it achieves a satisfactory solution to the whole of the problem.
7) Generalized Benders Decomposition (GBD) [26]: *the original problem is projected onto the space of first-stage variables and reformulated into a dual problem that contains an infinite number of constraints, which is then relaxed into a lower bounding problem with a finite subset of these constraints. After fixing the first-stage variables to the solution of the lower bounding problem, the original problem becomes an upper bounding problem, which can naturally be decomposed into smaller subproblems for each of the scenarios. The solutions of a sequence of upper bounding problems give a sequence of non-decreasing upper bounds on the optimal objective function value while the solutions of a sequence of lower bounding problems give a sequence of non-increasing lower bounds. An optimum of the original problem is obtained when the upper and lower bounds converge* [27].
8) Deep Reinforcement Learning (DRL): DRL is a method of solving decision-making problems, combining the idea of Reinforcement Learning (RL) and the structure of Deep Learning (DL). The essence of RL is learning through trial-and-error interactions with the environment. The RL agent first senses the state of the environment and then selects an action according to the current state. After reaching a new state, the agent receives a reward associated with the new state. The obtained reward tells the agent how good or bad the taken action was. The aim of the agent is to find the optimal policy. The policy is the strategy that the agent employs to determine the next action based on the current state so as to maximize the reward. The role of DL in DRL is to use the powerful representation capabilities of neural networks to fit the strategy of RL agent, such as to deal with the problems in a complex dynamic environment.
9) Game theory: Game theory is the study of theoretical models of strategic interaction among competing players. It is usually used to solve the problem of choosing the optimal decisions for each player in the presence of competition, cooperation, or conflict. A game model usually consists of two or more players, a set of strategies and utility functions. In general, the players which make decisions independently can be malicious, cooperative or selfish, and a player's success in making decisions depends on the choices of others. In game theory, players compete with other opponents taking turns sequentially to maximize their payoff until they achieve the Nash Equilibrium (NE). A NE is a steady state where no player has an incentive to deviate from its chosen strategy after considering the choices of other opponents.

In general, Randomized/LP rounding, Primal-Dual and Markov Approximation techniques are often used to devise the approximation algorithms. Local-search approach can aid to accelerate the speed to find local optimum solutions, but there is no guarantee to find the global optimal solution which makes it a heuristic solution. ADMM, Column generation and GBD can converge to find an optimal solution with fast speed in practice, but theoretically, they still have exponential running time. DRL provides a new perspective to solve combinatorial optimization problems (e.g., NFV resource allocation problem), but DRL has uncertain convergence time and/or long training time. Game theory provides a distributed manner to solve the resource allocation problem in NFV, but since due to the lack of global network information, it cannot essentially solve the

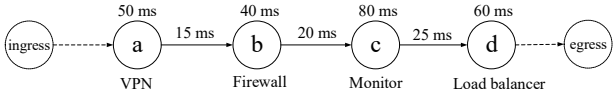
resource allocation problem in NFV in order to get the optimal solution as well as taking into account more QoS parameters such as delay and availability.

3 QoS Models in NFV

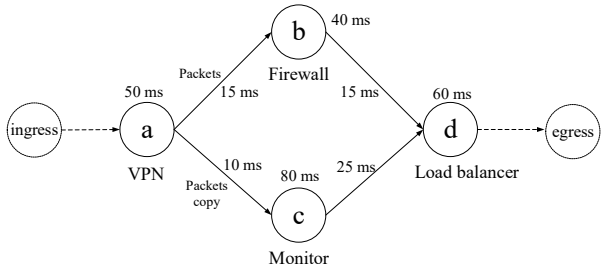
While delay and availability are two important QoS parameters in an NFV service, we study how to quantitatively calculate them in NFV in general. Moreover, we present a VNF placement availability calculation model. The other QoS models will be introduced when the respective literature is mentioned in Section 4 due to its simplicity or similarity to the introduced models in this section.

3.1 Delay Calculation of an SFC

According to the feature of function chains, there are three kinds of dependencies between VNFs, namely, (1) non-ordered: there is no dependency between VNFs, (2) totally-ordered: there is a total dependency order on the VNF set, and (3) partially-ordered: there exists dependency among a subset of VNFs. The traversing delay calculation mainly depends on the totally-ordered and partially-order scenario, since the non-ordered scenario does not occur frequently in practice. For example, Fig. 3(a) illustrates a totally ordered SFC.



(a) A totally ordered SFC.



(b) A partially ordered SFC.

Fig. 3: Totally ordered SFC v.s. partially ordered SFC, derived from [28].

We assume that an SFC contains a set of $|F|$ ordered VNFs, f_1, f_2,..., $f_{|F|}$ which need to be placed in the network. Suppose there is a total dependency order on $|F|$ VNFs, then the traversing delay in a totally ordered SFC is calculated as follows[3]:

$$\sum_{f\in F, n\in\mathcal{N}} \Psi_n^f + \sum_{l\in\mathcal{L}^s} d(l) \tag{7}$$

where Ψ_n^f represents the processing delay for function f on node n, $\mathcal{L}^s$ denotes the link set that connects the placed VNF, and $d(l)$ indicates the flow delivering delay on link l. For example, in Fig. 3(a) where the flow delivering delay and node processing delay are associated, the traversing delay is $\underbrace{(50+40+80+60)}_{\text{total node delay}} + \underbrace{(15+20+25)}_{\text{total link delay}} = 290$ ms.

However, since a monitor function only maintains the packets and does not change the packets in practice, this indicates that firewall and monitor functions are not dependent on each other. In this sense, these two functions can work in two cases, namely, (1) they work sequentially with any order between each other in an SFC, and (2) they work in parallel in an SFC. In case (1), ingress-a-c-b-d-egress is also a feasible possible feasible SFC alternative to Fig. 3(a). In case (2), firewall and monitor work in parallel as shown in Fig. 3(b). After that, firewall and monitor functions send their individual processed packets to the load balancer. As a result, the load balancer only needs to select the packets from firewall and process them. Therefore, there may exist multiple data routes traversing from the ingress node to the egress node in a partially ordered chaining. As a result, the traversing delay of a parallel SFC is equal to the longest delay route, which means that the sum of node delay and link delay in this data delivering route is the largest. For example, the traversing delay of SFC in Fig. 3(b) is equal to the route ingress$\rightarrow -c-d \rightarrow$egress, whose delay is $50+10+80+25+60=225$ ms, which is less than the one in Fig. 3(a). It is obvious to see that a partially ordered SFC can shorten the service delay compared to a totally ordered delay, but this comes at the expense of consuming more link bandwidth. Therefore, whether to use totally ordered SFC or partially ordered SFC depends on the tradeoff of network budget and application requirement.

From the above example we see that given a required non-ordered chain, there exists more than one possible SFCs (either sequential or parallel). We call the process of generating such possible SFCs as chain composition problem [8]. Moreover, Fig. 3 is also called as SFC forwarding graph [8]. In this survey, we assume that the SFC forwarding graph is given an input to the resource allocation problem in NFV, as most of existing literature do, we therefore do not address the chain composition problem and refer the readers to [8] for more details.

3.2 NFV resilience

Resilience is another important QoS parameter of NFV that defines the level the provided service can survive in the face of failures. Since, once a node or a link in an SFC fails, the whole SFC cannot operate and hence the provided service has to be stopped. An efficient way to tackle this issue is to provide redundant SFCs, and these SFCs usually place requested VNFs on different nodes (called node-disjoint) to prevent node failure and/or select paths that do not traverse same link (called link-disjoint) to overcome link failure. By doing this, the service is normally provisioned by the primary SFC, and in case of any node/link failure in the primary SFC, the backup SFC will be activated to work. This method is also known as SFC protection.

Hmaity *et al.* [29] study the SFC protection in three cases, namely, (1) end-to-end protection: a primary SFC and a backup SFC are completely (both node- and link-) disjoint;

3. We mention that it is possible more than one VNFs can be placed on one same node. In that case, we only calculate their processing time and regard that their communication delay is neglectable, leveraged by the NIC virtualization techniques (e.g., VIRTIO, DPDK, SRIOV).

(2) link-protection, and (3) node-protection. Fig. 4 reflects these three protection schemes for placing 3 VNFs in the network, where the red line indicates the primary path and the blue line represents the backup path.

However, the proposed SFC protection schemes in [29] do not consider the nodes' and links' failure probabilities. For instance, if both primary SFC and backup SFC contain nodes or links with higher failure probabilities, then this protection can also not guarantee a reliable service. Moreover, in [29] it is only considered $k = 2$ SFCs for protection (i.e., one VNF replica). However, in practice, in order to provide a more reliable service, we need to provide $k \geq 2$ SFCs sometimes. In order to deal with this issue, in [16], we present a quantitative model to calculate the VNF placement availability based on the proposed protected schemes in Fig. 4 [29]. Similar to [30], we consider both the node and link availabilities to quantitatively measure the reliability of SFC protection for any $k \geq 1$ VNF replicas/backups. More specifically, the availability of a system is the fraction of time that the system is operational during the entire service time. The availability A_j of a network component j can be calculated as [31]:

$$A_j = \frac{MTTF}{MTTF + MTTR} \tag{8}$$

where $MTTF$ represents Mean Time To Failure and $MTTR$ denotes Mean Time To Repair. A node in the network represents a server, and a link denotes a physical link connecting two physical servers. Their availabilities are equal to the product of the availabilities of all their components (e.g., hard disk and memory for a node, amplifiers and fibers for a link). For a server n, let $A(n)$ and $p(n)$ denote its availability value and failure probability, respectively, then we have $A(n) = 1 - p(n)$. Node failures can vary from server age, the number of hard disks, etc [32], and they can be caused by hardware component failure, software bugs, power loss events, etc [33]. In reality, we can obtain the server's availability value by accessing the detailed logs recording every hardware component repair/failure incident during the lifetime of the server. The details for characterizing server failures and statistically calculating node availability can be found in [32], [33] and papers therein. Reference [34] and the papers therein provide failure statistics for other network devices (e.g., switches) in datacenters.

We distinguish and analyze the VNF placement availability under two different cases, namely (1) Unprotected SFC: only one SFC is allowed to be placed in the network, and (2) Protected SFC: at most $k \geq 2$ SFCs can be placed in the network.

Suppose an unprotected SFC places VNFs on w nodes n_1, n_2, ..., n_w and traverses m links l_1, l_2, ..., l_m, its availability is calculated as:

$$\prod_{i=1}^{w} A_{n_i} \cdot \prod_{j=1}^{m} A_{l_j} \tag{9}$$

where $\prod_{i=1}^{w} A_{n_i}$ denotes the availability of all the used nodes[4] and $\prod_{j=1}^{m} A_{l_j}$ indicates the availability of all the traversed links[5].

In the protected SFC, we stress that it is composed of (maximum) k unprotected SFCs. For the ease of clarification, we further term each of the k unprotected SFC in the protected placement as placement group ρ_i. We denote by ρ_1 the primary placement group. Since different placement groups may place one or more VNFs on the same node and/or traverse the same link, we distinguish the protected placement as two cases, namely (1) *fully protected SFC*: each one of k placement groups places VNF on different nodes and traverses different links and (2) *partially protected SFC*: at least two placement groups place one or more VNFs on the same node and/or traverse the same link. In the fully protected placement case, the availability can be calculated as:

$$1 - \prod_{i=1}^{k} (1 - A_{\rho_i}) \tag{10}$$

Eq. (10) indicates that the availability of k SFCs is equal to the probability that at least one SFC can work normally (does not fail). For example, in Fig. 5a where the node and link availabilities are associated, we use s and d to represent ingress and egress nodes. For the ease of expression, their availabilities are always 1. SFC s-a-b-d and s-c-g-d are fully protected, since they do not contain any same link or node. As a result, the total availability of these two SFCs are: $1 - (1-0.9\cdot0.99\cdot0.8\cdot0.85\cdot0.95)\cdot(1-0.95\cdot0.98\cdot0.75\cdot0.99\cdot0.88) \approx 0.8338$.

Next, we consider a general scenario where at least two of k placement groups place the same VNF on the same node or traverse the same links. In this case, Eq. (10) cannot be used to calculate the availability in this scenario since the availabilities of overlapped nodes or links will be counted more than once. To amend this, we use a new operator $\circ$. Suppose there are m different nodes $n_1, n_2, \ldots, n_m$ with availabilities $A_{n_1}, A_{n_2}, \ldots, A_{n_m}$. For a node n_x with availability A_{n_x}, $\circ$ can be defined as follows:

$$A_{n_1} \cdot A_{n_2} \cdot \cdots \cdot A_{n_m} \circ A_{n_x} = \begin{cases} \prod_{i=1}^{m} A_{n_i} & \text{if } \exists n_i = n_x \\ \prod_{i=1}^{m} A_{n_i} \cdot A_{n_x} & \text{otherwise} \end{cases} \tag{11}$$

where the $\circ$ computations for link availabilities can be defined analogously.

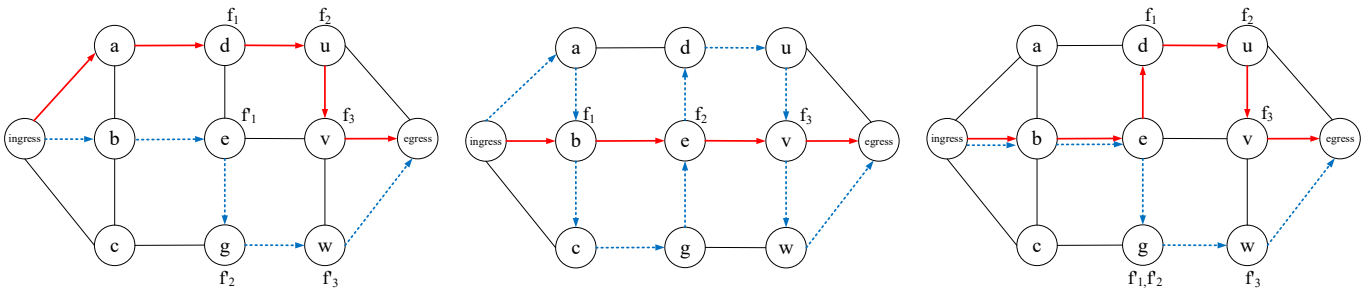
Let $\coprod$ denote consecutive $\circ$ operations of the different sets, and it is commutative, associative and distributive. Hence, the availability of k partially protected SFCs can now be represented as:

$$\begin{aligned} & 1 - \coprod_{i=1}^{k} (1 - A_{\rho_i}) \\ =& 1 - (1 - A_{\rho_1}) \circ (1 - A_{\rho_2}) \circ \cdots \circ (1 - A_{\rho_k}) \\ =& \sum_{i=1}^{k} A_{\rho_i} - \sum_{0<i<j\leq k} A_{\rho_i} \circ A_{\rho_j} + \\ & \sum_{0<i<j<u\leq k} A_{\rho_i} \circ A_{\rho_j} \circ A_{\rho_u} + \cdots + (-1)^{k-1} \coprod_{i=1}^{k} A_{\rho_i} \end{aligned} \tag{12}$$

where A_{ρ_i} denotes the availability of placement group ρ_i and can be calculated from Eq. (9). Let us take Fig. 5(b) as

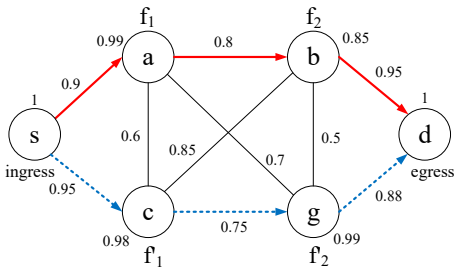
4. Even though one node can host more than one VNFs, its availability will be counted only once.

5. It is possible that one SFC traverses one link multiple times, but we consider the availability of each of the traversed links only once.

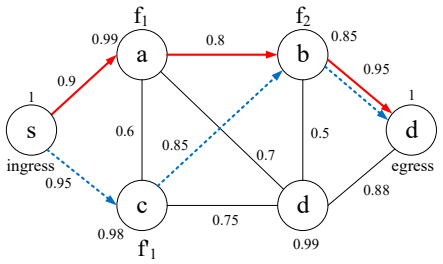


(a) End-to-end protection. (b) Link protection. (c) Node protection.

Fig. 4: Protection schemes [29].



(a) Fully Protected SFC.



(b) Partially Protected SFC.

Fig. 5: VNF placement availability calculation.

an example. SFCs s-a-b-d and s-c-b-d are partially protected, since they jointly place VNF f_2 on node b and they traverse the same link (b, d) as well. Consequently, the total availability is calculated as $1-(1-A_{sa}\circ A_a\circ A_{ab}\circ A_b\circ A_{bd})\circ(1-A_{sc}\circ A_c\circ A_{cb}\circ A_b\circ A_{bd}) = A_a\circ A_b\circ A_d\circ A_{sa}\circ A_{ab}\circ A_{bd}+A_c\circ A_b\circ A_d\circ A_{sc}\circ A_{cb}\circ A_{bd}-A_a\circ\underline{A_b}\circ A_d\circ A_{sa}\circ A_{ab}\circ\underline{A_{bd}}\circ A_c\circ\underline{A_b}\circ A_d\circ A_{sc}\circ A_{cb}\circ\underline{A_{bd}} = A_a\cdot A_b\cdot\overline{A_d}\cdot A_{sa}\cdot A_{ab}\cdot A_{bd}+\overline{A_c}\cdot A_b\cdot\overline{A_d}\cdot A_{sc}\cdot A_{cb}\cdot A_{bd}-\overline{A_a}\cdot\underline{A_b}\cdot A_d\cdot A_{sa}\cdot A_{ab}\cdot\underline{A_{bd}}\cdot A_c\cdot A_d\cdot A_{sc}\cdot A_{cb} \approx 0.759$.

4 Resource Allocation in NFV

In this section, we survey literature about resource allocation according to different QoS parameters (e.g., cost, delay, availability, energy). We also survey the related literature in the context of edge clouds, online and distributed provisioning schemes. In each category (subsection), we further survey the literature based on four generalized resource allocation problems in Section 2.1, which are VPTR problem, VNFP problem, TRR problem and VRC problem. We conclude each subsection with a discussion.

4.1 Cost-Aware Resource Allocation

In this subsection, we survey the literature about cost-aware resource allocation in NFV. This is the basic and fundamental resource allocation problem in NFV, whose problem formulation is shown in Section 2.4.

4.1.1 VPTR problem

Ma *et al.* [35] study the VPTR problem to minimize the maximum link load. They prove that the VPTR problem is NP-hard even under the non-ordered VNF dependence case by a reduction to the NP-hard Hamiltonian Cycle problem [36]. Ma *et al.* [37] later propose a Least First Greatest Last (LFGL)-based min-max routing algorithm to solve the VPTR problem and implement the proposed algorithm in an SDN-enabled environment. In the non-ordered VNF dependence case, Cohen *et al.* [38] propose a near-optimal approximation algorithm using rounding technique to solve the VPTR problem to minimize the total system cost. The total system costs in [38] consists of the setup costs of a function f placed on node n and the sum of the distances between the clients and the nodes from which they get service. Ghaznavi *et al.* [39] advocate to allocate the requested instance of the VNF on multiple nodes in order to minimize the link load ratio (or maximize the throughput). They subsequently present a local search heuristic that employs a tuning parameter to balance the speed-accuracy trade-off to solve the VPTR problem. Spinnewyn *et al.* [40] jointly consider both partially ordered chain and location constraint in the VPTR problem. They first formulate the problem as an ILP based on an augmented VNF tree with precomputed SFCs which satisfy the precedence requirements. Subsequently, they propose an efficient heuristic based on greedy chain selection and embeddings.

Kuo *et al.* [41] relax/approximate the VPTR problem based on the intuition that placing VNFs on a shorter path consumes less link bandwidth, but might also reduce VM reuse opportunities; reusing more VMs might lead to a longer path, and so it consumes more link bandwidth. Feng *et al.* [42] propose an $O(\epsilon)$ approximation algorithm in time $O(1/\epsilon)$ to find the minimum cost solution for the VPTR problem for a given set of requests, where ϵ denotes a tunable parameter for setting the approximation ratio. In particular, multiple paths are allowed to route the traffic among each requested VNF pair. Guo *et al.* [43] jointly consider the VPTR problem in datacenters. They propose a

randomized approximation algorithm when the traffic matrix is known in advance and a competitive online algorithm when the future arriving traffic is not known. However, they assume that one configuration in data centers consists of one traffic routing path and a VNF placement solution. A (limited) set of configurations is given as an input in their problem, which simplifies the problem to some extent. Hamann and Fischer [44] formulate the VPTR problem as an ILP where (a set of k) paths between each node pair in the network are precalculated and known.

4.1.2 VNFP Problem

You and Li [45] model the VNF placement constraint via a bipartite graph and present a load-balanced max-min heuristic to solve the VNFP problem. Ma *et al.* [35] propose an exact polynomial-time algorithm to solve the VNFP problem for the non-ordered VNF dependence case, and prove that the VNFP problem for the totally and partially-ordered dependence case is NP-hard. A dynamic programming and an efficient heuristic are proposed to solve the VNFP problem under these two cases, respectively. Pham *et al.* [46] propose a sampling-based Markov approximation algorithm to jointly minimize the operational and network traffic cost for the VNFP problem. Tomassilli *et al.* [47] study the VNFP problem with the aim of minimizing total costs for servicing a set of requests. By transforming this problem to a hitting-cut problem, Tomassilli *et al.* [47] propose two logarithmic factor approximation algorithms. The first algorithm is based on LP rounding and the second one is a greedy algorithm. In [48], the time-varying workload of requests has been considered in the VNFP problem. By saying the two workloads have high correlation, Li *et al.* [48] mean that their peaks and valleys coincide with each other frequently. In this sense, every workload variable within time interval is corresponding to a sequence of values. They propose an exact Integer Nonlinear Programming (INLP) and a two-stage algorithm consisting of a correlation-based greedy algorithm. They further adjust the algorithm to solve the VNFP problem. Basta *et al.* [49] propose three optimization models that aim at minimizing the network load cost and data center resources cost by finding the optimal placement of the data centers positions and the SDN and NFV mobile network functions.

4.1.3 TRR Problem

Wang *et al.* [50] propose a distributed Alternating Direction Method of Multipliers (ADMM) algorithm to solve the load-balanced TRR problem for multiple requests. The proposed algorithm in [50] is proved to have a fixed coverage rate.

The SFC constraint implies that from the source to the destination the route must traverse each ordered VNF. Sallam *et al.* [51] devises a polynomial-time algorithm to solve the SFC-constraint Shortest Path (SP) problem via an auxiliary graph. The SFC-constraint in the Maximum Flow (MF) problem imposes that each flow traverses a set of network functions in a pre-specified order before reaching its destination. They later propose an ILP to solve how to place VNFs such that the value of the maximum flow with SFC constraint is equal to the value of the original maximum flow without SFC constraint. In the Multicast TRaffic Routing (MTRR) problem in NFV, there is one source node and multiple destination nodes, and the problem is to find a multicast tree from the source to the destinations such that each path from the source to the destination traverses required SFC. Xu *et al.* [52] devise an approximation algorithm to solve the MTRR problem by reducing it to the Steiner tree problem in an auxiliary undirected graph.

4.1.4 VRC Problem

In [53], it is assumed that the network lifetime consists of several time intervals. At each time interval, the service provider needs to serve two kinds of users, which are new users to allocate their required VNFs and in-service users to migrate their requested VNFs to new locations. They subsequently present an exact ILP formulation to solve the VRC problem and also, they first approximate the ILP by column generation and then devise a heuristic to further accelerate the problem-solving. The column generation leverages the idea to generate only the variables which have the potential to improve the objective function [54].

4.1.5 Discussion

The cost-aware resource allocation problem in NFV is basic and fundamental, since only network cost is taken into account and the problem asks for whether a corresponding solution exists when only the (basic) link and node capacity constraints are satisfied. Nevertheless, even under this case, the VPTR problem together with all its variants are still NP-hard in a generic case, which indicates the difficulty of the problem(s). So far, there is no approximation algorithm to solve the problem in the generic case. Even though some approximation algorithms (e.g., [42], [43]) are proposed, they simplify the problem inputs or constraints to some extent. As a result, only exact ILP solutions and heuristics are presented to solve this problem. In the following, more QoS constraints such as delay and reliability will be incorporated in this problem and we will survey related literature in these fields.

4.2 Delay-Aware Resource Allocation

In this subsection, we will survey the literature about delay-aware resource allocation in NFV, where delay is additionally taken into account on basis of fundamental NFV resource allocation problem formulation in Section 2.4. The related literature mainly adopt a similar delay model that is presented and discussed in Section 3.1.

4.2.1 VPTR and TRR problem

Qu *et al.* [55] consider the VNF transmission and processing delays, and formulate the VPTR problem as a Mixed Integer Linear Program (MILP). However, they assume that the virtual link between two physical nodes can at most process one traffic flow at a time. Zhang *et al.* [56] devise an ADMM-based algorithm to solve the delay-aware VPTR problem. However, they do not consider the nodes' delay in their problem. Li *et al.* [57] address the delay-aware VPTR problem by leveraging a packet queuing model. Sun *et al.* [28] propose and implement a framework that enables (independent) network function to work in parallel, which largely improve NFV performance in terms of delay (as shown in Fig. 3(b)). By duplicating an original graph

with m connected copy graphs, Huin *et al.* [58] present a mathematical formulation with the aid of column generation (using a limited number of configurations) that can scale well with problem inputs (e.g., number of requests or nodes). Allybokus *et al.* [59] propose an exact ILP and a greedy heuristic to solve the VPTR problem for both fully and partially ordered SFCs. However, their solutions only consider a simple path within an SFC. Li *et al.* [60] consider a fat tree data center topology, and study (1) service chaining consolidation, (2) pod assignment, and (3) machine assignment per pod problems in both offline and online situations. The delay concern is also taken into account. Consequently, they propose efficient heuristics to solve these problems.

Moreover, [61] and [62] devise a layered-graph based heuristic to solve the delay-aware TRR problem. The general idea is first to construct a multi-layer graph by creating $|F|$ copies of the original graph, where $|F|$ represents the number of requested VNFs. The inter-layer links are created to connect the same node in different layers, and the inter-layer nodes are created to represent the possible places to host each VNF in each layer. By assigning link delay weights, the proposed algorithm finds the shortest path from the ingress node in layer 1 to the egress node in layer $|F|$. Xie *et al.* [63] study delay-aware multi-source multicast routing problem in NFV, where there are multiple source and destination nodes. Xie *et al.* [63] present a minimum spanning tree-based heuristic that always tries to find common links so that the deployed SFC can be shared by more requests (users).

4.2.2 VNFP problem

Chen *et al.* [64] propose a hidden Markov Chain based heuristic to place VNFs which jointly minimize the cost and delay for a set of given flows. In [65], the NFV service is modeled as an M/M/1 queue. The service rate $\mu(f)$ reflects the amount of CPU each VNF is assigned to, and $\Lambda(f)$ expresses the total arrival rate of requests. As a result, the request processing delay is expressed as:

$$\frac{1}{\mu(f) - \Lambda(f)} \tag{13}$$

Agarwal *et al.* [65] formulate the VNF placement and CPU allocation problem as a non-convex optimization model and subsequently divide this problem into two subproblems. Then, they propose an efficient heuristic to solve the delay-aware VNFP problem, and devise a polynomial time to solve the CPU allocation problem by reducing it to the KKT conditions.

In [66], it is assumed that the packets of a request r arrive stochastically as a Poisson stream with an arrival rate. Zhang *et al.* [66] further model each request as an open Jackson network. They merge several flows of requests into a service instance as one flow, and model each service instance as an M/M/1 queue. Subsequently, they further devise a $2-$ approximation algorithm to solve the delay-aware VNFP problem.

4.2.3 Discussion

When SFC delay constraint is additionally taken into account, we see that the VPTR problem and its variants become more difficult to solve than in Section 4.1. Moreover, some literature usually adopt queueing theory to solve the VPTR problem by incorporating both the queueing delay and processing delay. However, queueing theory usually assumes that the arrival rate of traffic follows the Poisson process and the node processing time follows the exponential distribution, which is not always the case in practice (e.g., bursty traffic).

4.3 Availability/Resilience-Aware Resource Allocation

In this subsection, we will survey the literature about availability-aware resource allocation in NFV, where availability or resilience is additionally considered on basis of fundamental NFV resource allocation problem formulation in Section 2.4. The related literature adopts either a similar or different availability model that we present in Section 3.2, and we will demonstrate it when necessary.

4.3.1 VPTR problem

Beck *et al.* [67] propose a recursive heuristic for survivable VNF placement to guarantee an end-to-end VNF protection. Han *et al.* [68] explore resilient respects of the individual VNF in terms of fault management (e.g., failure detection and automated recovery) or state management. They also discuss existing solutions for these aspects. Herker *et al.* [69] formulate how to calculate the VNF placement availability in data center topology. They also present an efficient heuristic on how to place VNFs and their backups to satisfy the requested availability. Fan *et al.* [70] consider how to compute one backup SFC when the primary SFC is given such that the overall availability is satisfied. However, the considered availability model ignores the global information of the entire SFC, which is not precise enough. Ding *et al.* [71] formulate how to calculate VNF placement availability when at most one backup chaining is allowed. On the basis of [70], given that the primary SFCs are already placed in the network, they [71] propose a heuristic on how to calculate the respective backup SFCs with the minimum cost such that the total availability for each request is satisfied.

Qu *et al.* [72], [73] consider both delay and availability constraints to route and place VNFs. Both node delay and link availability are not taken into account in their model. In addition, in their proposed VNF placement availability calculation model, the VNF placement availability is equal to:

$$\prod_{i=1}^{m} \chi(f_i) \tag{14}$$

and

$$\chi(f_i) = 1 - \prod_{j=1}^{k} (1 - Z_n^{f_i} \cdot A_n) \tag{15}$$

where $Z_n^{f_i}$ is a boolean variable. It is 1 if one replica of f_i is placed on n (assuming at most k replicas of each VNF can be placed), and 0 otherwise. For example the VNF placement availability in Fig. 5 is equal to $0.99 \cdot (1 - (1 - 0.85)(1-0.98)) \cdot 0.99 \approx 0.977$. The proposed VNF placement availability model calculates the total availability based on each *SFC*, while the calculation model in [72], [73] computes the availability based on each *VNF*, which implies that the flow must go through each (redundant) VNF located node in an SFC.

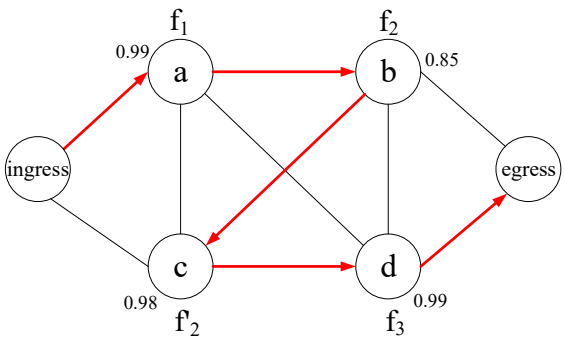


Fig. 6: SFC Protection availability calculation from [72], [73].

In [74], it is assumed that a set of functions with function failure probabilities and a number of servers are given. They tackle how to find an assignment that maximizes the probability for a recovery of all faulty functions or maximizes the expected number of operating functions. In a recovery, each failed function can be matched to one server and each server can be matched to at most one failed function. They prove these two problems are NP-hard in general, and develop heuristics relying on a graph theoretical model to find properties of an efficient assignment. In [75], the authors assume that for n backup machines, at most c functions can be backed up on each machine. Assuming that at most t functions may fail simultaneously, they prove the maximal number of functions that can be recovered with full recovery when $t = 1, 2, 3, 4$. However, they assume that upon failures of some functions, a machine can recover at most one of the functions it implements, which simplifies the problem to some extent. Moreover, in [76], the reliability level is directly reflected as the number of replicas of each requested VNF.

4.3.2 VNFP problem

In [77], it is assumed that there are in total U failure scenarios, and each failure scenario i fails with probability p_i. They formulate the VNFP problem as an ILP, and then decompose it into a master problem and a sub-problem. By leveraging the Generalized Benders Decomposition (GBD), the master problem and the sub-problem are solved iteratively over different partitions. In each iteration, they generate an upper bound and a lower bound of the original problem's objective value. The iterative process stops when the upper bound and the lower bound match the termination condition. Vizarreta *et al.* [78] solve the QoS VNF placement problem, where the end-to-end delay and service chaining availability are considered by proposing an ILP and a heuristic. However, the VNF placement availability in the VNF protection case is not considered in their work.

4.3.3 TRR problem

Yu *et al.* [79] define the reliability level as during an arbitrary single service failure, at most r_j bandwidth is lost, which implies that multiple paths are used to traverse data between each adjacent VNF pair. Assuming that the VNF has already been placed in the network and a set of paths between node pair is given, Yu *et al.* [79] investigate how to find a subset of feasible routing paths between each adjacent VNF pair, such that the reliability requirement is satisfied. They propose a Fully-Polynomial Time Approximation Scheme (FPTAS)[6] based on Primal-Dual theory to solve this problem.

4.3.4 Discussion

The reliability (level) definition varies by different literature, which results in different problem input and constraints as well as solutions. Some literature only considers node/link backup, while some literature additionally takes into account the node and link availability. When $k \geq 2$ disjoint SFCs are provided, the service can recover from $k-1$ simultaneously node and/or link failures. However, this approach ignores the node/link failure probability and its cost is high. On the contrary, the SFC availability calculation is more exact and can quantitatively measure how reliable the NFV service is provided, even though the node and link availability value is not easy to achieve.

4.4 Energy-Efficient Resource Allocation

In this subsection, we will survey the literature about energy-efficient resource allocation in NFV. We will first introduce the main energy consumption model and then survey related literature in this field.

4.4.1 VPTR problem

In [80], Kar *et al.* assume that there are three states for a machine, namely (1) Idle, (2) Off and (3) Active. It is modelled that the energy consumption of a server n in idle state is:

$$E^n_{\text{Idle}} = \frac{\tau(n)}{\pi(n)} \cdot E^n_{\max} \tag{16}$$

where $\tau(n)$ and $\pi(n)$ stand for the default capacity and maximum capacity of n, and $E^n_{\max}$ means the maximum energy consumption. The energy consumption of n in the active state can be expressed as:

$$E^n_{\text{Active}} = \frac{\tau(n) + \theta(n)}{\pi(n)} \cdot E^n_{\max} \tag{17}$$

where $\theta(n)$ implies the total actual load of all the VNFs on n.

Via a M/M/C queuing model, Kar *et al.* [80] propose an exact ILP and an efficient heuristic to solve the energy-aware VPTR problem. While this work is only for the dynamic single request, Jang *et al.* [19] deal with how to minimize the energy consumption for multiple requests by proposing an INLP and a rounding-based heuristic. Huang *et al.* [81] solve the problem of VNF placement and routing in hybrid NFV networks with the objective to maximize the profit of the admitted traffic minus the energy cost and routing cost. A hybrid NFV network models a network where both hardware network function and virtual network function coexist. To solve it, they propose a Markov Approximation based algorithm with a bounded approximation ratio. Eramo *et al.* [82] first target the VPTR problem with the goal of maximizing the amount of data that can be processed within the network at peak traffic. They subsequently study how to

6. An FPTAS has a time complexity that is polynomial in both the problem size and $\frac{1}{\epsilon}$ and produces a solution that is within a factor $(1+\epsilon)$ of the optimal solution (or $(1-\epsilon)$ for maximization problems).

consolidate VNFs and shut down unused servers such that the total operation cost (energy consumption+revenue loss) is minimized. They propose an exact ILP and an efficient heuristic to solve each problem.

4.4.2 VRC problem

Eramo *et al.* [83] formulate the Energy-aware VRC problem and prove that it is NP-hard. Here, some VNFs need to be migrated from their "source" servers to "destination" servers in order to switch off extra idle servers for saving energy. Hence, the total energy consumption consists of (1) consolidation energy consumption: which characterizes the energy to execute the VNFs on "destination" servers and (2) migration energy consumption, which accounts for the energy of keeping active of "source" servers. They subsequently propose a heuristic with the aid of a multi-stage graph, which can represent the energy consumption in different time intervals. In [84], three components are assumed to be energy-consuming, namely (1) servers, (2) network router nodes, some of which are connected with servers and (3) links. When there is no traffic traversing network nodes and links, they can be switched off to save energy. Moreover, they model that the requested resource in terms of capacity of VNF f varies in the range of: $[\widehat{\eta(f)} - \Delta, \widehat{\eta(f)} + \Delta]$, where $\widehat{\eta(f)}$ indicates the average processing capacity of f (requested resources) and Δ implies the maximum deviation. They first present an ILP formulation that solves the Γ-robustness VPTR problem, where the traffic demands are allowed to deviate within a given bound Γ. Subsequently, they present a fast variable fixing heuristic that exploits structural information coming from the linear relaxation of the problem.

4.4.3 Discussion

The energy consumption can be regarded as a "cost", but it is not linear according to Eqs. (16)-(17). Due to this reason, the corresponding solutions for solving the cost-aware resource allocation in NFV cannot be directly applied here. Moreover, due to the nonlinearity and nonconvexity of energy constraints, it is difficult to devise approximation algorithms via the aforementioned approaches in Section 2.5. Therefore, the algorithms that apply column-generation, GBD or ADMM approaches may be efficient to work based on an exact solution.

4.5 (Multi-Objective) Resource Allocation in Edge Cloud

The concept of Mobile Edge Computing (MEC) [85] has been proposed to bring the computing resources closer to end users by installing small resource-limited cloud infrastructures at the network edge (also called edge clouds), so as to provide delay-guaranteed services to end users. Applying NFV to MEC will not only shorten the servicing delay for end users but service providers will also benefit from lower expenditures and higher efficiency. The user requested service, in this context, consists of a sequence of VNFs that need to be placed on edge or public clouds. In this subsection, we will survey the literature about resource allocation in NFV in Edge Cloud.

4.5.1 VPTR problem

In [86], Riggio *et al.* address the wireless VNF placement and traffic routing problem in the radio access network. Compared to the conventional wireline networks, there is an additional requirement, which is called radio resources (but also additive, similar to e.g., cost) for both VNF and node. Riggio *et al.* [86] present an ILP and a heuristic for the problem. Wang *et al.* [87] study the VPTR problem to jointly minimize the ratio of resource utilization regarding link load ratio and node's CPU utilization ratio. They present a rounding-based approximation algorithm to solve this problem. Yang *et al.* [88] propose an approximation algorithm to solve the VPTR problem in edge clouds by taking into account both fully ordered and partially ordered SFCs. The proposed approximation algorithm in [88] leverages randomized rounding technique and assumes that the paths between each node pair are known/given.

Moreover, there are also some work addressing the multi-objective VPTR problem (e.g., jointly optimize cost, delay, reliability, etc.). However, as the VPTR problem itself is NP-hard to solve, the multi-objective VPTR problem is even harder and only straightforward heuristics [89], [90], [91] are devised to solve it.

4.5.2 VNFP problem

Cziva *et al.* [92] tackle the VNFP problem at the network edge in order to minimize the total expected latency from all users to their respective VNFs. They further employ Optimal Stopping Theory to determine when to re-evaluate the optimal placement problem by taking into account the migration cost and random path delay. However, they assume that only one VNF is enough to provide a service to one user instead of a sequence of VNFs. Yang *et al.* [93] study how to place NFV-enabled service on a minimum number of edge nodes and find routes from the access point to the requested service located node in order to meet the delay requirement. They propose a heuristic-based incremental allocation mechanism to solve this problem. Nam *et al.* [94] provide a clustered NFV service chaining (cNSC) scheme that applies the stochastic model to compute the optimal number of clusters that place VNFs according to their popularity to minimize the end-to-end time of MEC services. Jemaa *et al.* [95] model the requests to the end cloud as M/M/1 queue, and the requests to the public cloud as M/G1/∞ queue. Subsequently, they propose an exact ILP to solve how to place the VNFs in edge clouds without violating the delay constraint.

Song *et al.* [96] study the VNFP problem in 5G edge networks by considering the user's mobility. They [96] first propose a user grouping model based on users' context geographical information and then define (and compute the optimum number of the) clusters to minimize the end-to-end delay of network services. Subsequently, a graph partitioning algorithm assigning VNFs to clusters in the edge network is presented to minimize user movement between clusters and optimize the data rate that users lose due to VNF migration.

4.5.3 Discussion

In the context of edge clouds, the VPTR problem needs to additionally consider the user's location as well as both

the wireless transmission between the user and edge server and wireline link transmission between edge server and cloud server. In this sense, jointly considering multiple QoS parameters will make this problem difficult to solve but deserves us to further explore it in future work.

4.6 On-Line Provisioning for NFV

In the on-line provisioning, the knowledge of the future traffic matrix is not known. An online policy x is said to be β-competitive, if $\frac{g(x)}{OPT} \leq \beta$, where $g(x)$ is the achieved cost of policy x and OPT is the cost of the optimal solution. In this subsection, we will focus on surveying the literature about on-line NFV resource allocation in NFV.

4.6.1 Prediction-based Online Scheduling

Mijumbi *et al.* [97] propose a graph neural network-based algorithm by exploiting topology information of service chains, to predict future resource requirements for each VNF component. Bu *et al.* [98] leverage the Auto Regression (AR) to predict the long-term and short-term VNF popularity, which is defined as how frequently a VNF is requested. Based on the prediction, they devise a dynamic network function deployment mechanism, which pre-deploys appropriate function before they are massively requested, and deploys a few of the newly requested functions according to the current network status in real time.

In [99], [100], it is assumed that there is a number of $|T|$ time intervals, and a traffic demand r_i arrives at time t_i and lasts for z_i time intervals. The traffic rate of traffic demand varies over time. The VNF capacity should be no less than the traffic demand's rate, otherwise new VNF instances should be launched in order to satisfy the requested traffic rate. Zhang *et al.* [99] address the cost-aware VNFP problem by employing an online gradient descent (OGD) prediction algorithm to predict the future VNF demand. Fei *et al.* [100] apply an efficient online learning method called follow-the-regularized-leader (FTRL) to predict the upcoming flows with bounded prediction regret compared to the best static prediction strategy. Based on the prediction algorithm, they further devise approximation algorithms on how to place new required VNFs and route flows among them. Together with prediction and approximation algorithms, an online competitive algorithm has been proposed to solve the cost-aware VNFP problem. However, as we mentioned earlier, the traffic in reality behaves dynamically and irregularly, which is not easy to predict. In this sense, the prediction-based online scheduling algorithms may not work very efficiently in practice.

4.6.2 Regularization-based Online Scheduling

Guo *et al.* [101] develop an online algorithm based on Primal-Dual technique to decide whether to accept or reject each incoming NFV request by finding paths between each VNF pair such that the network throughput is maximized. In [101], the paths between each node pair are assumed to be known and the VNFs are also assumed to be placed on existing nodes. Jia *et al.* [102] study the problem of online scaling of NFV service chains across geo-distributed datacenters. Jia *et al.* [102] first leverage the regularization-based technique to transform the integer offline formulation of the problem into a sequence of one-shot sub-problems, and then design an online dependent rounding scheme to obtain the final solution yielding a guaranteed competitive ratio. Similar to the approach in [102], Zhou *et al.* [103] devise an online orchestration framework for placing VNFs and routing in edge clouds by leveraging the regularization and rounding technique. However, in [102], [103], it is assumed that each (edge) cloud or datacenter node is inter-connected with high-speed links and the network constraints such as path selection issues and the link capacity constraint are not taken into account.

Moreover, the aim in [104] is to minimize both the operational costs (placing VNF on a node) and deployment costs (transferring a VM image from one node to another). An online algorithm based on the ski-rental algorithm for the online VPTR problem with the bounded competitive ratio is presented in [104]. Chen *et al.* [105] present a decentralized online approach for optimal placement and operations of VNFs by using a new stochastic dual gradient method, where prices for service processing and routing costs as well as traffic demands are assumed to be stochastic.

4.6.3 Discussion

The surveyed on-line provisioning literatures have shown that a provable competitive ratio can be achieved by their proposed algorithms, which are the main contributions and novelties of these algorithms. Nevertheless, even with a provable bound or ratio, the worst-case performance sometimes is still not very satisfied and this encourages us to further improve the online algorithm's competitive ratio theoretically and increase the efficiency of the algorithms in practice.

4.7 Distributed Provisioning: Game theory

So far, all the surveyed literatures adopt a centralized resource allocation manner, where a global view of the network knowledge is known. In this subsection, we will survey the literatures about game theory approaches which can solve the resource allocation problem in NFV in a distributed way.

4.7.1 C-VNFP problem

Leivadeas *et al.* [106] propose a partition game model to solve the VNFP problem and prove that a Nash Equilibrium (NE) exists. The multiple VNFs can be placed on the same server as components or internal software processes. Hence, the authors define the VNFP problem as a partition problem where each server can be regarded as a partition. The goal of the partition game is that the VNFs can be placed in appropriate cloud sites, while minimizing deployment cost. Bian *et al.* [107] propose a distributed and low-complexity algorithm that is inspired by game theory, where the players are the users who behave selfishly until they reach a NE. Each user subscribes to a specific network service which is denoted by an ordered VNFs chain. The strategies are the VNF locations of each service chain. The utility function is the sum of the latency and congestion. The innovation of [107] is to balance latency and congestion by considering failures due to user/resource unavailability in the model. Chen *et al.* [108] present a mixed strategy non-cooperative

game, where servers compete for the optimal VNFs placement strategies and distribution due to revenue and QoS incentives.

4.7.2 C-VPTR problem

In [109], the players are the Access Points (AP), and the strategies are the set of task rates of each AP. Therefore, the network utility function is defined as the sum of the operational cost and average response time. The problem can be considered as a weighted sum approach of a general multi-objective optimization problem, which is an NP-hard problem. So, the equilibrium of the algorithm is α-approximate equilibrium. In the proposed algorithm, the players selfishly choose their paths among APs and MEC servers with the least cost accordingly. Furthermore, the authors propose an enhanced algorithm based on public service advertising (PSA), which improves the convergence performance and equilibriums efficiency. Obadia *et al.* [110] regard each request as a greedy player trying to optimize its own placement to minimize its own costs, and they propose a game theoretic-based heuristic to solve the cost-aware VPTR problem. The NE is defined as the combination of strategies, where the players choose the optimal placement and the shortest path to route their traffic. D'Dro *et al.* [111] formulate the cost-aware VPTR problem as an atomic weighted congestion game, and show that the game possesses a weighted potential function and admits a NE. Subsequently, they propose a distributed and privacy-preserving algorithm which can converge to the NE to solve the cost-aware VPTR problem.

4.7.3 Discussion

Due to the lack of global network knowledge, it is usually assumed in the above surveyed literature that a set of VNF placement configurations/solutions is known or given. Moreover, in terms of routing issues, either conventional shortest path [110] is adopted or it is assumed that there always exists a path in the network [111]. In this sense, the game theory-based approach does not essentially solve the resource allocation problem in NFV compared to combinatorial optimization-based approaches, and cannot guarantee QoS such as service delay.

5 Emerging Topics and Future Works

There are still open domains or interesting topics that need to be considered in depth about resource allocation in NFV which we will suggest below. In all, a summary of the whole literature review is given in Table 1.

5.1 Machine Learning-based Approaches

With the proliferation of Machine Learning (ML) approaches especially Deep Reinforcement Learning (DRL), researchers have recently explored these methods to solve the resource allocation problem in NFV. For instance, Wahab *et al.* [112] model the VRC problem as an Integer Linear Programming (ILP) problem. Subsequently, they design ML-based algorithms that intelligently eliminate some cost functions from the proposed ILP to boost its feasibility in a large-scale network. Afterwards, for the purpose of parting a large-scale network, they propose an optimized k-medoids clustering approach from ML which proactively partitions the substrate network into a set of disjoint on-demand clusters. Each cluster seeks to optimize some attributes such as CPU, energy, delay, and bandwidth. The approach is time-aware in the sense that it works in a repeated manner to provide network administrators with different placement and readjustment decisions at different time moments. Gupta *et al.* [113] propose a distributed and dynamic placement algorithm namely Predictive-Adaptive Real Time (P-ART) in a multi-cloud environment for advantageous flexibility in optimizing performance and cost. By taking care of long-term traffic variations, the method makes the predictions closer to reality. These predictions are then used by a randomized placement heuristic that carries out a fast cloud selection using a least-cost latency-constrained policy. Specifically, the policy is used as a feature to train ML models such as Random Forest, SVR, KNN, and MLP. Thus, the algorithm optimizes the cost by selecting the clouds and increases the speed of placement. NFVdeep [129] is an adaptive, online, deep reinforcement learning (DRL) method that automatically deploys VNFs with different QoS requirements. The goal of this method is to jointly optimize the throughput and cost. The architecture of NFVdeep consists of two parts: NFV environment and NFVdeep agent. The NFV environment is the NFV network, including the servers and links in the network topology, and the NFVdeep agent is designed as a DNN. In particular, the NFVdeep agent obtains state information from the NFV environment and automatically selects an operation as a return. After the agent takes action, the NFV environment transfers the reward to the agent. Finally, the agent updates the relevant policy according to the reward and repeats the process until the reward converges. Moreover, there are some other work [114], [115], [116], [117] dealing with VNFP problem via DRL approach. Nevertheless, we found that using the ML-based approach to solve resource allocation problems in NFV shows its great potential and needs to receive more attention based on the following challenges/aspects:

- Since an ML-based approach cannot guarantee to find an optimal solution due to the use of training, can we improve the accuracy of ML-based approach?
- How to further reduce the convergence/training time of ML-based approach.
- What is the performance ratio of using ML-based approach for solving NFV resource allocation problem in theory (if any).
- Apart from DRL related approach, can some other ML-based approaches be used to solve NFV resource allocation problem? (e.g., graph neural networks)

5.2 Security-Aware Resource Allocation

As NFV yields numerous benefits such as lower CAPEX and OPEX, the software-enabled functions are vulnerable to a number of security threats [130], compared to the hardware-implemented middleboxes. Even though there is research (please see [131] and papers therein) that analyzes security threats and conduct studies on security mechanisms that are applied in traditional scenarios and in NFV environments, the security-aware resource allocation in NFV receives less

TABLE 1: Summary of the literature review about resource allocation in NFV.

Ref.	Problem	Solution	Technique	Ref.	Problem	Solution	Technique
[37]	C-VPTR	exact heuristic	min-max	[80]	E-VPTR	exact	ILP
[38]	C-VPTR	approximation	rounding	[19]	E-VPTR	heuristic	rounding
[39]	C-VPTR	heuristic	concrete alg.	[81]	E-VPTR	approximation	Markov
[39]	C-VPTR	heuristic	concrete alg.	[83]	E-VRC	heuristic	multi-stage graph
[40]	C-VPTR	heuristic	greedy	[84]	E-VRC	heuristic	variable fixing
[41]	C-VPTR	heuristic	concrete alg.	[86]	EC-VPTR	exact	ILP
[42]	C-VPTR	approximation	relaxation	[87]	EC-VPTR	approximation	rounding
[43]	C-VPTR	approximation	rounding	[88]	EC-VPTR	approximation	rounding
[44]	C-VPTR	exact	ILP	[89] [90] [91]	EC-VPTR	heuristic	concrete
[45]	C-VNFP	heuristic	bipartite graph	[92]	EC-VNFP	heuristic	OST
[35]	C-VNFP	heuristic	dynamic programming	[93]	EC-VNFP	heuristic	incremental allocation
[46]	C-VNFP	heuristic	Markov	[94]	EC-VNFP	heuristic	stochastic theory
[47]	C-VNFP	approximation	rounding	[95]	EC-VNFP	exact	ILP
[48]	C-VNFP	exact, heuristic	correlation-based	[96]	EC-VNFP	heuristic	partitioning
[49]	C-VNFP	exact	ILP	[97]	O-VPTR	heuristic	GNN
[50]	C-TRR	heuristic	ADMM	[98]	O-VNFP	competitive	AR
[51]	SP MF	exact	auxiliary ILP	[99]	O-VNFP	competitive	OGD
[52]	Multicast	approximation	Steiner tree	[100]	O-VNFP	competitive	FTRL
[53]	C-VRC	heuristic	column generation	[101]	O-TRR	competitive	primal-dual
[55]	D-VPTR	exact	MILP	[102]	O-VPTR	competitive	regularization, rounding
[56]	D-VPTR	heuristic	ADMM	[103]	O-VPTR	competitive	regularization, rounding
[57]	D-VPTR	heuristic	queueing	[104]	O-VNFP	competitive	ski-rental
[58]	D-VPTR	exact	column generation	[105]	O-VNFP	competitive	dual gradient
[59]	D-VPTR	heuristic	greedy	[112]	C-VNFP	heuristic	k-medoids clustering
[60]	D-VPTR	heuristic	concrete	[106] [107] [108]	C-VNFP	heuristic	Game theory
[61] [62]	D-TRR	heuristic	layered-graph	[109] [110] [111]	C-VPTR	heuristic	Game theory
[63]	D-multicast	heuristic	MST	[113]	C-VNFP	heuristic	ML
[64]	D-VNFP	heuristic	Markov Chain	[114] [115] [116] [117]	C-VNFP	heuristic	DRL
[65]	D-VNFP	heuristic	KKT	[118]	S-VNFP	heuristic	PRBC
[66]	D-VNFP	approximation	Jackson network	[119]	security	exact	model
[67]	R-VPTR	heuristic	recursive	[120]	security	exact	model
[69]	R-VPTR	heuristic	C-SP	[121]	defense patterns	heuristic	partitioning
[70]	R-VPTR	heuristic	concrete	[122]	multipath	heuristic	load balancing
[71]	R-VPTR	heuristic	concrete	[123]	VNFP mobile 5G	exact	ILP
[72] [73]	R-VPTR	heuristic	K-SP	[124]	CDN	heuristic	bargaining
[74] [75]	R-VPTR	exact heuristic	graph model bounds	[125]	RAN	exact	Benders
[77]	R-VNFP	heuristic	GBD	[126]	VPTR mobility	heuristic	layer graph
[78]	R-VNFP	exact	ILP	[127]	TRR-SDN	heuristic	matching
[79]	R-TRR	approximation	primal-dual	[128]	VPTR-SDN	approximation	Markov approximation

C: Cost, D: Delay, R: Resilience, E: Energy, EC: Edge Clouds, O: Online, S: Security

attention. For example, how to securely place VNFs and (re)route traffic to defend against DDoS attack [132], VM escape attack or other threads. Among the security-related research in NFV, Guan *et al.* [118] claim a subset of network nodes in the network are "key" nodes, and present a Path Routing Betweenness Centrality (PRBC) metric to represent this group of nodes, which implies the maximum probability that packets go through at least one node in this group. They subsequently present a successive heuristic to place Virtual Security Network Function (VSNF) such as firewall and intrusion detection on the calculated PRBC nodes. Park *et al.* [133] presents a light intrusion detection system by utilizing a chain of network functions in NFV based on ClickOS. In [119], [134], the security constraints in NFV refer to that VSNF must be placed on a certain subset of nodes (e.g., close to the user for a shorter delay) and prevented to be placed on a certain subset of nodes (e.g., critical regions from potentially malicious user traffic). An exact ILP and an efficient heuristic are proposed in [134] to solve the VNF placement and routing problem where both delay and security are taken into account. Similarly, in [120], the security constraints include: (1) a set of VNFs that can be co-located, (2) a set of VNFs that should be co-located, (3) a set of VNFs that should not be co-located, and (4) a set of VNFs that should not share instances. Shameli-Sendi *et al.* [121] define a set of network security defense patterns, which means the sequence/relation among required VNFs. More specifically, for any two VNFs, the patterns include, unordered, ordered, composition, location-aware, collaborative, etc. According to these defined security patterns,

Shameli-Sendi *et al.* [121] further propose a partitioning and segmentation-based heuristic to solve the VNFP problem. .

5.3 State-of-the-Art

Since the VPTR problem and its variants are proved to be NP-hard, approximation algorithms with proved approximation performance ratio have been devised, but with an assumption of a loose constraint (e.g., non-ordered SFC, a set of placement and routing configurations are known.) of the problem. We even found that no approximation algorithm has been devised to solve the VRC problem. Hence, we still need to further investigate the "hardness" of the addressed problems, analyze the problem properties and devise approximation algorithms (in the general form) or efficient heuristics. Moreover, in this survey, we assume a single-path routing scheme (unless otherwise specified), which means that the data packets cannot be split through an SFC. Therefore it is interesting to further study the VPTR problem with multiple QoS parameters when multipath routing [42], [122], [135] or multicast routing [52] is allowed.

5.4 Wireless Virtual Network Functions and Other Network Application Domains

While the majority of work as we summarized above have been directed to the scenario where data flow traverses in wireline networks in an SFC, the VPTR problem over wireless networks has received less attention. More specifically, when the nodes are connected via wireless channels, the traffic routing in terms of delay and packet loss among these nodes depends on the signal attenuation intensity, transmitting frequency or other metrics. The data transmission and communication models [136], [137] differ from the ones in the wireline networks, and hence the VPTR problem should be reconsidered. As such, the NFV resource allocation problem in 5G mobile networks [123], network slicing [138], IoT [139] and other network application domains such as Content Delivery Networks (CDN) [124] and Radio Access Networks (RAN) [125] need to be further investigated in future.

5.5 Mobility management in NFV

In 5G-enabled mobile edge computing networks, the network service region is partitioned into different cells and each cell is located with edge servers. The end users stay in one cell whose NFV service request is accommodated by its local edge servers in this cell. Typically, the end user moves erratically, and when an end user roams to another cell, it will trigger handover delay. In this sense, the NFV services should be dynamically migrated [126] among multiple edge clouds to maintain the service performance (e.g., guarantee a user-perceived delay). Therefore, it is necessary to consider the user's mobility by solving which VNF to migrate to which edge server, how to find appropriate paths within an SFC after migration, etc.

5.6 SDN and NFV

Software-Defined Network (SDN) [140] defines a network connection and management methodology that decouples the control plane from the data plane. In SDN, the network intelligence stays in a logically centralized software-controller (control plane), and network equipment (data plane) can be programmed via an open interface (like OpenFlow [141]). NFV and SDN are two independent innovation technologies, but they can work together for a joint flexible, efficient, agile network management and service development [142]. The surveyed resource allocation algorithms in this survey can, for instance, be implemented in an SDN controller in an SDN-enabled NFV framework. Hence, it is necessary to consider network management and control related metrics such as network control overhead when designing resource allocation algorithms [127], [128].

6 Conclusion

NFV offers more flexibility by replacing dedicated hardware implementations with software instances and it provides more possibilities for shorter deployment cycles and service upgrades. In this survey, we first generalize and analyze four representative resource allocation problems (variants), namely, (1) the VNF Placement and Traffic Routing problem, (2) VNF Placement problem, (3) Traffic Routing problem in NFV and (4) the VNF Redeployment and Consolidation problem. After that, we study the SFC delay calculations and VNF protection schemes as well as VNF placement availability in NFV resource allocation. Subsequently, we classify and summarize the representative work for solving the generalized problems by considering various QoS parameters (e.g., cost, delay, reliability, energy) and different scenarios (e.g., edge cloud, online provisioning, distributed provisioning). Finally, we conclude our survey with a short discussion on the state-of-the-art and emerging topics in the related field, and highlight areas where we expect high potential for future research.

Acknowledgments

The work of Song Yang is partially supported by the National Natural Science Foundation of China (NSFC, No. 61802018) and Beijing Institute of Technology Research Fund Program for Young Scholars. The work of Fan Li is partially supported by the NSFC (No. 61772077), and the Beijing Natural Science Foundation (No. 4192051). The work of Xiaoming Fu is partially supported by the EU H2020 RISE COSAFE project (No. 824019). Song Yang is the corresponding author.

References

[1] "Cisco visual networking index: Forecast and methodology, 2016–2021," 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[2] "ETSI Publishes First Specifications for Network Functions Virtualisation," http://www.etsi.org/news-events/news/700-2013-10-etsi-publishes-first-nfv-specifications.

[3] Q. Zhang, Q. Zhu, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, "Dynamic service placement in geographically distributed clouds," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 762–772, 2013.

[4] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint VM placement and routing for data center traffic engineering," in *IEEE INFOCOM*, 2012, pp. 2876–2880.

[5] S. Yang, P. Wieder, R. Yahyapour, S. Trajanovski, and X. Fu, "Reliable virtual machine placement and routing in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2965–2978, Oct 2017.

[6] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[7] B. Yi, X. Wang, K. Li, M. Huang *et al.*, "A comprehensive survey of network function virtualization," *Computer Networks*, 2018.

[8] 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.

[9] G. MIRJALILY, "Optimal network function virtualization and service function chaining: A survey," *Chinese Journal of Electronics*, vol. 27, pp. 704–717, July 2018.

[10] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *Journal of Network and Computer Applications*, vol. 75, pp. 138–155, 2016.

[11] Y. Xie, Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," *CoRR*, vol. abs/1608.00095, 2016. [Online]. Available: http://arxiv.org/abs/1608.00095

[12] S. Demirci and Ş. Sağıroğlu, "Optimal placement of virtual network functions in software defined networks: A survey," *Journal of Network and Computer Applications*, p. 102424, 2019.

[13] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1409–1434, 2019.

[14] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1888–1906, Fourth 2013.

[15] G. Even, M. Rost, and S. Schmid, "An approximation algorithm for path computation and function placement in sdns," in *International Colloquium on Structural Information and Communication Complexity*. Springer, 2016, pp. 374–390.

[16] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for NFV," *IEEE Transactions on Services Computing*, 2019.

[17] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on selected areas in communications*, vol. 20, no. 1, pp. 122–133, 2002.

[18] B. Wu, K. L. Yeung, and S. Xu, "ILP formulation for p-cycle construction based on flow conservation," in *IEEE Global Telecommunications Conference*, 2007, pp. 2310–2314.

[19] I. Jang, D. Suh, S. Pack, and G. Dán, "Joint optimization of service function placement and flow distribution for service function chaining," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2532–2541, 2017.

[20] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Transactions on Mobile Computing*, 2019.

[21] "Lecture notes," https://cse.buffalo.edu/~hungngo/classes/2006/594/notes/Primal-Dual.pdf.

[22] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE transactions on information theory*, vol. 59, no. 10, pp. 6301–6327, 2013.

[23] Wikipedia contributors, "Local search (optimization) — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Local_search_(optimization)&oldid=928471895, 2019, [Online; accessed 3-January-2020].

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[25] L. R. Ford Jr and D. R. Fulkerson, "A suggested computation for maximal multi-commodity network flows," *Management Science*, vol. 5, no. 1, pp. 97–101, 1958.

[26] A. M. Geoffrion, "Generalized benders decomposition," *Journal of optimization theory and applications*, vol. 10, no. 4, pp. 237–260, 1972.

[27] X. Li, A. Tomasgard, and P. I. Barton, "Nonconvex generalized benders decomposition for stochastic separable mixed-integer nonlinear programs," *Journal of optimization theory and applications*, vol. 151, no. 3, p. 425, 2011.

[28] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "NFP: Enabling network function parallelism in NFV," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2017, pp. 43–56.

[29] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, and A. Pattavina, "Virtual network function placement for resilient service chain provisioning," in *8th IEEE/IFIP International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2016, pp. 245–252.

[30] J. Kong, I. Kim, X. Wang, Q. Zhang, H. C. Cankaya, W. Xie, T. Ikeuchi, and J. P. Jue, "Guaranteed-availability network function virtualization with network protection and vnf replication," in *IEEE Global Communications Conference*, 2017, pp. 1–6.

[31] J. I. McCool, *Probability and Statistics With Reliability, Queuing and Computer Science Applications*. Taylor & Francis, 2003.

[32] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *Proc. of the 1st ACM symposium on Cloud computing*, 2010, pp. 193–204.

[33] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems." in *OSDI*, vol. 10, 2010, pp. 1–7.

[34] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, 2011, pp. 350–361.

[35] W. Ma, O. Sandoval, J. Beltran, D. Pan, and N. Pissinou, "Traffic aware placement of interdependent NFV middleboxes," in *IEEE INFOCOM*, 2017.

[36] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Co., 1979.

[37] W. Ma, J. Beltran, Z. Pan, D. Pan, and N. Pissinou, "SDN-based traffic aware placement of NFV middleboxes," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 528–542, 2017.

[38] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *IEEE INFOCOM*, 2015, pp. 1346–1354.

[39] M. Ghaznavi, N. Shahriar, S. Kamali, R. Ahmed, and R. Boutaba, "Distributed service function chaining," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2479–2489, 2017.

[40] B. Spinnewyn, P. H. Isolani, C. Donato, J. F. Botero, and S. Latré, "Coordinated service composition and embedding of 5G location-constrained network functions," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1488–1502, Dec 2018.

[41] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *IEEE INFOCOM*, 2016, pp. 1–9.

[42] H. Feng, J. Liorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the NFV service distribution problem," in *IEEE INFOCOM*, 2017.

[43] L. Guo, J. Pang, and A. Walid, "Joint placement and routing of network function chains in data centers," in *IEEE INFOCOM*, 2018.

[44] M. Hamann and M. Fischer, "Path-based optimization of nfv-resource allocation in sdn networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[45] C. You and L. M. Li, "Efficient load balancing for the VNF deployment with placement constraints," in *IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[46] 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*, 2017.

[47] A. Tomassillik, F. Giroire, N. Huin, and S. Pérennes, "Provably efficient algorithms for placement of service function chains with ordering constraints," in *IEEE INFOCOM*, 2018.

[48] D. Li, P. Hong, K. Xue, and J. Pei, "Virtual network function placement considering resource optimization and SFC requests in cloud datacenter," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 7, pp. 1664–1677, 2018.

[49] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, "Towards a cost optimal design for a 5G mobile core network based on SDN and NFV," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1061–1075, 2017.

[50] T. Wang, H. Xu, and F. Liu, "Multi-resource load balancing for virtual network functions," in *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 1322–1332.

[51] G. Sallam, G. R. Gupta, and B. Ji, "Shortest path and maximum flow problems under service function chaining constraints," *IEEE INFOCOM*, 2018.

[52] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Approximation and online algorithms for NFV-enabled multicasting in SDNs," in *IEEE ICDCS*, 2017, pp. 625–634.

[53] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 543–553, 2017.

[54] V. Chvatal, V. Chvatal *et al.*, *Linear programming*. Macmillan, 1983.

[55] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3746–3758, 2016.

[56] Z. Zhang, Z. Li, C. Wu, and C. Huang, "A scalable and distributed approach for NFV service chain cost minimization," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2151–2156.

[57] Q. Li, Y. Jiang, P. Duan, M. Xu, and X. Xiao, "Quokka: Latency-aware middlebox scheduling with dynamic resource allocation," *Journal of Network and Computer Applications*, vol. 78, pp. 253–266, 2017.

[58] N. Huin, B. Jaumard, and F. Giroire, "Optimal network service chain provisioning," *IEEE/ACM Transactions on Networking*, 2018.

[59] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi, and E. Gourdin, "Virtual function placement for service chaining with partial orders and anti-affinity rules," *Networks*, vol. 71, no. 2, pp. 97–106, 2017.

[60] Y. Li, L. T. X. Phan, and B. T. Loo, "Network functions virtualization with soft real-time guarantees," in *IEEE INFOCOM*, 2016, pp. 1–9.

[61] A. Dwaraki and T. Wolf, "Adaptive service-chain routing for virtual network functions in software-defined networks," in *Proceedings of the 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ser. ACM HotMIddlebox, 2016, pp. 32–37.

[62] J. Pei, P. Hong, K. Xue, and D. Li, "Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2179–2192, 2018.

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

[64] H. Chen, X. Wang, Y. Zhao, T. Song, Y. Wang, S. Xu, and L. Li, "MOSC: a method to assign the outsourcing of service function chain across multiple clouds," *Computer Networks*, vol. 133, pp. 166–182, 2018.

[65] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, "Joint VNF placement and CPU alloaction in 5G," in *IEEE INFOCOM*, 2018.

[66] Q. Zhang, Y. Xiao, F. Liu, J. C. Lui, J. Guo, and T. Wang, "Joint optimization of chain placement and request scheduling for network function virtualization," in *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 731–741.

[67] M. T. Beck, J. F. Botero, and K. Samelin, "Resilient allocation of service function chains," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2016, pp. 128–133.

[68] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, "On the resiliency of virtual network functions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 152–157, 2017.

[69] S. Herker, X. An, W. Kiess, S. Beker, and A. Kirstaedter, "Data-center architecture impacts on virtualized network functions service chain embedding with high availability requirements," in *Globecom Workshops (GC Wkshps), 2015 IEEE*, 2015, pp. 1–7.

[70] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "Grep: Guaranteeing reliability with enhanced protection in NFV," in *ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, 2015, pp. 13–18.

[71] W. Ding, H. Yu, and S. Luo, "Enhancing the reliability of services in nfv with the cost-efficient redundancy scheme," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[72] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in nfv-enabled enterprise datacenter networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 554–568, 2017.

[73] L. Qu, M. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 558–573, 2018.

[74] Y. Kanizo, O. Rottenstreich, I. Segall, and J. Yallouz, "Optimizing virtual backup allocation for middleboxes," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2759–2772, 2017.

[75] ——, "Designing optimal middlebox recovery schemes with performance guarantees," *IEEE INFOCOM*, 2018.

[76] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for loac balancing in NFV networks," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[77] P. Zhao and G. Dán, "Resilient placement of virtual process control functions in mobile edge clouds," in *IFIP Networking*, 2017.

[78] P. Vizarreta, M. Condoluci, C. Mahuca, T. Mahmoodi, and W. Kellerer, "QoS-driven function placement reducing expenditures in NFV deployments," in *IEEE ICC*, 2017.

[79] R. Yu, G. Xue, and X. Zhang, "QoS-aware and reliable traffic steering for service function chaining in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2522–2531, 2017.

[80] B. Kar, E. H.-K. Wu, and Y.-D. Lin, "Energy cost optimization in dynamic placement of virtualized network function chains," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 372–386, 2018.

[81] H. Huang, S. Guo, J. Wu, and J. Li, "Service chaining for hybrid network function," *IEEE Transactions on Cloud Computing*, 2017.

[82] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2008–2025, 2017.

[83] V. Eramo, M. Ammar, and F. G. Lavacca, "Migration energy aware reconfigurations of virtual network function instances in nfv architectures," *IEEE Access*, vol. 5, pp. 4927–4938, 2017.

[84] A. Marotta, F. D'Andreagiovanni, A. Kassler, and E. Zola, "On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures," *Computer Networks*, vol. 125, pp. 64–75, 2017.

[85] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," *ETSI White paper*, 2015.

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

[87] J. Wang, H. Qi, K. Li, and X. Zhou, "PRSFC-IoT: A performance and resource aware orchestration system of service function chaining for internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1400–1410, 2018.

[88] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Transactions on Mobile Computing*, 2019.

[89] M. Yoshida, W. Shen, T. Kawabata, K. Minato, and W. Imajuku, "Morsa: A multi-objective resource scheduling algorithm for nfv infrastructure," in *The 16th Asia-Pacific Network Operations and Management Symposium*, Sep. 2014, pp. 1–6.

[90] C. Han, S. Xu, S. Guo, X. Qiu, A. Xiong, P. Yu, K. Guo, and D. Guo, "A multi-objective service function chain mapping mechanism for iot networks," in *15th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2019, pp. 72–77.

[91] C. Zhang, X. Wang, Y. Zhao, A. Dong, F. Li, and M. Huang, "Cost efficient and low-latency network service chain deployment across multiple domains for sdn," *IEEE Access*, vol. 7, pp. 143 454–143 470, 2019.

[92] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal vNF placement at the network edge," in *IEEE INFOCOM*, 2018.

[93] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 475–488, March 2018.

[94] Y. Nam, S. Song, and J.-M. Chung, "Clustered NFV service chaining optimization in mobile edge clouds," *IEEE Communications Letters*, vol. 21, no. 2, pp. 350–353, 2017.

[95] F. B. Jemaa, G. Pujolle, and M. Pariente, "Qos-aware VNF placement optimization in edge-central carrier cloud architecture," in *IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.

[96] S. Song, C. Lee, H. Cho, G. Lim, and J. Chung, "Clustered virtualized network functions resource allocation based on context-aware grouping in 5G edge networks," *IEEE Transactions on Mobile Computing*, 2019.

[97] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, "A connectionist approach to dynamic resource management for virtualised network functions," in *IEEE International Conference on Network and Service Management (CNSM)*, 2016, pp. 1–9.

[98] C. Bu, X. Wang, M. Huang, and K. Li, "SDNFV-based dynamic network function deployment: Model and mechanism," *IEEE Communications Letters*, vol. 22, no. 1, pp. 93–96, 2018.

[99] X. Zhang, C. Wu, Z. Li, and F. C. M. Lau, "Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization," in *IEEE INFOCOM*, 2017.

[100] X. Fei, F. Liu, H. Xu, and H. Jin, "Adaptive vnf scaling and flow routing with proactive demand prediction,"," in *IEEE INFOCOM*, 2018.

[101] L. Guo, J. Pang, and A. Walid, "Dynamic service function chaining in SDN-enabled networks with middleboxes," in *IEEE 24th International Conference on Network Protocols (ICNP)*, 2016, pp. 1–10.

[102] Y. Jia, C. Wu, Z. Li, F. Le, and A. Liu, "Online scaling of NFV service chains across geo-distributed datacenters," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 699–710, 2018.

[103] Z. Zhou, Q. Wu, and X. Chen, "Online orchestration of cross-edge service function chaining for cost-efficient edge computing," *IEEE Journal on Selected Areas in Communications*, 2019.

[104] X. Wang, C. Wu, F. Le, A. Liu, Z. Li, and F. Lau, "Online VNF scaling in datacenters," in *IEEE International Conference on Cloud Computing*, 2016, pp. 140–147.

[105] X. Chen, W. Ni, T. Chen, I. B. Collings, X. Wang, R. P. Liu, and G. B. Giannakis, "Multi-timescale online optimization of network function virtualization for service chaining," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2899–2912, 2019.

[106] A. Leivadeas, G. Kesidis, M. Falkner, and I. Lambadaris, "A graph partitioning game theoretical approach for the vnf service chaining problem," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 890–903, 2017.

[107] S. Bian, X. Huang, Z. Shao, X. Gao, and Y. Yang, "Service chain composition with failures in nfv systems: A game-theoretic perspective," in *IEEE ICC*, 2019, pp. 1–6.

[108] X. Chen, Z. Zhu, J. Guo, S. Kang, R. Proietti, A. Castro, and S. Yoo, "Leveraging mixed-strategy gaming to realize incentive-driven vnf service chain provisioning in broker-based elastic optical inter-datacenter networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. A232–A240, 2018.

[109] B. Wu, J. Zeng, L. Ge, S. Shao, Y. Tang, and X. Su, "Resource allocation optimization in the nfv-enabled mec network based on game theory," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

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

[111] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "Exploiting congestion games to achieve distributed service chaining in NFV networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 407–420, 2017.

[112] O. A. Wahab, N. Kara, C. Edstrom, and Y. Lemieux, "Maple: A machine learning approach for efficient placement and adjustment of virtual network functions," *Journal of Network and Computer Applications*, 2019.

[113] L. Gupta, R. Jain, A. Erbad, and D. Bhamare, "The P-ART framework for placement of virtual network services in a multi-cloud environment," *Computer Communications*, vol. 139, pp. 103–122, 2019.

[114] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in sdn/nfv-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 263–278, 2020.

[115] J. Pei, P. Hong, K. Xue, D. Li, D. S. L. Wei, and F. Wu, "Two-phase virtual network function selection and chaining algorithm based on deep learning in sdn/nfv-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1102–1117, 2020.

[116] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 292–303, 2020.

[117] M. Karimzadeh-Farshbafan, V. Shah-Mansouri, and D. Niyato, "A dynamic reliability-aware service placement for network function virtualization (nfv)," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 318–333, 2020.

[118] J. Guan, Z. Wei, and I. You, "GRBC-based network security functions placement scheme in SDS for 5G security," *Journal of Network and Computer Applications*, vol. 114, pp. 48–56, 2018.

[119] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, and E. Salvadori, "Application-centric provisioning of virtual security network functions," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 276–279.

[120] H. Jmila and G. Blanc, "Designing security-aware service requests for nfv-enabled networks," in *IEEE 28th International Conference on Computer Communication and Networks (ICCCN)*, 2019, pp. 1–9.

[121] A. Shameli-Sendi, Y. Jarraya, M. Pourzandi, and M. Cheriet, "Efficient provisioning of security service function chaining using network security defense patterns," *IEEE Transactions on Services Computing*, vol. 12, no. 4, pp. 534–549, July 2019.

[122] Q. Wang, G. Shou, Y. Liu, Y. Hu, Z. Guo, and W. Chang, "Implementation of multipath network virtualization with SDN and NFV," *IEEE Access*, 2018.

[123] O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "Joint VNF placement and multicast traffic routing in 5G core networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[124] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 616–627, 2018.

[125] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "FluidRAN: Optimized vRAN/MEC orchestration," in *Proc. IEEE Infocom*, 2018.

[126] Y.-T. Chen and W. Liao, "Mobility-aware service function chaining in 5G wireless networks with mobile edge computing," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[127] C. Bu, X. Wang, H. Cheng, M. Huang, K. Li, and S. K. Das, "Enabling adaptive routing service customization via the integration of SDN and NFV," *Journal of Network and Computer Applications*, vol. 93, pp. 123–136, 2017.

[128] S. Q. Zhang, A. Tizghadam, B. Park, H. Bannazadeh, and A. Leon-Garcia, "Joint NFV placement and routing for multicast service on SDN," in *IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 333–341.

[129] Y. Xiao, Q. Zhang, F. Liu, J. Wang, M. Zhao, Z. Zhang, and J. Zhang, "NFVdeep: adaptive online service function chain deployment with deep reinforcement learning," in *Proceedings of the International Symposium on Quality of Service*. ACM, 2019, p. 21.

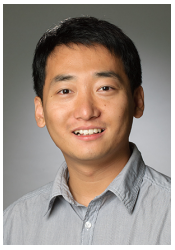
[130] S. Lal, T. Taleb, and A. Dutta, "NFV: Security threats and best practices," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 211–217, 2017.

[131] M. Pattaranantakul, R. He, Q. Song, Z. Zhang, and A. Meddahi, "Nfv security survey: From use case driven threat analysis to state-of-the-art countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3330–3368, 2018.

[132] B. Rashidi, C. Fung, and E. Bertino, "A collaborative DDoS defence framework using network function virtualization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2483–2497, 2017.

[133] Y. Park, P. Chandaliya, A. Muralidharan, N. Kumar, and H. Hu, "Dynamic defense provision via network functions virtualization," in *Proceedings of the ACM International Workshop on Security*

in Software Defined Networks & Network Function Virtualization, 2017, pp. 43–46.
[134] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, M. Savi, and E. Salvadori, "Dynamic and application-aware provisioning of chained virtual security network functions," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 294–307, 2020.
[135] T.-M. Pham and L. M. Pham, "Load balancing using multipath routing in network functions virtualization," in *IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2016, pp. 85–90.
[136] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.
[137] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
[138] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, p. 106984, 2020.
[139] C. Mouradian, T. Saha, J. Sahoo, M. Abu-Lebdeh, R. Glitho, M. Morrow, and P. Polakos, "Network functions virtualization architecture for gateways for virtualized wireless sensor and actuator networks," *IEEE Network*, vol. 30, no. 3, pp. 72–80, 2016.
[140] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
[141] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.
[142] Q. Duan, N. Ansari, M. Toy *et al.*, "Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks." *IEEE Network*, vol. 30, no. 5, pp. 10–16, 2016.

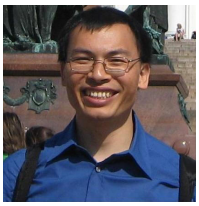


Song Yang is currently an associate professor at School of Computer Science and Technology in Beijing Institute of Technology, China. Song Yang received the B.S. degree in software engineering and the M.S. degree in computer science from Dalian University of Technology, Dalian, Liaoning, China, in 2008 and 2010, respectively, and the Ph.D. degree from Delft University of Technology, The Netherlands, in 2015. From August 2015 to August 2017, he worked as postdoc researcher for the EU FP7 Marie Curie Actions CleanSky Project in Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), Göttingen, Germany. His research interests focus data communication networks, cloud/edge computing and network function virtualization. He is a member of IEEE and ACM.

Fan Li received the PhD degree in computer science from the University of North Carolina at Charlotte in 2008, MEng degree in electrical engineering from the University of Delaware in 2004, MEng and BEng degrees in communications and information system from Huazhong University of Science and Technology, China in 2001 and 1998, respectively. She is currently a professor at School of Computer Science in Beijing Institute of Technology, China. Her current research focuses on wireless networks, ad hoc and sensor networks, and mobile computing. Her papers won Best Paper Awards from IEEE MASS (2013), IEEE IPCCC (2013), ACM MobiHoc (2014), and Tsinghua Science and Technology (2015). She is a member of IEEE and ACM.

Stojan Trajanovski received his PhD degree (cum laude, 2014) from Delft University of Technology, The Netherlands and his master degree in Advanced Computer Science (with distinction, 2011) from the University of Cambridge, United Kingdom. He is currently an applied scientist in Microsoft, working in London, UK and Bellevue, WA, USA. He was in a similar role with Philips Research in Eindhoven, The Netherlands from 2016 to 2019. Before that, he spent some time as a postdoctoral researcher at the University of Amsterdam and at Delft University of Technology. He successfully participated at international science olympiads, winning a bronze medal at the International Mathematical Olympiad (IMO) in 2003. His main research interests include network science & complex networks, machine learning, game theory, and optimization algorithms.

Ramin Yahyapour is full professor at the Georg-August University of Göttingen. He is also managing director of the GWDG, a joint compute and IT competence center of the university and the Max Planck Society. Dr. Yahyapour holds a doctoral degree in Electrical Engineering and his research interest lies in the area of efficient resource allocation in its application to service-oriented infrastructures, clouds, and data management. He is especially interested in data and computing services for eScience. He gives lectures on parallel processing systems, service computing, distributed systems, cloud computing, and grid technologies. He was and is active in several national and international research projects. Ramin Yahyapour serves regularly as reviewer for funding agencies and consultant for IT organizations. He is organizer and program committee member of conferences and workshops as well as reviewer for journals.



Xiaoming Fu received his Ph.D. in computer science from Tsinghua University, Beijing, China in 2000. He was then a research staff at the Technical University Berlin until joining the University of Göttingen, Germany in 2002, where he has been a professor in computer science and heading the Computer Networks Group since 2007. He has spent research visits at universities of Cambridge, Uppsala, UPMC, Columbia, UCLA, Tsinghua, Nanjing, Fudan, and PolyU of Hong Kong. Prof. Fu's research interests include network architectures, protocols, and applications. He is currently an editorial board member of IEEE Communications Magazine, IEEE Transactions on Network and Service Management, and Elsevier Computer Communications, and has served on the organization or program committees of leading conferences such as INFOCOM, ICNP, ICDCS, MOBICOM, MOBIHOC, CoNEXT, ICN and COSN. He is an IEEE Senior Member, an IEEE Communications Society Distinguished Lecturer, a fellow of IET and member of the Academia Europaea.