Kianpisheh, Somayeh; Glitho, Roch H.

Joint Admission Control and Resource Allocation with Parallel VNF Processing for Time-Constrained Chains of Virtual Network Functions

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2021.3129710

Published: 01/01/2021

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2021, accepted November 17, 2021, date of publication November 19, 2021,
date of current version December 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129710

Joint Admission Control and Resource Allocation With Parallel VNF Processing for Time-Constrained Chains of Virtual Network Functions

SOMAYEH KIANPISHEH[1] AND ROCH H. GLITHO[2,3], (Senior Member, IEEE)
[1]Department of Communications and Networking, School of Electrical Engineering, Aalto University, 00076 Espoo, Finland
[2]Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC H3G 1M8, Canada
[3]Computer Science Program, University of the Western Cape, Bellville 7535, South Africa

Corresponding author: Somayeh Kianpisheh (somayeh.kianpisheh@aalto.fi)

This work was partially funded by the Canada Research Chair programme.

ABSTRACT Network Function Virtualization decouples network function deployment from dedicated hardware and reduces costs. Network services are structured as chains of VNFs. Each chain is a set of VNFs that should be executed according to a predefined order. For some applications, VNF chains should be executed within time constraints to meet the application's objectives. Most studies provide a solution to allocate substrate network resources to the chains without considering admission control. Allocating resources to all chains may not be possible due to resource limitations. Efficient admission control is therefore required to determine chains admission. This paper proposes a joint admission control and resource allocation mechanism for VNF chains. We propose a resource allocation mechanism based on the idea of parallel VNF processing to meet tight time constraints. As the used assumptions in deterministic modeling of the system do not hold in a wide range of network conditions, we propose a stochastic modeling at which VNF chain execution is modeled by a Queue network. The Queue network is analyzed to calculate the expected value of the probability of deadline meeting in chains, according which the joint resource allocation and admission control problem is modeled as a non-linear optimization. The proposed optimization framework maximizes the profit of the network provider while keeping the confidence level of deadline-meeting for the admitted chains at desired levels. To have an efficient power usage, power consumption is also considered in network provider profit calculation. A heuristic for the joint resource allocation and admission control of VNF chains is proposed. The effectiveness of the proposed method is demonstrated through simulation.

INDEX TERMS Network function virtualization, resource allocation, admission control, queue network, Tabu search.

I. INTRODUCTION

The advances in information and communication technology, as well as the high capacity and low latency requirements in new generations of communications, necessitate the use of emerging technologies such as Network Function Virtualization (NFV) [1] and Software-Defined Networks (SDNs) [2]. NFV enables network functions like Network Address Translators, Intrusion Detection Systems, Intrusion Prevention Systems, firewalls, and WAN optimizers to be executed on Virtual Machines (VMs) hosted on general-purpose hardware. This decouples network function deployment from dedicated hardware, thereby reducing Capital Expenditures (CAPEX) and Operation Expenditures (OPEX) while enhancing network flexibility [1]. Such Virtualized Network Functions are referred to as VNFs. SDN technology is utilized to decouple network control logic from the underlying transport infrastructure, i.e., switches and routers, in the data plane. Therefore, network switches/routers will function as forwarding devices operated by a logically-centralized controller [2].

Network providers can offer services with different Quality of Services (QoSs) requirements using a shared infrastructure. Each service can be considered as a chain

The associate editor coordinating the review of this manuscript and approving it for publication was Antonino Orsino.

of Virtual Network Functions (VNFs) that are designed to operate on the traffic flow from a source node to a destination node [1], [3], [4]. A major challenge in the management and orchestration of services in NFV is allocating substrate network resources to the VNF chains in such a way that application requirements are satisfied [1], [5]. To allocate resources to a VNF chain, the involved VNFs are assigned to computational resources, while the connections between the VNFs are mapped to the physical routes [5]. Some applications require VNF chains be executed within a predefined time constraint to address the application objectives. The constraint may be tight in a Tactile Internet application [6], or looser in a video streaming application [7]. For such applications, the resources should be allocated to the VNF chains such that the QoS in terms of constraints for chain execution time be met.

Another resource management issue is admission control, defined as a mechanism that a network provider applies to determine the admission of the VNF chains to the system. Most studies in the literature provide resource allocation solutions without considering admission control [4], [8]–[31]. However, admitting all the requests in a system is not always possible due to system resource limitations [32]–[34]. An appropriate mechanism is required to control the admission of chains.

A few studies have shown joint admission control and resource allocation can offer a significant improvement in the performance of NFV [35]–[43], among which few studies e.g., [40]–[43] consider time constraint satisfaction for VNF chains. These studies apply an admission decision mechanism that selects a subset of the input VNF chains to be admitted to the system. The resource allocation is performed for the admitted chains. VNF chain admission decision is performed based on the resource capacity to accommodate chains, chains' demands like required bandwidth, and QoS constraints, e.g., the time bound for chain execution.

The resource allocation solutions or joint admission control and resource allocation solutions available in the literature, model systems as a deterministic process [3], [4], [8]–[31], [35]–[44]. Under such circumstances, the traffic arrival amount to the chains and the processing rate of computational and communication nodes (i.e., VMs, switches) are assumed to be constant values. However, these assumptions do not hold in a wide range of network conditions due to dynamicity in traffic arrival, work load variation in a node (i.e., VM, switch), and competition on the usage of resources of a node [45], [46]. Furthermore, in the literature, the computational and transmission delays are calculated according to simplified deterministic models based on the constant traffic arrival amount and the constant processing/transmission rate which can involve low precision estimations in resource allocation procedure for a wide range of network conditions. This can end to low quality solutions and consequently time constraints violation at run time. However, stochastic modeling of a system gives a more realistic representation, in which the nonlinear and non-deterministic behaviour of the system is estimated by performance modeling analysis [47], [48]. This modeling shifts the time constraint satisfaction assessment from a binary space (satisfied or not satisfied) to a probabilistic space, i.e., determining the probability of the time constraint satisfaction. Such probability calculation, however, is not trivial.

In addition, an appropriate resource allocation is required to meet the tight deadlines of VNF chains. In our previous work [12], based on a deterministic modeling of the system, we have suggested exploiting parallel VNF processing when allocating resources to the chains. In this approach, the processing of individual flow is distributed among multiple VMs (performing the same VNF functionality) whenever the deadline is tight. This is in contrast to the existent studies, including [3], [4], [8]–[10], [13]–[29], and [36]–[44], where there is sequential traffic processing at the flow level, i.e., every flow is assigned to a single VM for a VNF functionality. In this paper, we extend the idea of parallel VNF processing in [12] by sharing a VM among multiple VNF chains when allocating resources. This will avoid waste of processing capacities of VMs (as a result of dedicating each VM to a chain as in [12]) and, it will enhance admission ratio in comparison with [12] as the results in Section VI show. Furthermore, we extend the idea in [12], by proposing a resource allocation mechanism based on a stochastic modeling for VNF chain execution, and providing an admission control mechanism.

Although parallelism brings gains in terms of traffic processing speed, it makes the communication model more complex because of the multiple routes the traffic may traverse. Accordingly, stochastic analysis of time constraint satisfaction becomes difficult because of the complexity of the communication model. This paper provides a joint admission control and resource allocation of time-constrained VNF chains while tackling the above-mentioned issues.

In an NFV scenario, power consumption due to resource utilization will bring electricity costs for network providers [49]. To provide an efficient resource allocation mechanism, it is important to meet the QoS in terms of time constraints for VNF chains while considering the power consumption. Indeed, appropriate resource allocation to a chain (i.e., degree of VNF parallelism and allocation of VMs) is required that considers the power consumption characteristics of the physical nodes. To address this aim, similar to [44], [50], we define the profit for the network provider based on the revenue obtained from the admission of chains and the cost imposed by power consumption as a result of resource allocation for the admitted chains. Next, the admission control and resource allocation is done such that the profit of the network provider is maximized. To maximize the network profit, the proposed method will also be efficient from the aspect of power consumption. This paper makes the following contributions:

1) Extending our idea of parallel VNF processing in [12] by sharing a VM among multiple VNF chains in resource allocation;

2) Stochastic modeling of VNF chain execution with a queue-network and analyzing it to calculate the expected value of the probability of deadline meeting in VNF chains;
3) Modeling of joint admission control and resource allocation for time-constrained VNF chains as an optimization problem. In the optimization, the profit of the network provider is maximized while the Confidence-Levels (CLs) for the deadline meeting of the admitted chains are met;
4) Proposing a heuristic for joint admission control and resource allocation for the VNF chains; and
5) Utilizing the stochastic analysis, and proposing a Tabu-based heuristic that exploits parallel VNF processing for allocating the substrate network resources to admitted chains.

This paper continues with the related works in Section 2. The system model is explained in Section 3. In Section 4, we model VNF chain execution with a queue network and provide its analysis. The optimization for joint VNF chain admission control and resource allocation is explained in Section 5, followed by the proposed heuristic to solve the optimization. Section 6 presents the performance evaluation, and finally, Section 7 gives the conclusion and future work.

II. RELATED WORK

We explain the related work in three categories: 1) resource allocation for VNF chains; 2) admission control; and 3) joint admission control and resource allocation, and indicate our contribution in each category.

A. RESOURCE ALLOCATION FOR VNF CHAINS

Generally, the studies in this category allocate resources i.e., computational resources and link bandwidth, to an input set of VNF chains without applying any admission control mechanism. Most studies model the problem as (Mixed) Integer Linear Programming (ILP) optimization; they optimize an objective function while considering some constraints: e.g., computational resource usage capacity, link usage capacity, constraints on chain's demands, and QoS constraints. They employ optimization tools/heuristics to solve the problem. We review these studies with a focus on objective functions.

Many studies have not considered time constraints for chain execution in their resource allocation solution and optimize various criteria including bandwidth utilization [13]; energy consumption [26]; NFV deployment cost [14], [15], [51], [52]; network congestion [16]; revenue [17]; resource usage [27], [53]; and amount of processed traffic [54]. These solutions may violate the required deadlines of chains.

There are studies that consider time constraints for chains while allocating substrate network resources. The works in [4], [18], [19] minimize the resource usage cost for VNF chain execution while meeting their deadlines. Resource utilization minimization is considered in [20]–[22]. The study in [23] minimizes network load cost. A resource allocation algorithm that minimizes the energy consumption in VNF chains while considering deadline satisfaction is presented in [24], [55]. The authors of [25] decide on the amount of resources needed for each VNF in the chain to satisfy the delay requirement while minimizing resource consumption. The authors of [28] consider resource allocation to VNF chains in a network consisting of hierarchical resources from edge to 5G core. They migrate VNFs among resources to avoid deadline violation, and minimize the migration frequencies. The study in [29] focuses on the consolidation of VNF instances while allocating resources to the chains. The allocation of resources to 5G network slices have been considered in [8]–[10]. Resource usage cost minimization for slices is considered in [8], [9]. The work in [10] considers the satisfaction of availability, reliability and delay tolerance requirements of the slices in resource allocation.

There are three main differences between studies in this category and our work: 1) We provide an admission control mechanism along with an resource allocation solution. Admission decisions are required when a system cannot meet the requirements of all the chains. 2) We analyze the system using stochastic modeling, while the above-mentioned works employ a deterministic analysis. 3) All the studies mentioned above process the traffic flow sequentially. Indeed, for each flow, a single VM serves the entire traffic for a VNF functionality. In contrast, we apply a parallel chain traffic processing to be able to meet tight time constraints. The works in [11], [12], [30], [31] use parallel traffic processing when allocating resources to the chains. However, they do not have an admission control mechanism. Furthermore, they are based on a deterministic system analysis which does not provide a precise representation of the system.

B. ADMISSION CONTROL

The studies in this category focus only on the admission control of VNF chains. The admission decision is based on the system's resource capacity and the constraints on chains' demands like required bandwidth or resources. In [32] the authors assume that the VNFs of the chain have already been deployed in the system. Their approach decides on the admission of flows passing through the VNFs. The problem is modeled as an ILP with the objective of revenue maximization. The study in [33] considers the admission of slices. Focused on the uncertainty of the resource demands of slices, they model the admission problem as a Markov Decision Process to admit a maximum amount of requests. The works in [32], [33] assume that infrastructure can provide the QoS requirement of VNF chains, e.g., time constraints for chain execution. However, meeting this requirement depends on resource allocation decisions. This correlation makes the admission decision a complicated task, which is the focus of this paper. The work in [34] decides on the packet admission at each VNF such that end-to-end latency is kept within a predefined deadline. However, it does not consider communication latency between VNFs. In comparison with the works

in this category, we consider joint resource allocation and admission control.

C. JOINT ADMISSION CONTROL AND RESOURCE ALLOCATION

The studies in this category consider both admission control and resource allocation for VNF chains. The authors of [36] maximize the number of chains admitted to the system and allocate resources to the admitted requests. In [3], researchers consider the relation between **L**ink resource usage and the **V**M reuse factor, called the LV relation. They propose a mechanism to obtain a chain LV relation such that the maximum number of chains can be admitted, admitting those chains for which the obtained LV relation holds. A game theoretic approach for admission and resource allocation to VNF chains has been presented in [56]. The works in [35], [37]–[39], [44] propose a joint optimization for admission control and resource allocation in an NFV environment. In [35] the focus is on maximizing the system revenue, while [44] maximizes the network profit. The study in [37] maximizes the number of chains admitted. In [38], a chain will bring revenue/penalty for admission/rejection. The aim is to maximize the system utility. Admission mechanisms for the network services in mobile edge computing are proposed in [39]. The aim is to maximize the revenue by admitting as many requests as possible while meeting their reliability requirements. The works in [3], [35]–[39], [44], [56] do not consider time constraints, either in resource allocation or admission decision. Thus, deadline violation is probable in these works.

The studies in [40]–[43], [57]–[59] consider time constraints. The authors of [43] utilize the migration of VNFs to serve the newly-arrived requests and the already-existent chains. They consider the end-to-end delay of the chains be less than the required deadline. The problem is modeled as an ILP to maximize the number of admitted requests while minimizing the migration cost. VNF remapping and rescheduling in space-air-ground integrated networks has been considered in [57]. The migration or instantiation of the VNFs on the network nodes in order to admit newly-arrived chain as well as serving the already-existent chains has been modeled as an optimization problem. The objective is to perform the admission and resource allocation such that the service provider profit be maximized while the chains deadlines are respected. Algorithm is proposed to obtain suboptimal solution. The authors of [41] maximizes an aggregation of the number of chains admitted to the system and the resource preference usage. The authors of [40] admit slices to the system such that system throughput is maximized. Dynamic VNF placement and routing mechanism to decide about admission and resource allocation of VNF chains in a NFV-enabled SDN environment, has been considered in [58]. The aim is to minimize resource consumption cost, while respecting QoS constraints including end-to-end delay, packet loss, and jitter. The study in [42] provide placement mechanisms of VNFs with the aim of increasing fault-tolerance by exploiting back-up VNF instances. They focus on maximizing the number of admitted chains such that the chains' deadlines are met while the deployment cost remains within a budget. The deployment cost is defined as the costs of VNF processing and traffic transmission to back-up VNFs. The authors of [59] provide resource allocation to admit maximum amount of services from IoT or mobile devices, to the edge computing resources while considering constraints for services response time. A linear response time modeling of edge resources at which the response time depends on the number of admitted requests has been utilized. A scaling mechanism for edge resources is proposed to adopt to the workload. However, the proposed method in [59] is applicable for services with a single VNF.

The methods proposed in [3], [36]–[44], [56]–[59] are based on sequential traffic processing, while we advocate for parallel traffic-chain processing that allows tight deadlines to be met. Furthermore, [3], [35]–[44], [56]–[58] consider a deterministic process, under assumptions that the traffic arrival and processing rate of computational resources is constant. This paper utilizes a stochastic modeling of the system, which provides a more realistic representation of the system.

III. SYSTEM MODEL

In this section, we explain the modeling of VNF chains, NFV infrastructure, communication, power consumption, decision variables, auxiliary notations, and assumptions. Table 1 indicates the used symbols.

A. VNF TYPES

Let $F = \{f_1, f_2 \ldots f_K\}$ be the ordered set of VNF types that any subset of which can be included in various VNF chains. Each VNF instance runs on a VM in order to process the traffic. The traffic size may be expanded, shrunk, or remain the same after VNF application. We define $\alpha_i > 0$ as the traffic scaling ratio of VNF type f_i. When $\alpha_i < 1$, shrinking will occur. When α_i is 1, the traffic size will not change. Finally, when $\alpha_i > 1$, the traffic will be expanded.

B. VNF CHAINS

Let CS be the set of M VNF chains. We define binary variable $y_{i,j}$ to be 1 when VNF type $f_i \in F$ is used by chain j. We also define $PD(i, j)$ as the set of VNF predecessors of VNF type f_i in chain j. Like [30], [60], we consider a Poisson traffic arrival pattern to the chain. The traffic for chain j is generated at source node o_j according to a Poisson process with a mean rate of λ_j chunks per second. When the traffic has been processed by all the VNFs defined in the chain, it will be forwarded to the destination node d_j. Let CL_j be the Confidence Level (CL) for meeting deadlines for the traffic chunks of chain j. Indeed, chunks of traffic that come from the source should be processed through the VNFs and reach the destination according to the specified deadline and CL. For example, a CL of 0.98 means that the processing of at least 98% of the traffic chunks should be completed by the predefined deadline. In the application of Tactile Internet as

TABLE 1. Symbols.

Symbol	Description
K	Number of VNF types
f_i	VNF type $\boldsymbol{i}$
α_i	Traffic scaling ratio of VNF f_i
CS	The set of VNF chains including M chains
$\boldsymbol{\lambda_j}$	The mean traffic rate that goes through chain j
o_j	Source node of VNF chain j
$\boldsymbol{d_j}$	Destination node of chain j
D_j	Deadline requirement for chain j
CL_j	Confidence level for meeting deadline D_j
V	Set of NVFI nodes including physical servers and switches
E	Communication edges among NFVI nodes
PS	Set of physical servers
SW	Set of switches
PC	Set of switches connecting the pools
Pl_i	Pool of VMs for function f_i
I_i	Number of VMs in pool Pl_i
v_n^i	VM $\boldsymbol{n}$ in Pl_i
H_n^i	A $1 \times N$ vector defining the host of $\boldsymbol{v_n^i}$
μ_n^i	The mean traffic processing speed of $\boldsymbol{v_n^i}$
λ_n^i	Traffic arrival rate to $\boldsymbol{v_n^i}$
$SS^{i-1,i}$	Software-defined switch connecting Pl_{i-1} to Pl_i
$\mu^{i-1,i}$	The mean traffic processing speed of $SS^{i-1,i}$
μ^{sw}	The mean traffic processing speed of an arbitrary switch sw
$p_n^{i,j}$	The portion of input from chain j that goes toward $\boldsymbol{v_n^i}$
σ_i^j	Traffic arrival rate to Pl_i for chain j
$\aleph_j$	the revenue gain for chain j when it is admitted
e_v	Power usage effectiveness of device v
β_v^S	Static power consumption of device v
β_v^D	Dynamic power consumption of device v
$y_{i,j}$	Binary variable indicating if f_i is used by chain j
$x_n^{i,j}$	Binary variable indicating if v_n^i is used for by chain j
ζ_j	Binary variable indicating if chain $\boldsymbol{j}$ is admitted
ip	Index for immediate VNF predecessor of VNF type $\boldsymbol{i}$
$PT(v_m^{ip}, v_n^i)$	The shortest path between the two VMs of v_m^{ip} and v_n^i

an example, in remote orthopedic surgeries which are already being conducted in 5G [61], artificial intelligence could be used to predict the contents of traffic chunks that do not reach their destination in time [62]. Thus, for the orthopedic surgeries application, a value of less than 100% can be set for the CL, to impose less cost to the user. In contrast, CL might be selected as 100% for applications like remote heart surgery, in which predicting the content of the delayed chunk is difficult and so all the chunks need to reach within the specified deadline [62]. Meanwhile, in applications like VoIP data transmission, video conferencing, and streaming media (with looser deadline in comparison with Tactile Internet), the CL might be selected as less than 100%, since the delayed traffic chunks can be dropped and the application can tolerate the loss of some traffic chunks, up to a threshold that will still assure an acceptable QoS [63]. The deadline profiles for all chains and their CL profiles are represented by vectors $\boldsymbol{D}$ and $\boldsymbol{CL}$, respectively i.e., every chain has a predefined deadline and CL. We define $\aleph_j$ as the revenue gain for chain j in the case when the chain is admitted to the system.

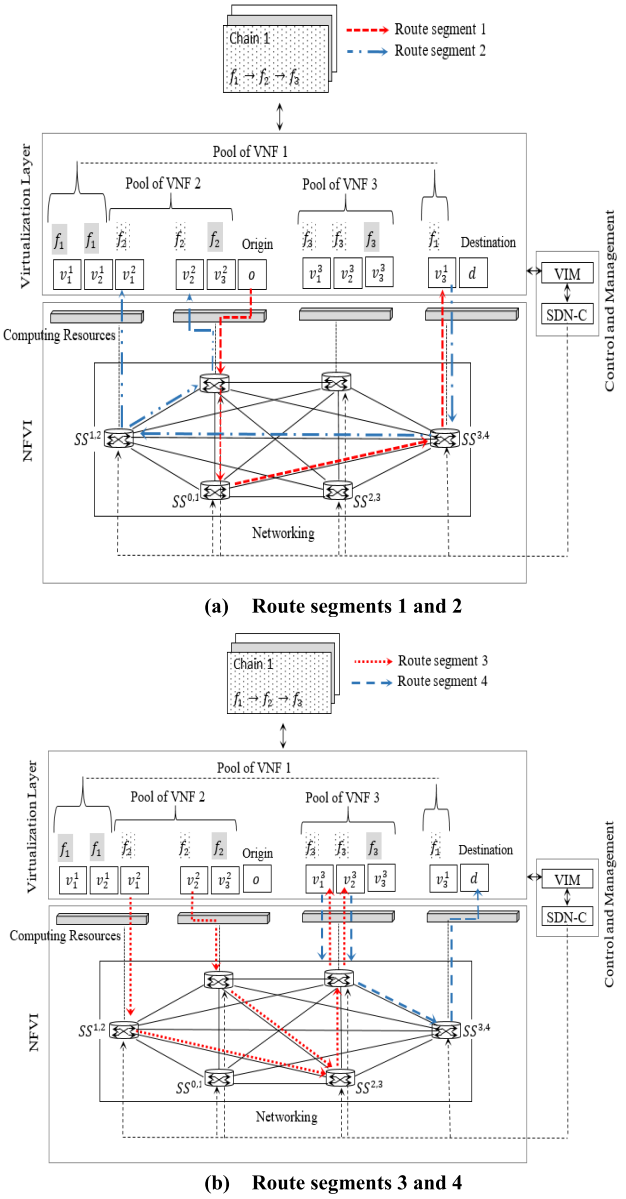

FIGURE 1. The concept of parallel processing of VNFs. There are three VNF types, and there is a pool of VMs for each. The VMs have been allocated to two chains one with dotted the other with the grey-shadow. The blue flow shows the data path to enable parallel processing for the dotted chain.

C. NFV INFRASTRUCTURE

NFVI is modeled as a graph $G = (V, \boldsymbol{E})$. V consists of all nodes, including N physical servers represented by PS and $|SW|$ software-defined switches represented by SW; i.e., $V = PS \cup SW$. Vector $\boldsymbol{E}$ is a matrix of size $(N+|SW|)\times(N+|SW|)$ that indicates the connectivity between nodes. Here, element $e_{ij} \in E$ is 1 when node $i \in V$ is directly connected to node $j \in V$. Communications between physical servers are carried out by a network of software-defined switches (see Fig. 1). Each physical server is connected to one arbitrary switch. Thus, for server $n \in PS$ we have: $\sum_{sw \in SW} e_{n,sw} = 1$. Note that this modeling of connectivity, let pool of servers be connected to a single switch.

Like [7], [64], [65], we assume there are preinstalled VNF instances for each VNF type, where each VNF instance is running in a VM. Indeed, there is a pool of VMs for each VNF type. All VMs in a pool have the hardware/software required to execute the VNF. To have an efficient VM utilization, like [3], [66]–[68] we assume different VNF chains that demand the same VNF type can share the same VM running that VNF type. Security mechanisms like threat detection by monitoring the status of the chain running over the VM [69] or applying security policies for the chains through VNFs connected to the VM [70] can be adopted to enhance the security of VNF chains under VM sharing circumstances. Also like [3], we assume that each VM can run at most one VNF type. Physical servers host the VMs. Note that the VMs in a single pool can be distributed over several servers. Pool i consists of $I_i \geq 1$ VMs, capable of hosting $f_i \in F$ in parallel. Pool i is represented by set $Pl_i = \left\{v_1^i, v_2^i \dots v_{I_i}^i\right\}$. Here, v_n^i ($n \in \{1, \dots I_i\}$) is the VM with index n inside Pl_i. For VM v_n^i, we define a $1 \times N$ vector $\boldsymbol{H_n^i}$ with the element of 1 for the physical server hosting v_n^i, while the other elements are 0. Like [46], [71] we model each VM as an M/M/1 FCFS (First Come First Service) queue. VM v_n^i processes the arrived traffic at a speed with exponential distribution, with a mean rate of μ_n^i chunks per second, which is equivalent to the VM processing capacity. In this regard, the VMs within a pool are heterogeneous from the aspect of traffic processing speed.

D. COMMUNICATION

Switches are programmed by the SDN-Controller to distribute traffic among VMs. For simplicity, let us focus on a chain j that uses all VNF types. There are two end-point communications: First, the traffic which is generated at source node o_j arrives into a switch, $SS^{0,1}$, which forwards the traffic towards the VMs in the first pool i.e., Pl_1. Second, the traffic processed by the VMs of the last pool Pl_K are transmitted by a switch denoted by $SS^{K,K+1}$ towards the destination d_j. Like [45], we model switches as M/M/1 FCFS queues with exponential transmission with mean rates of $\mu^{0,1}$ and $\mu^{K,K+1}$ chunks per second, respectively. The communication between two adjacent pools of Pl_{i-1} and Pl_i ($1 \leq i \leq K$) is done by switch $SS^{i-1,i}$ with a mean transmission rate of $\mu^{i-1,i}$ chunks per second. We represent the switches providing connection between pools with the set $PC = \left\{SS^{i-1,i} \,|\, i = 1 \dots K+1\right\} \subseteq SW$. The switch $SS^{i-1,i}$ operates according to a probabilistic transmission strategy. It sends each traffic chunk belonging to chain j to the VM v_n^i with probability $p_n^{i,j}$. The whole distribution policy is represented by vector $\boldsymbol{p}$. Direct transmission of traffic chunks among VMs hosted on the same server is possible. In this case the traffic does not need to go through a switch. To reduce latency, the switch, in which a maximum number of shortest paths between every pair of VMs in pools Pl_{i-1}, and Pl_i crosses, is labeled as $SS^{i-1,i}$. We refer to the mean transmission rate of an arbitrary switch in the network $sw \in SW$ with μ^{sw}.

E. POWER CONSUMPTION

We use the power consumption model of [72], in which the power consumption of an electronic device depends on the Power Usage Effectiveness (PUE) as well as the static (leakage) and dynamic power consumption. Here, a device can be a physical server or a switch. Static power consumption is caused by current leakage and is unrelated to the device usage. Dynamic power consumption is caused by device circuits' activities and thus is determined by the device utilization. For a physical server, the utilization of the capacity of traffic processing is used, while for a switch the utilization of capacity of traffic transmission is considered. Let e_v be the PUE of device $v \epsilon V$. The static power consumption of the device is represented by β_v^S. Let β_v^D be the dynamic power consumption of the device when the utilization is maximum (all the processing/transmission capacity of the physical server/switch is utilized). The power consumption of the device is calculated as (1). Here, ρ_v is the utilization of device v.

$$E = e_v(\beta_v^S + \beta_v^D.\rho_v). \tag{1}$$

F. DECISION VARIABLES

Depending on the deadline and the CL, a subset of VMs inside a pool is allocated to each chain. The allocation profile is represented by vector $\boldsymbol{X}$ with binary variables $x_n^{i,j}$; it has the value of 1 when VM $v_n^i \epsilon Pl_i$ serves VNF type f_i to chain j. We also define ζ_j as the decision variable defining if chain j is admitted to the system (1 for the case of admission and 0 for the case of rejection).

G. INDICES AND AUXILIARY NOTATIONS

We introduce several auxiliary notations in this paper. |X| is used for the size of set X. Notation $1.(X)$ or $1[X]$ is 1 when boolean X is true, otherwise it is 0. The symbol $\mathbf{0}_{1\times N}$ is a vector of dimensions $1 \times$ N with all elements of 0. Symbol $\mathbf{0_m}$ is a vector of $1 \times N$ with all elements of 0 except the m^{th} element, which is 1. We also use some indices in this paper. We use index i for pool, index n for VM, and j for VNF chain. We also introduce ip to denote immediate a VNF predecessor of VNF type i. Note that ip is used in the context of a chain. For example, for a chain that uses VNF types 2, 5, 7, when i is 5, then ip is 2; and when i is 7, then ip has the value of 5.

H. ASSUMPTIONS

These are the assumptions used in this paper:

a) To enable parallel VNF processing, we assume there are at least K + 1 switches, i.e., $|SW| \geq K+1$. Considering that the number of VNF types is rather small, in real networks, the number of switches is commonly more than $K+1$.

b) The traffic is distributed among VMs in the pool, in proportional to their processing speed. In this regard, the probability that a traffic chunk of chain j goes through v_n^i is calculated using (2). Note that the proposed method is applicable to any other policy as well. From the aspect of

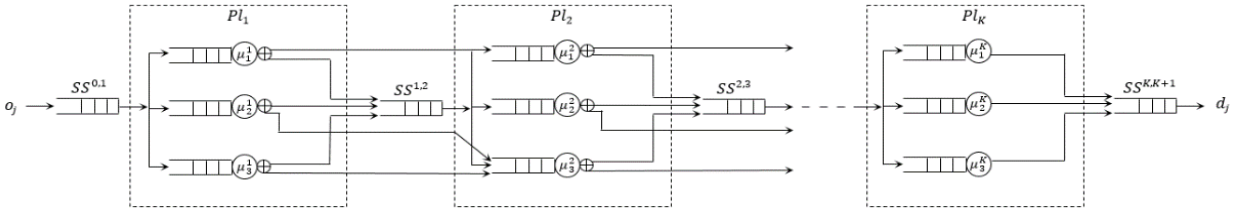

FIGURE 2. Queue network for a chain execution that uses all K VNF types. The XORs denote the possible routing of traffic.

implementing the traffic distribution, programmable data plane technology provides facilities through which the switches can be programmed to specifically process the packets of an application i.e., VNF chain. One of the most renowned architectures for programmable switches, i.e. Protocol Independent Switch Architecture (PISA) [73], provides programmability of switches through match-action tables. The switches inter-connecting the pools can be programmed such that matching recognize the VNF chain packets, while action performs the probabilistic routing in (2), which is a load balancing strategy. Note that implementing load balancing in programmable switches has been shown to be feasible in [74], [75]. As the hardware of programmable switches can provide a line-rate of processing [73], it is expected that the splitting traffic in switches, has ignorable impact on VNF chain execution latency.

$$p_n^{i,j} = \frac{\mu_n^i x_n^{i,j}}{\sum_{k=1}^{I_i} \mu_k^i x_k^{i,j}}. \tag{2}$$

c) Similar to [30], for the sake of simplicity we advocate the benefit of using the shortest path routing (from the aspect of latency), and we assume that traffic is transmitted from any physical server to any switch $SS^{i-1,i} \in PC$, using the shortest path between them. Similarly, the traffic is transmitted from any switch $SS^{i-1,i} \in PC$ to any physical server via the shortest path between them. In Section V.C we explain how the proposed method can be generalized to decide about routing.

d) The probabilistic transmission used in switches connecting the pools, might alter the order of packets at destination. Though there exist efficient reordering mechanisms that can be applied at destination-side to deliver the packets in order [76], [77], appropriate value for size of traffic chunk will further diminish the reordering overhead. Indeed, as each traffic chunk can include multiple packets, the order of packets inside each chunk is kept and does not require reordering. In this paper, we assume that size of chunk has been chosen appropriately so that the reordering will be done with tolerable overhead.

Example: To clarify the VNF parallel processing, Fig. 1 shows an example. The network consists of four physical servers that are connected through six switches. There are three VNF types, and accordingly, three pools. Each pool has three VMs. A sample allocation is shown for two chains which need all VNF types. From the VNF1 pool, two and one VMs are allocated to the grey-shaded and the dotted chains, respectively. From the VNF2 (VNF3) pool, two and one VMs are allocated to the dotted and the grey-shaded chains, respectively. We focus on the dotted chain for the communication pattern. A possible routing has been illustrated. Here, VM o on the second and VM d on the last physical server are the origin and destination of the traffic, respectively, and the traffic will traverse the VNFs in the sequence of f_1, f_2, f_3. Four switches have been labeled to provide connections among the pools. The whole route consists of four route segments. A VNF type is visited at each route segment. Route segments 1 and 2 are shown in Fig. 1(a) while route segments 3 and 4 are shown in Fig. 1(b). Traffic from the origin node comes to $SS^{0,1}$, which sends the traffic to v_3^1 (route segment 1). The processed traffic is then forwarded to $SS^{1,2}$, which forwards a fraction of the traffic to v_1^2 and the rest to v_2^2 (route segment 2). The processed traffic is next sent to $SS^{2,3}$, which sends a fraction of the traffic to v_1^3 and the rest to v_2^3 (route segment 3). Finally, the traffic is routed towards $SS^{3,4}$, from where it goes to the destination (route segment 4). The SDN controller program switches to provide such routing among the pools.

IV. VNF CHAIN EXECUTION ANALYSIS

In this section, we first explain how VNF chain execution can be modeled by a queue network. Next, we provide the analysis.

A. MODELING CHAIN EXECUTION WITH A QUEUE NETWORK

We can model the VNF chain execution as a queue network. Fig. 2 shows the queue network when the traffic traverses all K VNF types. In this model, each queue inside a pool illustrates a VM, and the queues between pools illustrate the switches. The traffic chunks traverse the VNFs from source to destination. Each switch probabilistically transmits each chunk toward a VM inside the next pool to apply the VNF.

As traffic might traverse several switches before reaching a switch like $SS^{i-1,i}$ (See Fig. 1), the effective transmission rate of $SS^{i-1,i}$ namely, $\hat{\mu}^{i-1,i}$ is reduced to the transmission speed of the slowest switch on the path. Eq. (3) shows the calculation. Here, $PT(v_m^{ip}, v_n^i)$ indicates the shortest path between the two VMs of $v_m^{ip} \in Pl_{ip}$ and $v_n^i \in Pl_i$ that

traverses $SS^{i-1,i}$. In (3), $\frac{p_n^{i,j}}{\sum_n p_n^{i,j}} . \frac{p_m^{ip,j}}{\sum_m p_m^{ip,j}}$ represents the probability of traffic transmission from $v_m^{ip} \in Pl_{ip}$ to $v_n^i \in Pl_i$. In the case that traffic is transmitted from v_m^{ip} to v_n^i, the minimum transmission rate of the switches on the shortest path from v_m^{ip} to v_n^i i.e., $\min_{SW \in PT(v_m^{ip}, v_n^i)} \mu^{sw}$ is involved in the calculation. Note that the minimum transmission rate of the switches on the path is equivalent to the bandwidth capacity of the path connecting the two VMs. Similarly, the effective mean transmission rates of $SS^{0,1}$ and $SS^{K,K+1}$ are calculated by (4) and (5), respectively. For the $SS^{0,1}$ switch, the shortest path from the source nodes to the switch is considered. For switch $SS^{K,K+1}$ the shortest path from the switch to the destination nodes is considered.

$$\hat{\mu}^{i-1,i} = \frac{1}{M} . \sum_{j \in CS} \sum_{n=1}^{I_i} \sum_{m=1}^{I_{ip}} \times \frac{p_n^{i,j}}{\sum_n p_n^{i,j}} . \frac{p_m^{ip,j}}{\sum_m p_m^{ip,j}} . \min_{SW \in PT(v_m^{ip}, v_n^i)} \mu^{sw}, \quad (3)$$

$$\hat{\mu}^{0,1} = \frac{\sum_{j \in CS} \min_{SW \in PT(o_j, SS^{0,1})} \mu^{sw}}{M}, \quad (4)$$

$$\hat{\mu}^{K,K+1} = \frac{\sum_{j \in CS} \min_{SW \in PT(SS^{K,K+1}, d_j)} \mu^{sw}}{M}. \quad (5)$$

As we see in Fig. 2, some XORs are appended just after a VM queue. This indicates the possible routes of chunks inside the queue network. After a chunk is processed by a VNF instance, it may go directly to the next VNF instance (in the succeeding pool); this happens when both VMs hosting the VNF instances are located on the same server. Otherwise, the traffic goes to the switch connecting the two pools.

The traffic exit rate from Pl_i for chain j is calculated using (6). Indeed, the original traffic rate of the chain is multiplied by the traffic scaling ratios of all the predecessor VNFs of f_i.

$$\sigma_i^j = \lambda_j . \alpha_i . \prod_{f \in PD(i,j)} \alpha_f . \quad (6)$$

B. QUEUE NETWORK ANALYSIS

In this subsection, we analyze the queue network. Before proceeding to our analysis, we define $\lambda_n^{i,j}$ as the arrival traffic rate to VM v_n^i from chain j, and define $\lambda^{i-1,i,j}$ as the arrival traffic rate to switch $SS^{i-1,i}$ from chain j.

To calculate the rate $\lambda_n^{i,j}$, we introduce the variable Col_n^i with the value of 1 only when v_n^i has been co-located with any of the VMs in the immediate predecessor VNF pool; i.e., Pl_{ip}. Eq. (7-1) shows the Col_n^i calculation. Every individual v_m^{ip} is checked to see if it has the same host as v_n^i or not. This will be verified by a logical statement $\left(\mathbf{H_n^i} - \mathbf{H_m^{ip}} == \mathbf{0}_{1\times N}\right)$. In this regard, $\sum_{m=1}^{I_{ip}} 1.\left(\mathbf{H_n^i} - \mathbf{H_m^{ip}} == \mathbf{0}_{1\times N}\right)$ counts the number of VMs in the immediate predecessor VNF pool that have been co-located with v_n^i. In the case where the count is above zero, co-location has occurred and Col_n^i takes the value of 1.

The rate $\lambda_n^{i,j}$ is calculated as given in (7-2). The first relation is for the case when co-location has not occurred; i.e., $Col_n^i = 0$. In this case, the source of the traffic arrival is switch $SS^{i-1,i}$, which sends a fraction of its arrival rate to v_n^i i.e. $\lambda^{i-1,i,j} . p_n^{i,j}$.

The second relation in (7-2) is for the case where co-location has occurred. In this case, a fraction of the traffic comes to v_n^i via the switch $SS^{i-1,i}$, and a fraction of traffic comes directly from the immediate predecessor pool. The first and second terms in the relation calculate the specified amounts of traffic. In the second term, the numerator is the total traffic that comes from the VMs in Pl_{ip} that have been co-located with v_n^i. The denominator indicates the number of VMs allocated to the chain in Pl_i that have been co-located with v_n^i. Indeed, the traffic is divided among these VMs and v_n^i.

$$Col_n^i = 1[\sum_{m=1}^{I_{ip}} 1.\left(\mathbf{H_n^i} - \mathbf{H_m^{ip}} == \mathbf{0}_{1\times N}\right) > 0], \quad (7\text{-}1)$$

$$\lambda_n^{i,j} = \begin{cases} x_n^{i,j} . \lambda^{i-1,i,j} . p_n^{i,j} & Col_n^i = 0 \\ x_n^{i,j} [\lambda^{i-1,i,j} . p_n^{i,j} & Col_n^i = 1 \\ + \frac{\sum_{m=1}^{I_{ip}} 1.\left(\mathbf{H_n^i} - \mathbf{H_m^{ip}} == \mathbf{0}_{1\times N}\right) . x_m^{ip,j} . \alpha_{ip} . \lambda_m^{ip,j}}{\sum_{k=1}^{I_i} x_k^{i,j} . 1.\left(\mathbf{H_n^i} - \mathbf{H_k^i} == \mathbf{0}_{1\times N}\right)}]. \end{cases} \quad (7\text{-}2)$$

The arrival traffic rate to VM v_n^i is calculated as the summation of the traffic arrival rates from all chains. Eq. (8) shows the calculation. Here, we have added a traffic rate, $\lambda_n^{i,back}$, which indicates any background traffic that might come into VM v_n^i because of already-admitted chains to the system that have been assigned to the VM.

$$\lambda_n^i = \lambda_n^{i,back} + \sum_{j \epsilon CS} \lambda_n^{i,j} . \quad (8)$$

Let $\omega_{ip \to i}^j$ be the traffic amount of chain j that comes directly from pool Pl_{ip} to pool Pl_i without passing the intermediate switch. To calculate $\omega_{ip \to i}^j$, we define the variable Ct_m^{ip} that counts the number of VMs in pool Pl_i that have been assigned to the chain j and co-located with VM v_m^{ip}. Eq. (9-1) shows the calculation of Ct_m^{ip}. The traffic $\omega_{ip \to i}^j$ is calculated by (9-2). When $Ct_m^{ip} > 0$, the traffic amount of $\alpha_{ip} . \lambda_m^{ip,j}$ goes directly from VM v_m^{ip} to pool Pl_i without passing the intermediate switch. These amounts of traffic are summed over all VMs in pool Pl_{ip}.

$$Ct_m^{ip} = \sum_{n=1}^{I_i} x_n^{i,j} . 1.\left(\mathbf{H_n^i} - \mathbf{H_m^{ip}} == \mathbf{0}_{1\times N}\right), \quad (9\text{-}1)$$

$$\omega_{ip \to i}^j = \sum_{m=1}^{I_{ip}} x_m^{ip,j} . \alpha_{ip} . \lambda_m^{ip,j} . 1.\left(Ct_m^{ip} > 0\right). \quad (9\text{-}2)$$

The rate $\lambda^{i-1,i,j}$ is calculated as all of the traffic that leaves from the predecessor pool in the chain, i.e., Pl_{ip}, minus the

traffic that enters directly into the pool Pl_i without passing through the switch $SS^{i-1,i}$. Eq. (10) shows the calculation. Equations (11) and (12) show the calculation of the traffic arrival rate to the entry and exit switches, respectively. Here, $FV(j)$ indicates the first VNF in chain j. Similarly, $LV(j)$ represents the last function.

$$\lambda^{i-1,i,j} = y_{i,j}.\left(\sigma^{j}_{ip} - \omega^{j}_{ip\rightarrow i}\right), \tag{10}$$

$$\lambda^{FV(j)-1,FV(j),j} = \lambda^{j}, \tag{11}$$

$$\lambda^{LV(j),LV(j)+1,j} = \sigma^{j}_{LV(j)}. \tag{12}$$

The arrival traffic rate to switch $SS^{i-1,i}$ is the summation of the traffic rates for the chains and the background traffic rate of the switch. The background traffic rate includes the rate of traffic imposed to switch from the already admitted chains in the system that utilize switch $SS^{i-1,i}$ for communication among their VNFs. Eq. (13) shows the calculation.

$$\lambda^{i-1,i} = \lambda^{i-1,i,back} + \sum_{j\in CS} \lambda^{i-1,i,j}. \tag{13}$$

Let $f_{T_j}(t)$ be the probability density function for the response time of chain j, i.e., $f_{T_j}(t) = \Pr(T_j == t)$. The Probability of Deadline Meeting (PDM) for the chain j is calculated as below:

$$PDM_j(\boldsymbol{X}) = \int_0^{D_j} f_{T_j}(t)dt. \tag{14}$$

Now we explain the calculation of the expected value of the PDM. Without loss of generality, let us focus on a chain j that uses all VNF types. A chunk of such a chain may go through a path of $ss^{0,1}$, $v^1_{n_1}$, $ss^{1,2}$, $v^2_{n_2}$,$ss^{2,3}$, $\ldots$ $v^K_{n_K}$,$ss^{K,K+1}$ to be processed by the VNFs. The probability of being routed through such a path is calculated as $\prod_{i=1}^{K} p^{i,j}_{n_i}$. The path is a Tandem queue network. Let $T^i_{n_i}$ and $T^{i,i+1}$ represent the sojourn time in VM $v^i_{n_i}$ and switch $ss^{i,i+1}$, respectively. In a Tandem network, i.e., a series of M/M/1 queues with FCFS policy, the sojourn times of a given chunk of data in each queue are independent [71], [78]–[80].[1] Let T_j be the sojourn time of the whole path. We have:

$$T_j = \sum_{i=1}^{K} T^i_{n_i} + \sum_{i=0}^{K} T^{i,i+1}. \tag{15}$$

The sojourn time in the VMs and the switches (M/M/1 queues) obey an exponential distribution, i.e., $T^i_{n_i} \sim exp(\mu^i_{n_i} - \lambda^i_{n_i})$ and $T^{i,i+1} \sim exp\left(\mu^{i-1,i} - \lambda^{i-1,i}\right)$ [78]. The following $2K+1$ variables are defined:

$$\begin{aligned}
\beta_1 &= \mu^{0,1} - \lambda^{0,1} \\
\beta_2 &= \mu^1_{n_1} - \lambda^1_{n_1} \\
\beta_3 &= \mu^{1,2} - \lambda^{1,2} \\
\beta_4 &= \mu^2_{n_2} - \lambda^2_{n_2} \\
&\vdots \\
\beta_{2K+1} &= \mu^{K,K+1} - \lambda^{K,K+1}.
\end{aligned} \tag{16}$$

Relying on [71], [84], the distribution of T_j is calculated as shown below:

$$G_{T_j}(t) = Pr\left(T_j \le t\right) = \sum_{i=1}^{2K+1} \prod_{j=1,j\neq i}^{2K+1} \frac{\beta_j}{\beta_j - \beta_i}(1-e^{-\beta_i t}). \tag{17}$$

The expected value of the PDM is calculated as (18). Here, $I_{1\ldots K} = \{(n_1 \ldots n_K) \mid n_1 \in \{1 \ldots I_1\}, \ldots, n_K \in (1 \ldots I_K)\}$. Note that (16), (17) and (18) can simply be adjusted for arbitrary paths of any length; all that is required is to include the VM/switch terms that are used in the paths into these equations.

$$\overline{PDM}_j(\boldsymbol{X}) = \sum_{(n_1,\ldots n_K)\in I_{1..K}} \prod_{i=1}^{K} p^{i,j}_{n_i}.G_{T_j}\left(D_j\right). \tag{18}$$

V. OPTIMIZATION FOR JOINT ADMISSION CONTROL AND RESOURCE ALLOCATION FOR VNF CHAINS

In this section, we first give the optimization model for the joint admission control and resource allocation for VNF chains. Next, we propose a heuristic to solve the problem, and then analyze its complexity.

A. OPTIMIZATION MODEL

The optimization objective is to admit chains of VNFs such that the profit of the network provider is maximized. Furthermore, the confidence level for meeting the deadlines of the admitted chains should be kept according to the Service Level Agreement (SLA) requirement. The optimization problem is defined as given below, where (19) defines maximizing the network provider profit as the objective function. Equations (20)-(24) are elements involved in the objective function calculation, and (25)-(32) are the constraints.

$$max \sum_{j\epsilon CS} \zeta_j.\aleph_j - \Gamma.E(\boldsymbol{X}), \tag{19}$$

$$\begin{aligned}
E(\boldsymbol{X}) &= \sum_{ps=1}^{N} e_{ps}(\beta^S_{ps} + \beta^D_{ps}.\rho_{ps}) \\
&+ \sum_{i=1}^{K} e_{i-1,i}(\beta^S_{i-1,i} + \beta^D_{i-1,i}.\rho_{i-1,i}) \\
&+ \sum_{sw\in SW\setminus PC} e_{sw}(\beta^S_{sw} + \beta^D_{sw}.\rho_{sw}),
\end{aligned} \tag{20}$$

$$\rho_{ps} = \frac{\sum_{i=1}^{K}\sum_{l=1}^{I_i} \lambda^i_l.1.(\boldsymbol{H^i_l} - \boldsymbol{0_{ps}} == \boldsymbol{0_{1\times N}})}{\sum_{i=1}^{K}\sum_{l=1}^{I_i} \mu^i_l.1.(\boldsymbol{H^i_l} - \boldsymbol{0_{ps}} == \boldsymbol{0_{1\times N}})}, \tag{21}$$

$$\rho_{i-1,i} = \frac{\lambda^{i-1,i}}{\hat{\mu}^{i-1,i}}, \tag{22}$$

[1] For the sake of simplicity in analysis, like [46], [71], [45] we assume traffic arrival/processing in VMs and switches follow M/M/1 queueing model. For the case of general traffic arrival/processing in VMs and switches, G/G/1 queueing model can be applied and the response time analysis of a series of G/G/1 queues [81]–[83] will be involved in the calculation of (17).

$$\lambda^{sw}(\boldsymbol{X}) = \lambda^{sw,back} + \sum_{j\in CS}\sum_{\{ip|y_{ip,j}==1\}}\sum_{m=1}^{I_{ip}}\sum_{n=1}^{I_i} x_m^{ip,j}.x_n^{i,j}.z\left(sw, v_m^{ip}, v_n^i\right).1.\left(\mathbf{H}_{\boldsymbol{n}}^{\boldsymbol{i}} - \mathbf{H}_{\boldsymbol{m}}^{\boldsymbol{ip}} \neq \mathbf{0}_{1\times N}\right).\lambda_m^{ip,j}.\alpha_{ip}.p_n^{i,j}, \quad (23)$$

$$\rho_{sw} = \frac{\lambda^{sw}}{\mu^{sw}}. \quad \forall sw \in SW\backslash PC \quad (24)$$

Subject to:

$$\zeta_j.x_n^{i,j} = x_n^{i,j}, \quad \forall j \in CS,\ i:1\ldots K,\ n:1\ldots I_i \quad (25)$$

$$y_{i,j}.\sum_{n=1}^{I_i} x_n^{i,j} = \sum_{n=1}^{I_i} x_n^{i,j}, \quad \forall j\in CS,\ i:1\ldots K \quad (26)$$

$$\zeta_j.y_{i,j} \leq \sum_{n=1}^{I_i} x_n^{i,j}, \quad \forall j \in CS,\ i:1\ldots K \quad (27)$$

$$\sum_{n=1}^{I_i} x_n^{i,j} \leq I_n \quad \forall j\in CS,\ i:1\ldots K \quad (28)$$

$$\lambda^{i-1,i}(\boldsymbol{X}) < \hat{\mu}^{i-1,i}, \quad \forall i:1\ldots K+1 \quad (29)$$

$$\lambda^{sw}(\boldsymbol{X}) < \mu^{sw}, \quad \forall sw \in SW\backslash PC \quad (30)$$

$$\lambda_n^i(\boldsymbol{X}) < \mu_n^i, \quad \forall i:1\ldots K,\ n:1\ldots I_i \quad (31)$$

$$\overline{PDM}_j(\boldsymbol{X}) \geq \zeta_j.CL_j, \quad \forall j \in CS \quad (32)$$

Following the general calculation of profit as utility minus cost, we consider system revenue as utility and power consumption as the cost, similar to [45] and [49]. The network provider profit in (19) is defined as the amount remaining after the cost that the network provider should pay for power consumption has been deducted from the revenue the provider receives for giving service to the VNF chains. Γ is the coefficient utilized to convert the power consumption to the monetary term.

The power consumption of the system is calculated as the summation of the power consumption in the physical servers and switches, as shown in (20). Here, $e_{i-1,i}$, $\beta_{i-1,i}^S$, and $\beta_{i-1,i}^D$ are the PUE, static power consumption, and dynamic power consumption, respectively, of switch $SS^{i-1,i}$. ρ_{ps} is the utilization of physical server ps, calculated by (21). Indeed, the server can be viewed as a composite computational resource composed of several VMs. Server utilization is calculated as the ratio of the total traffic arrival rate to the server to the total traffic processing rate. Similarly, (22) illustrates the utilization of switch $SS_{i-1,i}$.

Eq. (23) calculates the mean arrival traffic to other switches in the network. Here, $\lambda^{sw,back}$ is the background traffic to the switch due to already admitted chains. Index i is the immediate successor pool of ip. The notation $z\left(sw, v_m^{ip}, v_n^i\right)$ is a binary variable with the value of 1 when switch $sw \in SW\backslash PC$ is on the shortest path from v_m^{ip} to v_n^i which traverses $SS^{i-1,i}$ i.e., $sw \in PT(v_m^{ip}, v_n^i)$. For every chain like j, when the switch is on the shortest path from any allocated VM in pool ip (like v_m^{ip}), to any allocated VM in the successor pool i (like v_n^i), traffic amount of $\lambda_m^{ip,j}.\alpha_{ip}.p_n^{i,j}$ will pass through the switch. The co-location issue has also been considered in calculations of (23). Eq. (24) shows the switch utilization calculation.

Constraint (25) ensures that only the admitted chains will be assigned to VMs. According to (26), when a chain does not use a VNF type, no instances of that VNF type are allocated to the chain. Constraint (27) ensures that at least one instance of a VNF type (one VM in the associated pool) is assigned to the chain when it needs that VNF type and is admitted to the system. Constraint (28) ensures that the number of VMs for a specific VNF type that are assigned to every VNF chain are bounded with the number of VMs inside the pool. Constraints (29), (30), and (31) ensure the ergodicity conditions in the queue network. Here, (29) and (30) ensure the admission of traffic within the capacity of the transmission rates of the switches, thereby avoiding congestion in the network (equivalent to a bandwidth capacity constraint). Constraint (31) is the VM processing capacity constraint which indicates that the arrived traffic to a VM should be less than its processing capacity. Constraint (32) ensures the deadline meeting confidence level for the admitted chains. Note that the $\overline{PDM}$ is calculated by (18).

B. PROPOSED HEURISTIC FOR JOINT ADMISSION CONTROL AND RESOURCE ALLOCATION FOR VNF CHAINS

The optimization problem formulated in subsection V.A is a binary nonlinear optimization (see equations (7-2) and (17), which are nonlinear); given the complexity of non- linear solvers, heuristic is required to efficiently solve the problem. We solve the optimization problem, i.e., maximizing the network provider profit in (19), by proposing the heuristic ACRA, which is abbreviation for **A**dmission **C**ontrol and **R**esource **A**llocation. ACRA iteratively calls another heuristic, RA, which is abbreviation for **R**esource **A**llocation. RA does not decide on the admission of chains; instead, it allocates resources to the maximum number of chains with minimum power usage, a required step to for optimize the objective function (19). ACRA utilizes RA to allocation resources, and furthermore it provides control over: 1) the admission of highly profitable chains to maximize network provider profit; 2) maintaining the confidence level of deadline meeting for the admitted chains; and 3) satisfying the ergodicity constraints (VMs' processing capacity constraints; switches' transmission rate capacity constraints i.e., equivalent to a bandwidth capacity constraint). Next, we explain ACRA and RA.

ACRA. ACRA is performed in two phases. In the first phase, resources are allocated to a maximum number of chains with minimum power consumption. This is performed by calling the RA (lines 1-2). The result of the RA includes an allocation profile, $\boldsymbol{X}$, and the queue network analysis for the allocation profile, QN_{Res}. The ergodicity condition of the system, i.e., the VMs' processing capacity constraints

Heuristic ACRA
CS: Set of all chains // CS is global variable
Phase 1(Chains: CS)**:** // CS is local variable
1 $EC \leftarrow \emptyset$
2 $\{\boldsymbol{X}, QN_{Res}\} \leftarrow RA(CS, \boldsymbol{D}, \boldsymbol{CL}, \bigcup_{i,n} \lambda_n^{i,back}, \bigcup_{i} \lambda^{i-1,i,back})$
3 While $\exists i : QN_{Res}.\lambda^{i-1,i} \geq QN_{Res}.\hat{\mu}^{i-1,i}$
or
$\exists v_n^i : QN_{Res}.\lambda_n^i \geq \mu_n^i$ or $\exists sw : \lambda_n^{sw} \geq \mu^{sw}$ do
4 $ch \leftarrow arg \min_{ch \in CS \backslash EC} \aleph_{ch}$
5 $EC \leftarrow EC \cup \{ch\}$
6 $\{\boldsymbol{X}, QN_{Res}\} \leftarrow RA(CS \backslash EC, \boldsymbol{D}, \boldsymbol{CL}, \bigcup_{i,n} \lambda_n^{i,back}, \bigcup_{i} \lambda^{i-1,i,back})$
7 End-While
8 Output: $\{CS, \boldsymbol{X}, QN_{Res}, EC\}$
Phase 2:
9 $BaseSol \leftarrow Output of\ Phase 1(CS)$
10 VS = {ch|ch ∈ CS\BaseSol.EC,
$BaseSol.QN_{Res}.\overline{PDM}_{ch} < CL_{ch}$}
11 $RevenueLoss \leftarrow \sum_{ch \in VS} \aleph_{ch}$
12 $ST \leftarrow Sort\ CS\ ascending\ by\ their\ revenue\ gain$
13 $ch_{ST}(k) \leftarrow Chain\ k\ from\ T$
14 $k \leftarrow \underset{k}{argmax} [\sum_{ch_{ST}(1)}^{ch_{ST}(k)} \aleph_{ch} < Revenue\ Loss]$
15 $RelaxSet \leftarrow \{ch_{ST}(1), \ldots ch_{ST}(k)\}$
16 $SolSet \leftarrow \{BaseSol\}$
17 $CS_{copy} \leftarrow CS$
18 While !$RelaxSet.empty()$ do
19 $ch \leftarrow Next\ element\ in\ RelaxSet$
20 $CS_{copy} \leftarrow CS_{copy} \backslash \{ch\}$
21 $NewSol \leftarrow Perform\ Phase\ 1(CS_{copy})$
22 $SolSet \leftarrow SolSet \cup \{NewSol\}$
23 $RelaxSet \leftarrow RelaxSet \backslash \{ch\}$
24 End-While
25 $BestSol \leftarrow The\ solution\ in\ SolSet\ that\ maximizes$ (19)
26 $\forall ch \in BestSol.EC : \zeta_{ch} \leftarrow 0$
27 $\forall ch \in CS \backslash BestSol.CS : \zeta_{ch} \leftarrow 0$
28 For other chains like ch
29 If $(BestSol.QN_{Re}.\overline{PDM}_{ch} < CL_{ch})$
30 $\zeta_{ch} \leftarrow 0$
31 Else
32 $\zeta_{ch} \leftarrow 1$
33 End-If
34 End-For
35 Return $BestSol.\boldsymbol{X}, \boldsymbol{\zeta}$

and the switches' transmission rates constraints (i.e., bandwidth capacity) in (29-31), are checked. In the case of violation, the least-profitable chain will be added to the set EC. The resource allocation is then performed once more with the excluded chains in set EC. The process of excluding chains is repeated until the ergodicity condition can be kept (lines 3-8). In the case where there are chains whose deadlines have not been met with the required CL, the second phase is performed.

The aim in the second phase is to select the subset of chains for the admission such that the network provider profit is maximized. The resource allocation result in the first phase is considered as a base solution. The solution is investigated to calculate the amount of revenue loss due to not meeting the CLs of some chains. The chains with the least revenue gain, i.e., those whose accumulated gain is less than the revenue loss, are considered as belonging to the *RelaxSet* (lines 9-15). This isthe set of chains that are candidates to be excluded in order to gain a new set of solutions, *SolSet*. The chains in *RelaxSet* are excluded from the set of chains in an additive manner. Each time the first phase is repeated and the result is added to *SolSet*. Note that by excluding chains with low revenue gains, there is a chance that resources can be allocated to other chains with high revenue gains. Therefore, the CL for more chains with high revenue gains, could be met in the new calls of the RA; thus enhancing the profit can be achieved (lines 16-24).

The solutions are evaluated by the objective function in (19) and the best solution is chosen (*BestSol*). The chains that were omitted due to ergodicity violation are regarded as non-admitted. The low-revenue gain chains that were excluded before obtaining the solution are also regarded as non-admitted. For the rest of the chains, if they have met their CL they are admitted, otherwise, they are not admitted. The resource allocation is done according to the best solution; i.e., $BestSol.\boldsymbol{X}$ (lines 25-35).

RA. Now, we explain the RA operation that is called by ACRA so that ACRA can optimize the objective function

i.e., (19). RA obtains a set of chains as input and allocates resources to the maximum number of chains while minimizing the power consumption. We exploit the Tabu method to implement the RA, as this meta-heuristichas proven quite promising to find near-optimal solutions in resource allocation problems [68], [85]. Tabu performs the search through an iterative process. It starts from an initial solution as the current solution. At each iteration it generates the neighbours of the current solution by applying some Tabu moves. The neighbours are evaluated according to a fitness function, and the search process continues from the best neighbour. The iteration continues until a stopping condition is met. A memory structure called Tabu-List is used to avoid looping during the search process, thereby preventing the exploration of previously-visited solutions. The main elements involved in Tabu-search are:

1) INITIAL SOLUTION

For each chain, a single VM in each required pool (from the VNF type that is required by the chain) is randomly chosen and allocated to the chain. Satisfaction of constraints (26-28) are considered in the selection.

2) TABU MOVES

The moves are defined below:

M0 (VM allocation for bulk chains) − A random VM in a stochastically-selected pool is allocated to all the chains which need that VNF type. The selection probability for pool

Heuristic RA

Input

CS : *Set of VNF chains*, $\boldsymbol{D}$: *Chains deadline vector*,

$\boldsymbol{CL}$: *CLs of Deadline meeting*, $\bigcup_{i,n} \lambda_n^{i,back}, \bigcup_i \lambda^{i-1,i,back}$

Output

$\boldsymbol{X_{best}}$: *Allocation profile*,

$QN_{Res}(\boldsymbol{X_{best}})$: *Queue network analysis for* $\boldsymbol{X_{best}}$

Function

1 For each chain //Initialize $\boldsymbol{X_{cur}}$

2 randomly assign a VM of any required pool (VNF type)

3 End-For

4 $QN_{Par} \leftarrow \{\bigcup_{i,n} \boldsymbol{H_n^i}, \bigcup_{i,n} \mu_n^i, \bigcup_i \hat{\mu}^{i-1,i}, \bigcup_{i,n} \lambda_n^{i,back}, \bigcup_i \lambda^{i-1,i,back}, \bigcup_{j\in CS} \lambda_j, \boldsymbol{X_{cur}}\}$

5 Analyze QN with QN_{Par} and store the result

$$QN_{Res}(\boldsymbol{X_{cur}}) \leftarrow \{\bigcup_{i,n} \lambda_n^i, \bigcup_i \lambda^{i-1,i}, \bigcup_j \overline{PDM}_j\}$$

6 $\boldsymbol{X_{best}} \leftarrow \boldsymbol{X_{cur}}$

7 $QN_{Res}(\boldsymbol{X_{best}}) \leftarrow QN_{Res}(\boldsymbol{X_{cur}})$

8 $it = 0, TbList = \emptyset$

9 While stopping condition is not met

10 $NeigList \leftarrow$ *Create Neighbours of* $\boldsymbol{X_{cur}}$

11 For each $\boldsymbol{X} \in NeigList$

12 $QN_{Par} \leftarrow \{\bigcup_{i,n} \boldsymbol{H_n^i}, \bigcup_{i,n} \mu_n^i, \bigcup_i \hat{\mu}^{i-1,i}, \bigcup_{i,n} \mu_n^{i,back}, \bigcup_i \mu^{i-1,i,back}, \bigcup_{j\in CS} \lambda_j, \boldsymbol{X}\}$

13 Analyze QN with QN_{Param}:

$$QN_{Res}(\boldsymbol{X}) \leftarrow \{\bigcup_{i,n} \lambda_n^i, \bigcup_i \lambda^{i-1,i}, \bigcup_j \overline{PDM}_j\}$$

14 Calculate $Fit(\boldsymbol{X}, \boldsymbol{CL})$

15 End-For

16 $\hat{\boldsymbol{X}} \leftarrow \underset{\boldsymbol{X}}{argmin}\, Fit(\boldsymbol{X}, \boldsymbol{CL})$

17 $M \leftarrow$ *Move that has yeilded* $\hat{\boldsymbol{X}}$

18 If M is not in *TbList*

19 Keep M in *TbList* for i_{tab} iterations

20 Else //aspiration criterion

21 if $Fit\left(\hat{\boldsymbol{X}}, \boldsymbol{CL}\right) < Fit(\boldsymbol{X_{best}}, \boldsymbol{CL})$ then

22 Remove M from *TbList*

23 End-If

24 If $Fit\left(\hat{\boldsymbol{X}}, \boldsymbol{CL}\right) < Fit(\boldsymbol{X_{best}}, \boldsymbol{CL})$

25 $\boldsymbol{X_{best}} \leftarrow \hat{\boldsymbol{X}}$

26 $QN_{Res}(\boldsymbol{X_{best}}) \leftarrow QN_{Result}(\hat{\boldsymbol{X}})$

27 End-If

28 End-While

29 Return $\boldsymbol{X_{best}}, QN_{Res}(\boldsymbol{X_{best}})$

Pl_i is given in (33). The pools that are used by more chains are more likely to be selected.

$$p\left(pl_i\right) = \frac{\sum_{j\in CS} y_{i,j}}{\sum_i \sum_{j\in CS} y_{i,j}} \tag{33}$$

M1 (VM allocation for a single chain) − A random chain among those who do not meet the CL of their deadlines, is stochastically selected in reverse-proportion to the deviation from CL. The logic behind the selection is that chains closer to their deadline-CL, are more probable to meet the CL by allocating more VMs. Eq. (34) shows the selection probability for chain j. An unassigned VM of a random pool that is required is assigned to that chain.

$$p(j) = \frac{\left[CL_j - \sum_{(n_1,\ldots n_K)\in I_{1..K}} \prod_{i=1}^{K} p_{n_i}^{i,j}.G_{T_j}(D_j)\right]^{-1}}{\sum_{j\in CS}\left[CL_j - \sum_{(n_1,\ldots n_K)\in I_{1..K}} \prod_{i=1}^{K} p_{n_i}^{i,j}.G_{T_j}(D_j)\right]^{-1}}. \tag{34}$$

M2 (VM deallocation from a single chain) − A chain is stochastically selected in proportion to the probability of deadline meeting. An assigned VM of a random pool that is required by that chain is deallocated for that chain. Note that to avoid disconnectivity, the selected pool should not have allocated just a single VM for that chain.

M3 (VM deallocation from bulk chains) − A random pool pl_i is selected. A VM that has been assigned to a minimum number of chains, i.e., $\underset{n}{arg\,min} \sum_{j\in CS} x_n^{i,j}$ is chosen and is deallocated from all hosted chains. To avoid chain disconnectivity, the hosted chains with a single assigned VM are randomly assigned to other VMs.

Note that M0 and M1 address meeting the CL of deadlines, while M2 and M3 reduce power consumption. Also, in all moves, constraints (26)-(28) are considered to be met. The other constraints are considered in the fitness function.

3) TABU-LIST MANAGEMENT

To avoid cycling in the search, the moves that yield to the best neighbour are marked as Tabu and stored in Tabu-list, *Tblist*, for a specific number of iterations, i_{tab}. A move can be removed from Tabu-list if it meets the aspiration criterion. The criterion is met when the move quality is better than the quality of the current best solution.

4) RESOURCE ALLOCATION SOLUTION FITNESS

The fitness function is defined as the aggregation of the power consumption and the penalty imposed by the constraints' violation, as shown in (35). Here, γ_1, γ_2, γ_3, γ_4, and γ_5 are coefficients for the purpose of normalization to assure that the deviation in the constraints and the power usage are at the same scale. Note that the optimization of (35) is consistent with the optimization model as defined in Section V.A (see constraints 29-32), and it is a required step to optimize the objective function (19).

$$Fit(\boldsymbol{X}, \boldsymbol{CL}) = \gamma_1 E(\boldsymbol{X}) + \gamma_2 \sum_{j=1}^{M} max\left(0, CL_j - \overline{PDM}_j\right) + \gamma_3 \sum_{i=1}^{K} max\left(0, \lambda^{i-1,i}(\boldsymbol{X}) - \hat{\mu}^{i-1,i}\right)$$

$$+\gamma_4 \sum_{sw\in SW\setminus PC} max\left(0,\lambda^{sw}(\boldsymbol{X})-\mu^{sw}\right) + \gamma_5 \sum_{i=1}^{K}\sum_{n=1}^{I_i} max\left(0,\lambda_n^i(\boldsymbol{X})-\mu_n^i\right). \quad (35)$$

C. DISCUSSION

For the sake of simplicity, like [30] we have advocated the benefit over using the shortest path routing (from the aspect of latency), and we have assumed the traffic is transmitted from physical servers to switches connecting the pools through the shortest path (or from switches to physical servers). Here, we discuss that the shortest path-approach is not mandatory and the proposed method can be adapted to decide about the routing within the resource allocation phase. To address this aim, the changes are required to be applied in the optimization model as below:

1) Decision variables that map the virtual links in the chains to the physical paths should be introduced. Let call them as ''link-allocation variables''.

2) For a chain that uses the two consecutive VNFs ip and i, for each VM in pool Pl_{ip} that has been allocated to the chain, the traffic should traverse through $SS^{i-1,i}$. Thus, a path from the VM to the switch $SS^{i-1,i}$ should be allocated to the virtual link, using the link-allocation variables. Similarly, from switch $SS^{i-1,i}$ to every allocated VM in pool Pl_i a path should be allocated using the link-allocation variables.

3) The link-allocation variables will be involved in calculating the effective transmission rate of switches interconnecting the pools, which demands modifications in equations (3), (4), (5). Similarly, the link-allocation variables will be involved in calculating the traffic arrival rate to the switches in $SW\setminus PC$, which demands modification in (23). Considering (3) as an example, the term μ^{sw} will be involved in (3), only when the switch sw has been located on the physical route from v_m^{ip} to v_n^i (passes through switch $SS^{i-1,i}$) which we have allocated to the virtual link between VNF type ip and i in the chain.

4) For every virtual link between two consecutive VNFs like ip and i in every chain, the connectivity of the allocated physical route should be met through defining some constraints which assign appropriate values to the link-allocation variables. Furthermore, for each chain, the traffic entrance to every switch on the allocated physical path should be equal to the traffic exit from that switch. This equality will be checked through some new constrains.

5) To adapt RA to decide about link-allocation variables, a random initial routing is required in initialization. Furthermore, for explorations purposes, changing the routing of the traffic for the chains, is required which can be performed by Tabu moves.

D. COMPLEXITY ANALYSIS

Let $|VM| = \sum_{i=1\ldots K} I_i$ be the number of VMs, and $I = \max_{i=1\ldots k} I_i$. The size of the search space is $2^{M.|VM|} \times 2^M$. Considering that the number of VNF types K is limited, we see it as ignorable constant for the analysis.

Complexity of RA: At each iteration of RA, the moves are performed, the queue network is analyzed, and the fitness function is calculated. The moves are performed at a complexity of $o(M + I.logI)$. The traffic arrival rates to the VMs and the switches in (8) and (13) are calculated with the complexity of $o(M.I^2)$.

The $\overline{PDM}$ in (18) is calculated in $o(M.I)$. The calculation of (23) takes $o(I^2.|SW|)$. The calculation of the utilization of physical servers and switches takes $o\left(N + |SW|\right)$. All the analyses are performed in $o(N + \mathrm{M}.I + I^2.|SW|)$.

The fitness function in (35) is calculated in $o\big(N + M + |SW| + |VM|\big)$, which is less than the complexity of queue network analysis. Therefore, the complexity of RA is $o(N + M.I + I^2.|SW|)$ (ignoring the constant number of iterations).

Complexity of ACRA: In the best case, only the first phase is performed with complexity of $o\left(N + \mathrm{M}.I + I^2.|SW|\right)$. In the worst case, the chains are sorted by their gain and RA is called M times with the complexity of $o(M.N + \mathrm{M}^2.I + M.I^2.|SW|)$.

VI. PERFORMANCE EVALUATION

In this section, we provide the simulation results to evaluate the efficiency of the proposed method in comparison with other algorithms in the literature.

A. SIMULATION SETUP AND BASELINES

The simulation was conducted with a Java program running on an Intel Core™ i7-6600U processor with 8 GB of memory. The NFVI consists of 15 physical servers (a scale similar to [35], [49]), and 9 software-defined switches. We used Dragonfly topology, a common topology in data centers [86], [87]. According to this topology, switches are fully connected and provide connection among physical servers. Each physical server is connected randomly to a switch. We consider 8 VNF types with traffic scaling ratios chosen randomly in the range of [0.1, 2]. A pool of VMs is associated to each VNF type. Each pool contains 5 VMs that are randomly distributed on the physical servers. Thus, there are a total of 40 VMs.

The mean transmission rate of the switches is chosen randomly between 1 Gbps and 10 Gbps according to real systems [88]. The mean traffic processing rate of the VMs is chosen randomly in the range of 10 to 100 Mbps to cover the required throughput of standard instances for VNF types including Firewall, WAN Optimization Controller, IDS, and IPS [89].

Similar to [89], [90], we set the PUE of the physical servers and switches randomly in the range of 1 to 3 Watts, static power consumption in the range of 40 to 60 Watts, and dynamic power consumption when the utilization of physical servers/switches are maximum in the range of 100 to 300 Watts. Parameter Γ is 0.02 of the unit of currency.

Each VNF chain requires a random subset of VNF types. The source and destination nodes of each chain are randomly

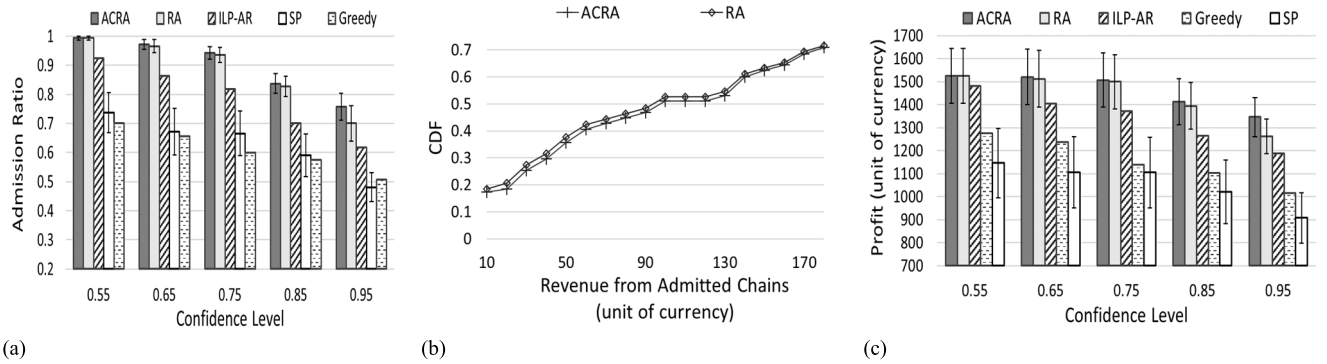

FIGURE 3. The results for 10 chains (a) The admission ratio for various CLs, (b) Fraction of admitted chains with less than a specific revenue. Here, CL is 0.65 (c) Network provider profit for various CLs. The results are average of 40 runs. The vertical bars indicate the confidence-interval of 90%.

selected from the physical servers. The mean traffic arrival rate at each chain is set in the range of 50 Kbps to 500 Kbps according to the demand of real applications in Web service, VoIP, and online Gaming [89]. The revenue gain for each VNF chain is selected randomly in the range of 10 to 300 units of currency such that the chains that have more VNFs and a higher CL bring more revenue for the system. The search in Tabu is performed for at least 100 iterations and at most 300 iterations. After 100 iterations, the search process terminates if the quality of the best resource allocation solution does not change within the last 40 iterations. We found the value of 2 for i_{tab} appropriate. Finally, we assume the size of a chunk of data to be the same as the average size of a packet, 256 bytes [91].

We compared our proposed heuristic (ACRA) with the following four baselines: Our proposed resource allocation heuristic RA, SP [12], ILP-AR [41], and a greedy algorithm called Greedy. RA and SP [12] utilize parallel VNF processing, while ILP-AR [41] and Greedy process the traffic sequentially. SP, ILP-AR, and Greedy employ deterministic modeling of the system. An overview of the baselines is presented below.

1) SP is a resource allocation (without an admission control) method based on deterministic modeling of the system, we proposed in [12]. SP utilizes the same pooling idea of this paper to enable parallel VNF processing however, the resource allocation mechanism in SP, dedicates each VM in a pool to a single chain. It allocates VMs to chains with the objective of cost minimization while respecting the chains' deadline constraints. We have defined power consumption as cost criterion, to have power efficiency in resource allocation mechanism of SP.

2) ILP-AR is a joint **A**dmission control and **R**esource allocation algorithm for VNF chains [41]. It decides on the admission of chains and allocates resources to them so as to maximize an aggregation of the revenue (obtained from chain admissions) and resource usage preference. Deadline constraints for chain execution time are considered in the optimization. We have defined the preferences to prioritize servers with less power consumption. The coefficient for weighting the revenue in comparison to the power consumption is the same value as the best selected value in [41]. This value gives priority to chain admission. ILP-AR models the problem as an ILP optimization. Like [41], we used CPLEX to implement ILP-AR.

3) Greedy gives priority in resource allocation to chains with high revenue gain. The chains are sorted in a list in descending order according to their revenue gain. For each VNF of a chain, the fastest VM in the pool with enough capacity to process the traffic, is allocated. When not enough VMs are found to host all the VNFs in the chain, the resources for that chain are released and the allocation process is performed for the next chain.

To have a fair comparison and illustrate the effectiveness of our proposed method, as baseline, we have selected the aforementioned methods since they have utilized similar criteria to our proposed method in their resource allocation and/or admission. SP which allocates resources based on parallel VNF processing considers power efficiency, while ILP-AR considers both revenue and power consumption in joint admission and resource allocation. Greedy also considers revenue gain of the chains in resource allocation.

B. RESULTS

In this subsection we give the results of our simulation. Due to the randomness nature of initial solution and the Tabu moves in RA and ACRA the results for these methods are reported for the average of 40 runs. Fig. 3 presents the results when there are 10 VNF chains and the chains' deadlines are randomly selected in the range of 4 to 8 msec.

Fig. 3(a) illustrates the admission ratios, showing that in all five methods, when the CL of deadline-meeting increases the admission ratio decreases. This occurs because when the CL increases, the size of the feasible domain is smaller and so solving the optimization becomes more difficult in all methods. Greedy has the lowest admission ratio, as it does not consider time constraints in resource allocation. Other methods consider time constraints, among which, SP has a poor admission ratio since it wastes resources by allocating

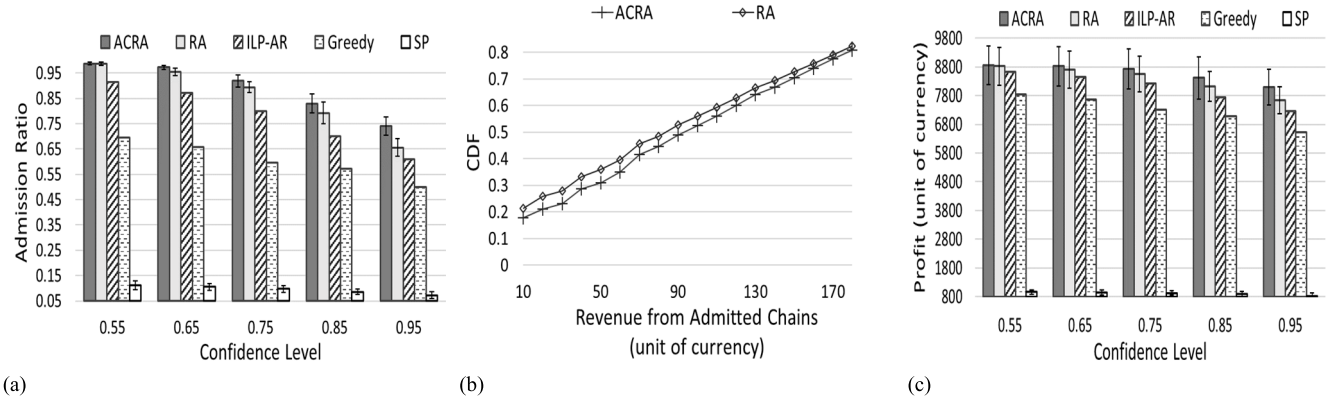

FIGURE 4. The results for 60 VNF chains (a) The admission ratio for various CLs, (b) Fraction of admitted chains with less than a specific revenue. Here, CL is 0.65. (c) Network provider profit for various CLs. The results are average of 40 runs. The vertical bars indicate the confidence-interval of 90%.

each VM to a single chain. ACRA and RA performed better than ILP-AR because: 1) they exploit parallelism for chain execution, which speeds up the chains' executions and tends to fewer deadline violations; and 2) they are based on stochastic analysis, which is more precise than the deterministic analysis used in ILP-AR. ACRA and RA are competitive, with ACRA performing particularly well for high confidence levels. ACRA offers such a high performance because it explores combinations of chains to decide about the admission of chains, which lead to higher admission ratios.

Fig. 3(b) illustrates the Cumulative Distribution Function (CDF) of the revenue obtained from VNF chain admission when the CL is 0.65. We show the CDF for the two methods that have the highest admission ratio: ACRA and RA. The slower the growth of a curve the greater the admission of chains with higher gain. The admitted chains in ACRA have a higher gain than those in the RA, as the ACRA gives resource usage priority to chains with higher gains when the admission of all chains is not possible.

Fig. 3(c) shows the network provider profit. The profit decreases in all methods for higher CLs, as the admission of chains becomes more difficult under higher CL conditions. While SP achieved higher admission ratios than Greedy, there was not a significant difference (See Fig. 3(a)). On the other hand, Greedy prioritizes the admission of chains with higher gains, and in comparison with SP, it gained higher profits. The ACRA that has the highest admission ratio also has the highest profit.

Fig. 4 shows the results for higher-load of system; in this case, 60 VNF chains. As indicated in Fig. 4(a), the SP shows a markedly poor admission ratio, mainly because it wastes resources by under-utilizing them due to a dedication of a VM to a chain. This restriction does not allow the SP to allocate resources to a fraction of chains, which causes the admission ratio to be reduced. Note that such under-utilization of resources does not have as much of an impact in lightly loaded systems, as indicated in Fig. 3(a). The difference between ACRA and RA is greater at 60 chains than it is with 10 chains, which highlights the importance of admission control applications in higher-load systems where there is more competition for resource usage. The ACRA outperformed the other methods. Its admission ratio is up to 8% higher than that of the RA due to its application of admission control and, up to 13% higher than ILP-AR because it exploits parallel VNF execution and stochastic analysis.

Fig. 4(b) shows the CDF for the revenue obtained from VNF chain admission for ACRA and RA when CL is 0.65. ACRA admitted chains with higher gains. For example, 51% of the chains admitted in ACRA have a gain of more than 90, while 47% of the chains admitted in RA have a gain of more than 90.

Fig. 4(c) shows the network provider's profit variation. ACRA outperforms the other methods because of its higher admission ratio and its admission of chains with higher gain.

Fig. 5 shows the results when the number of chains is changed from 40 to 160. The CLs were chosen randomly with uniform distribution in the range of 0.55 to 0.95. Fig. 5(a) indicates the admission ratio variation. In all five methods the admission ratio decreases when the number of chains increases. This decrease is due to the increased competition for resources at higher system loads. The performance of SP decreases most notably, with its admission ratio at only 0.04 when there are 160 chains. This low admission ratio shows the detrimental side-effect of dedicating a VM to a single chain. The outperformance of ACRA over other methods increases when there are more chains in the system. At the 160-chain level, ACRA shows an admission ratio that is greater than those of the RA, ILP-AR, Greedy, and SP by differences of 0.09, 0.24, 0.32, and 0.83, respectively. The improvement is due to ACRA's consideration of admission decisions in the resource allocation and its utilization of parallel processing for chain execution.

Fig. 5(b) indicates the power consumption. Generally, in all methods more power is consumed when the number of chains increases. Although the admission ratio goes down with

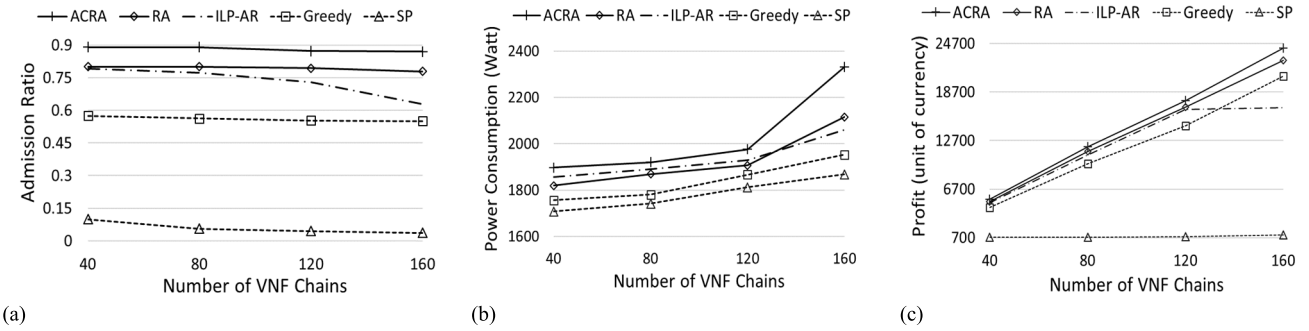

FIGURE 5. The effect of number of chains. (a) Admission ratio (b) Power consumption (c) Network provider profit. The results are average of 40 runs.

the increase in the number of chains, (considerably) more chains will be admitted to the system; thus increasing the power consumption because the higher admissions of chains increases the processing/transmission loads at VMs/switches. The ACRA admits more chains and thus consumes more power than the other methods. However, the power usage increment is less than the admission ratio increase. For example, at 160 chains, ACRA consumes only 217 additional Watts to increase its admission ratio by 0.09 compared to that of RA. This is possible because ACRA maximizes the admission of chains with a minimum of power consumption in order to maximize the network provider's profit.

Fig. 5(c) illustrates the network provider profit. When the number of chains increases, the profit increases for ACRA, RA, ILP-AR, and Greedy, as they admit more chains at higher loads. However, SP cannot admit more chains, and so it will not gain more profit. ACRA shows the highest amount of profit gain. Indeed, it increased the profit by 1562 in comparison with RA when there are 160 chains. ACRA has the highest performance because it has the highest admission ratio and prioritizes admitting chains with high revenue.

Fig. 6 illustrates the effect of pool size on admission ratio and profit where there exists 50 VNF chains. The CLs were chosen randomly with uniform distribution in the range of 0.55 to 0.95. We changed the pool size in the range of 2 (i.e., total of 16 VMs) till 5 (i.e., total of 40 VMs). Fig. 6(a) indicates the admission ratio. When the number of VMs per pool increases, the admission ratio increases as well in ACRA, RA, and SP. The reason is that these methods exploit parallel VNF processing and for higher number of VMs in the pool, they can exploit more parallelism (i.e., splitting traffic processing of a chain among more VMs is possible) which helps the satisfaction of deadlines with required CLs, thereby, increasing the admission ratio. In contrast, as sequential traffic processing has limited power in meeting CL of deadlines, increasing the number of VMs in the pools will not necessarily increases the admission ratio in sequential traffic processing based methods i.e., Greedy and ILP-AR, thereby, there exists fluctuations in admission ratio for these methods. Considering that higher admission ratio will have higher profit, similar behavioral patterns can

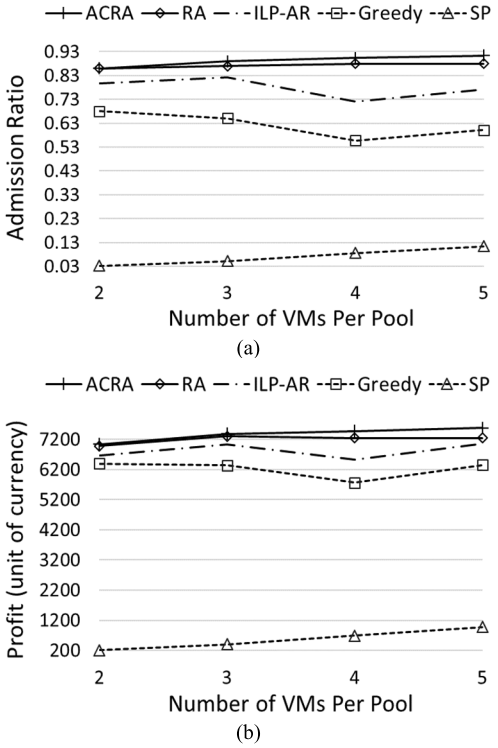

FIGURE 6. Effect of number of VMs per pool on admission ratio and network provider profit. There are 50 VNF chains. The results are average of 40 runs.

be seen for profit as indicated in Fig. 6(b). As it can be seen in Fig. 6(a), ACRA has gained higher admission ratio in comparison with the baselines as a result of stochastic modeling of the system and exploiting parallel VNF processing to perform a joint admission control and resource allocation. Accordingly, as it can be seen in Fig. 6(b) it has gained higher profit as a result of higher admission.

To assess the effect of traffic, Fig. 7 shows the admission ratio and network provider profit in ACRA. Here, there are 5 VMs per pool and 60 VNF chains. The mean traffic arrival rate of each chain changes in the range of 200 to 650 kbps. The deadlines and CLs have been chosen randomly in the range of [4, 8] msec., and [0.55, 0.95] respectively. As it

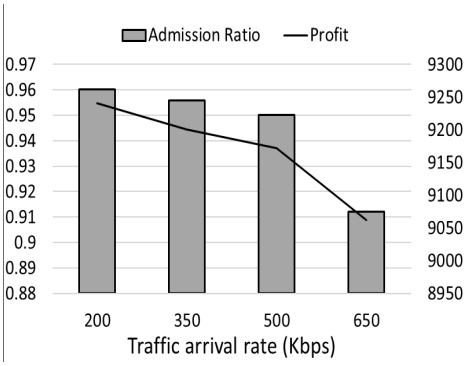

FIGURE 7. Effect of traffic on admission ratio and network provider profit in ACRA. There are 60 VNF chains and 5 VMs per pool. Profit is in unit of currency. The results are average of 40 runs.

TABLE 2. Comparison of the ACRA with the optimal solution. Two VNF types and two VMs per pool, search space size is 33,554,432 for 5 chains and 1.1 * 10^9 for 6 chains. The results are for an average of 40 runs.

		Admission ratio		Profit	
CL	Number of chains	ACRA	Optimal	ACRA	Optimal
0.6	5	1	1	836	836
	6	1	1	921	921
0.8	5	0.708	0.714	511.6	512.3
	6	0.70	0.71	664	666

can be seen, admission ratio decreases when the traffic rate increases. The reason is that higher traffic rates impose higher loads to the VMs for traffic processing, and to the switches for traffic transmission. Thus, meeting the deadlines according to the requested CLs, becomes more difficult and fewer chains will be admitted. Accordingly, less profit will be gained in higher traffic arrival rates as a result of admission reduction.

Table 2 illustrates the comparison of ACRA with an optimal solution determined by exhaustive research. Note that the optimal solution can be calculated in a reasonable amount of time for small scales of the problem. There are 2 VNF types and 2 VMs per pool (a total of 4 VMs). The simulation was conducted for 5 and 6 chains, where all chains need the 2 VNF types. The size of the search spaces for 5 and 6 chains are $33,554,432$ and 1.1×10^9, respectively. Note that for the case of 6 chains, finding the optimal solution at each run took an average of 7 hours and 12 minutes. The admission ratio and profit in ACRA for CL of 0.6 are the same as with the optimal solution. For a tighter CL of 0.8, where the feasible domain is smaller, ACRA does not obtain the optimal admission ratio and profit, but gets very close to the optimal values. This shows the effectiveness of ACRA in finding solutions.

VII. CONCLUSION

This paper provides a method for the joint admission control and resource allocation for VNF chains for applications with time constraints in chain execution. Pools of VNFs that execute the traffic in parallel are utilized to speed up traffic processing for tight time constraints. VNF chain execution is modeled by a Queue Network. The Queue theory is applied to calculate the expected value for the probability of deadline-meeting in VNF chains. The problem is modeled as a joint optimization that decides on the admission of VNF chains and the resource allocations for the admitted chains. The objective is to maximize the profit of the network provider while keeping the confidence level of deadline-meeting for the admitted chains at desired levels. The power consumption of the physical servers and of the switches is considered in the profit calculation. A heuristic is proposed to solve the problem. Simulation results show that our method improves the admission ratio and the network provider profit when compared to three other methods. We have assumed that the size of pools has been given. Providing a solution to determine the optimal size of VNF pools is a future work.

REFERENCES

[1] 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 Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[2] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[3] T. W. Kuo, B. H. Liou, K. C. Lin, and M. J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1562–1576, Aug. 2018.

[4] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. 1st IEEE Conf. Netw. Soft.*, Apr. 2015, pp. 1–9.

[5] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[6] M. Aazam, K. A. Harras, and S. Zeadally, "Fog computing for 5G tactile industrial Internet of Things: QoE-aware resource allocation model," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3085–3092, May 2019.

[7] N. T. Jahromi, S. Kianpisheh, and R. H. Glitho, "Online VNF placement and chaining for value-added services in content delivery networks," in *Proc. IEEE Symp. Local Metrop. Area Netw.*, Jun. 2018, pp. 19–24.

[8] T. Truong-Huu, P. M. Mohan, and M. Gurusamy, "Service chain embedding for diversified 5G slices with virtual network function sharing," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 826–829, Mar. 2019.

[9] A. Baumgartner, T. Bauschert, A. A. Blzarour, and V. S. Reddy, "Network slice embedding under traffic uncertainties—A light robust approach," in *Proc. 13rd Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–5.

[10] A. Kammoun, N. Tabbane, G. Diaz, A. Dandoush, and N. Achir, "End-to-end efficient heuristic algorithm for 5G network slicing," in *Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, May 2018, pp. 386–392.

[11] G. Wang, G. Feng, W. Tan, S. Qin, R. Wen, and S. Sun, "Resource allocation for network slices in 5G with network resource pricing," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[12] S. Kianpisheh and R. H. Glitho, "Cost-efficient server provisioning for deadline-constrained VNFs chains: A parallel VNF processing approach," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6.

[13] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 1–6.

[14] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Demand-aware network function placement," *J. Lightw. Technol.*, vol. 34, no. 11, pp. 2590–2600, Jun. 1, 2016.

[15] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Routing algorithms for network function virtualization enabled multicast topology on SDN," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 4, pp. 580–594, Dec. 2015.

[16] X. Wang, C. Wu, F. Le, and F. C. M. Lau, "Online learning-assisted VNF service chain scaling with network uncertainties," in *Proc. IEEE 10th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2017, pp. 205–213.
[17] F. Esposito, D. Di Paola, and I. Matta, "A general distributed approach to slice embedding with guarantees," in *Proc. IFIP Netw. Conf.*, 2013, pp. 1–9.
[18] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi, and R. Glitho, "CCVP: Cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, Jul. 2017, pp. 1–9.
[19] S. Sahhaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, and P. Demeester, "Network service chaining with optimized network function embedding supporting service decompositions," *Comput. Netw. J.*, vol. 93, pp. 492–505, Dec. 2015.
[20] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in *Proc. Workshop Local Metrop. Area Netw.*, 2015, pp. 1–6.
[21] D. B. Oljira, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, "A model for QoS-aware VNF placement and provisioning," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–7.
[22] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in *Proc. IFIP/IEEE Symp. Integr. Netw. Manage.*, 2015, pp. 98–106.
[23] 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 Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1061–1075, Dec. 2017.
[24] N. E. Khoury, S. Ayoubi, and C. Assi, "Energy-aware placement and scheduling of network traffic flows with deadlines on virtual network functions," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 89–94.
[25] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware VNF placement and chaining based on a flexible resource allocation approach," in *Proc. 13rd Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–7.
[26] B. Farkiani, B. Bakhshi, and S. A. MirHassani, "A fast near-optimal approach for energy-aware SFC deployment," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1360–1373, Dec. 2019.
[27] L. Fang, X. Zhang, K. Sood, Y. Wang, and S. Yu, "Reliability-aware virtual network function placement in carrier networks," *J. Netw. Comput. Appl.*, vol. 154, Mar. 2020, Art. no. 102536.
[28] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, "Latency-aware service function chain placement in 5G mobile networks," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 133–141.
[29] A. Hmaity, M. Savi, L. Askari, F. Musumeci, M. Tornatore, and A. Pattavina, "Latency and capacity-aware placement of chained virtual network functions in FMC metro networks," *Opt. Switching Netw.*, vol. 35, Jan. 2020, Art. no. 100536.
[30] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan, "Reconfiguration in network slicing—Optimizing the profit and performance," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 591–605, Jun. 2019.
[31] I. Jang, D. Suh, S. Pack, and G. Dán, "Joint optimization of service function placement and flow distribution for service function chaining," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2532–2541, Nov. 2017.
[32] R. Challa, V. V. Zalyubovskiy, S. M. Raza, H. Choo, and A. De, "Network slice admission model: Tradeoff between monetization and rejections," *IEEE Syst. J.*, vol. 14, no. 1, pp. 657–660, Mar. 2020.
[33] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1455–1470, Jun. 2019.
[34] V. Millnert, E. Bini, and J. Eker, "AutoSAC: Automatic scaling and admission control of forwarding graphs," *Ann. Telecommun.*, vol. 73, nos. 3–4, pp. 193–204, Apr. 2018.
[35] M. A. T. Nejad, S. Parsaeefard, M. A. Maddah-Ali, T. Mahmoodi, and B. H. Khalaj, "VSPACE: VNF simultaneous placement, admission control and embedding," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 542–557, Mar. 2018.
[36] T. Lukovszki and S. Schmid, "Online admission control and embedding of service chains," in *Proc. Int. Colloq. Structural Inf. Commun. Complex.*, 2015, pp. 104–118.
[37] S. Ayoubi, S. Sebbah, and C. Assi, "A logic-based benders decomposition approach for the vnf assignment problem," *IEEE Trans. Cloud Comput.*, vol. 7, no. 4, pp. 894–906, Dec. 2019.
[38] Y. Xie, S. Wang, and Y. Dai, "Revenue-maximizing virtualized network function chain placement in dynamic environment," *Future Gener. Comput. Syst.*, vol. 108, pp. 650–661, Jul. 2020.
[39] J. Li, W. Liang, M. Huang, and X. Jia, "Reliability-aware network service provisioning in mobile edge-cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1545–1558, Jul. 2020.
[40] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware VNF placement for service-customized 5G network slices," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 2449–2457.
[41] P. Cappanera, F. Paganelli, and F. Paradiso, "VNF placement for service chaining in a distributed cloud environment with multiple stakeholders," *Comput. Commun.*, vol. 133, pp. 24–40, Jan. 2019.
[42] G. Yuan, Z. Xu, B. Yang, W. Liang, W. K. Chai, D. Tuncer, A. Galis, G. Pavlou, and G. Wu, "Fault tolerant placement of stateful VNFs and dynamic fault recovery in cloud networks," *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106953.
[43] M. Huang, W. Liang, Y. Ma, and S. Guo, "Maximizing throughput of delay-sensitive NFV-enabled request admissions via virtualized network function placement," *IEEE Trans. Cloud Comput.*, early access, May 10, 2019, doi: 10.1109/TCC.2019.2915835.
[44] S. M. A. Araujo, F. S. H. de Souza, and G. R. Mateus, "Virtual network embedding in multi-domain environments with energy efficiency concepts," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 205–210.
[45] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an OpenFlow architecture," in *Proc. IEEE Teletraffic Congr.*, Sep. 2011, pp. 1–7.
[46] H. Deng, L. Huang, C. Yang, H. Xu, and B. Leng, "Optimizing virtual machine placement in distributed clouds with M/M/1 servers," *Comput. Commun.*, vol. 102, pp. 107–119, Apr. 2017.
[47] G. Bolch, S. Greiner, H. De Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation With Computer Science Applications*. Hoboken, NJ, USA: Wiley, 2006.
[48] N. Ashraf, A. Hasan, H. K. Qureshi, and M. Lestas, "Combined data rate and energy management in harvesting enabled tactile IoT sensing devices," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3006–3015, May 2019.
[49] M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, "Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 374–388, Mar. 2019.
[50] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: A profit maximization approach," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 1017–1025, Jun. 2013.
[51] Y. Zhang, F. He, and E. Oki, "Availability-aware service chain provisioning with sub-chain-enabled coordinated protection," in *Proc. IEEE/IFIP Symp. Integr. Netw.*, May 2021, pp. 1–6.
[52] H. Huang, C. Zeng, Y. Zhao, G. Min, Y. Zhu, W. Miao, and J. Hu, "Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2558–2571, Aug. 2021.
[53] X. Shang, Y. Huang, Z. Liu, and Y. Yang, "Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme," *IEEE Trans. Mobile Comput.*, early access, Jan. 1, 2021, doi: 10.1109/TMC.2020.3048885.
[54] G. Sallam and B. Ji, "Joint placement and allocation of VNF nodes with budget and capacity constraints," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1238–1251, Jun. 2021.
[55] C. Assi, S. Ayoubi, N. El Khoury, and L. Qu, "Energy-aware mapping and scheduling of network flows with deadlines on VNFs," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 192–204, Mar. 2019.
[56] J. Li, W. Shi, Q. Ye, N. Zhang, W. Zhuang, and X. Shen, "Multiservice function chain embedding with delay guarantee: A game-theoretical approach," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11219–11232, Jul. 2021.
[57] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic SFC mapping and scheduling in SDN/NFV-enabled space-air-ground integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, early access, Feb. 9, 2021, doi: 10.1109/JIOT.2021.3058250.
[58] L. Liu, S. Guo, G. Liu, and Y. Yang, "Joint dynamical VNF placement and SFC routing in NFV-enabled SDNs," *IEEE Trans. Netw. Service Manage.*, early access, Jun. 30, 2021, doi: 10.1109/TNSM.2021.3091424.

[59] D. Spatharakis, I. Dimolitsas, D. Dechouniotis, G. Papathanail, I. Fotoglou, P. Papadimitriou, and S. Papavassiliou, "A scalable edge computing architecture enabling smart offloading for location based services," *Pervas. Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101217.
[60] J. Sun, F. Liu, M. Ahmed, and Y. Li, "Efficient virtual network function placement for Poisson arrived traffic," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.
[61] *Remote Surgery Conducted in 5G*. [Online]. Available: https://www.chinadaily.com.cn/a/201908/29/WS5d670e17a310cf3e355686fa.html
[62] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi, "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 472–523, 1st Quart., 2021, doi: 10.1109/COMST.2020.3025995.
[63] *MEF Document on QoS*. [Online]. Available: https://www.mef.net/wp-content/uploads/2016/08/MEF-23-2.pdf
[64] Y. Ma, W. Liang, M. Huang, and S. Guo, "Profit maximization of NFV-enabled request admissions in SDNs," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.
[65] F. Callegati, W. Cerroni, C. Contoli, R. Cardone, M. Nocentini, and A. Manzalini, "SDN for dynamic NFV deployment," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 89–95, Oct. 2016.
[66] M. Dieye, S. Ahvar, J. Sahoo, and E. Ahvar, "CPVNF: Cost-efficient proactive VNF placement and chaining for value-added services in content delivery networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 2, pp. 774–786, Jun. 2018.
[67] H. Guo, Y. Wang, Z. Li, X. Qiu, H. An, P. Yu, and N. Yuan, "Cost-aware placement and chaining of service function chain with VNF instance sharing," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–8.
[68] C. Mouradian, S. Kianpisheh, M. Abu-Lebdeh, F. Ebrahimnezhad, N. T. Jahromi, and R. H. Glitho, "Application component placement in NFV-based hybrid cloud/fog systems with mobile fog nodes," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1130–1143, May 2019.
[69] Q. Liu, C. Weng, M. Li, and Y. Luo, "An in-VM measuring framework for increasing virtual machine security in clouds," *IEEE Secur. Privacy*, vol. 8, no. 6, pp. 56–62, Nov./Dec. 2010.
[70] B. Chafika, T. Taleb, C.-T. Phan, C. Tselios, and G. Tsolis, "Distributed AI-based security for massive numbers of network slices in 5G & beyond mobile systems," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit*, Jun. 2021, pp. 401–406.
[71] F. Luo, C. Jiang, S. Yu, J. Wang, Y. Li, and Y. Ren, "Stability of cloud-based UAV systems supporting big data acquisition and processing," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 866–877, Jul. 2019.
[72] Y. Xiao and M. Krunz, "Distributed optimization for energy-efficient fog computing in the tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2390–2400, Nov. 2018.
[73] E. F. Kfoury, J. Crichigno, and E. Bou-Harb, "An exhaustive survey on P4 programmable data plane switches: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 9, pp. 87094–87155, 2021.
[74] R. Miao, H. Zeng, C. Kim, J. Lee, and M. Yu, "SilkRoad: Making stateful Layer-4 load balancing fast and cheap using switching ASICs," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 15–28.
[75] J.-L. Ye, C. Chen, and Y. Huang Chu, "A weighted ECMP load balancing scheme for data centers using P4 switches," in *Proc. IEEE 7th Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2018, pp. 1–4.
[76] K. C. Leung, V. O. K. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (TCP): Problems, solutions, and challenges," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 522–535, Apr. 2007.
[77] N. Enomoto, H. Shimonishi, J. Higuchi, T. Yoshikawa, and A. Iwata, "High-speed, short-latency multipath Ethernet transport for interconnections," in *Proc. IEEE Symp. High Perform. Interconnects*, 2008, pp. 75–84.
[78] P. G. Harrison, "Response time distributions in queueing network models," in *Proc. Perform. Eval. Comput. Commun. Syst.*, 1993, pp. 147–164.
[79] E. Reich, "Waiting times when queues are in tandem," *Ann. Math. Statist.*, vol. 28, no. 3, pp. 768–773, Sep. 1957.
[80] M. Haviv, "Queues a course in queueing theory," Hebr. Univ. Jerus., Jerusalem, Israe, Tech. Rep. 91905, 2009.
[81] D. D. Kouvatsos, "A maximum entropy analysis of the G/G/1 queue at equilibrium," *J. Oper. Res. Soc.*, vol. 39, no. 2, pp. 183–200, Feb. 1988.
[82] M. A. Zazanis and R. Suri, "Estimating first and second derivatives of response time for G/G/1 queues from a single sample path," *Queueing Syst.*, Dec. 1985.
[83] Y.-K. Chu and J.-C. Ke, "Mean response time for a G/G/1 queueing system: Simulated computation," *Appl. Math. Comput.*, vol. 186, no. 1, pp. 772–779, Mar. 2007.
[84] M. Akkouchi, "On the convolution of exponential distributions," *J. Chungcheong Math. Soc.*, vol. 21, no. 4, pp. 501–510, Dec. 2008.
[85] W. Wang, P. Hong, D. Lee, J. Pei, and L. Bo, "Virtual network forwarding graph embedding based on Tabu search," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.
[86] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova, "Swap-and-randomize: A method for building low-latency HPC interconnects," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 7, pp. 2051–2060, Jul. 2015.
[87] N. Maksic and A. Smiljanic, "Improving utilization of data center networks," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 32–38, Nov. 2013.
[88] A. Leivadeas, G. Kesidis, M. Falkner, and I. Lambadaris, "A graph partitioning game theoretical approach for the VNF service chaining problem," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 890–903, Dec. 2017.
[89] 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 Trans. Services Comput.*, vol. 13, no. 1, pp. 172–185, Jan. 2020.
[90] M. K. Patterson, "The effect of data center temperature on energy efficiency," in *Proc. 11st Intersociety Conf. Thermal Thermomech. Phenomena Electron. Syst.*, May 2008, pp. 1167–1174.
[91] J. Rao and S. Vrzic, "Packet duplication for URLLC in 5G: Architectural enhancements and performance analysis," *IEEE Netw.*, vol. 32, no. 2, pp. 32–40, Mar./Apr. 2018.

SOMAYEH KIANPISHEH received the B.S. degree in software computer engineering from the University of Tehran, Iran, in 2004, and the M.S. and Ph.D. degrees in computer engineering from Tarbiat Modares University, Iran, in 2010 and 2016, respectively. From 2018 to 2020, for a period of two years, she was a Postdoctoral Researcher at Concordia University, Canada. Since 2021, she has been a Postdoctoral Researcher at Aalto University, Finland. Her research interests include NFV, SDN, fog/cloud systems, tactile internet, and 5G and beyond.

ROCH H. GLITHO (Senior Member, IEEE) received the M.Sc. degree in business economics from the University of Grenoble, France, the M.Sc. degree in pure mathematics and the M.Sc. degree in computer science from the University of Geneva, Switzerland, and the Ph.D. (Tech.Dr.) degree in informatics from the Royal Institute of Technology, Stockholm, Sweden. He worked in industry and has held several senior technical positions, such as a Senior Specialist, a Principal Engineer, and an Expert with Ericsson, Sweden and Canada. He is currently a Full Professor with Concordia University, where he holds a Canada Research Chair. He also holds the Ericsson/ENCQOR-5G Senior Industrial Research Chair in cloud and edge computing for 5G and beyond. In addition, he is also a Professor extraordinaire with the Computer Science Department, University of the Western Cape, Cape Town, South Africa. He has served as an IEEE Distinguished Lecturer and the Editor-in-Chief for the *IEEE Communications Magazine* and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.