IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 4, 2019, accepted February 19, 2019, date of publication March 1, 2019, date of current version March 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902432

Optimizing Network Slice Dimensioning via Resource Pricing

GANG WANG, GANG FENG, (Senior Member, IEEE), SHUANG QIN, (Member, IEEE), RUIHAN WEN, AND SANSHAN SUN

National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Gang Feng (fenggang@uestc.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61631004 and Grant 61871099.

ABSTRACT Network slicing has been viewed as a key enabler for the next-generation software-defined and cloud-based network (*e.g.*, 5G and beyond) to accommodate diverse services in a flexible and cost-efficient fashion. Network slicing allows a network slice provider (NSP) to operate on a common network infrastructure to create customized isolated logical networks (*i.e.*, network slices) for network slice customers (NSCs), (*i.e.*, service providers). NSP and NSCs are independent operators who pursue profit maximization, while in the literature, only network cost optimization is intensively investigated in terms of service function chain embedding, *i.e.*, virtual network function (VNF) placement and flow routing. Therefore, slices should be dimensioned (*i.e.*, resources allocated to slices) according to the resource availability and the economic mechanism in the network, so as to optimize the resource utilization and improve the profit of NSP/NSCs. In this paper, we study elastic slice dimensioning with resource pricing as a Stackelberg pricing game, in which the NSP sells slices by pricing resources and NSCs adjust their slice's resource demand on VNF capacity and bandwidth, while both are trying to maximize their profit. Then, we formulate optimization problems for the pricing game and find that a closed form solution of the optimal price cannot be obtained for a non-trivial network. Hence, we propose a resource pricing algorithm that aims to maximize the NSP's profit and the network's social welfare. Compared with existing usage-based pricing method and two heuristic methods, our proposed pricing algorithm for slice dimensioning strikes a trade-off between maximizing NSP's profit and other metrics, including the resource utilization. Hence, it will helpfully exploiting the benefits of network slicing.

INDEX TERMS 5G, network slicing, profit maximization, slice dimensioning, service function chaining, social welfare.

I. INTRODUCTION

The forthcoming next-generation mobile networks (5G) are envisaged to support diverse application scenarios and services with various requirements on high data rates, low latency, seamless coverage, and dense connectivity, etc. [1]–[3]. However, the monolithic design of the current network architecture (*e.g.* the LTE networks) cannot meet all the service requirements in a cost-efficient way, as it treats all services in the same way (*e.g.* the IoT metering service might not need mobility functions, and thus the related cost can be saved). Instead of building special-purpose networks for individual services, cloud-based network infrastructure can create multiple customized logical networks (network slices) for individual services with *network slicing*, thus saving numerous building and operational cost [2]–[4]. Hence, network slicing is prevailingly perceived as a foundational enabler of 5G networks. Network slicing is based on a stack of novel techniques, including network functions virtualization (NFV) and software-defined networking (SDN) [3]–[7]. NFV splits network functions (NFs) as software modules from proprietary purpose-built hardware [8], [9], while SDN decouples control functions from forwarding nodes and places them on the logically centralized controller [10]. Compared with legacy networks, SDN and NFV enable networks to accommodate new type of services more easily in the future by upgrading the software.

A service provider leases network slices from the Network Slice Provider (NSP) that manages the resources of the network infrastructure [7]. Hence, we also call a service

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

provider as network slice customer (NSC). A network slice carries a group of flows that belong to the end users that subscribe to the service provider. The functionality of a slice (service) is described by a Service Function Chain (SFC) [11], [12], which can be represented as an ordered sequence of NFs. In an SDN/NFV-enabled network, virtual network functions (VNFs) are created on nodes and interconnected by specifying forwarding rules so that flows are processed following the SFC logic [13], [14].

In network slicing, only required function modules will be provisioned for a slice, so the processing efficiency of service flows in a slice can be improved, and thus the Quality of Service (QoS), *e.g.* latency, can be guaranteed more easily. Besides, network slices are isolated from each other, so the QoS of each slice is not influenced by other slices. Furthermore, the role division of NSP and NSCs greatly simplifies the network deployment, operation and management, and improves the flexibility and efficiency for carrying services. The NSP does not need to concern about how to accommodate new type of services on its network. Instead, it can focus on how to deploy and manage its networking and computing resources, for delivering services at the lowest possible cost without compromising service quality. On the other hand, the NSCs realize a service logic by defining the corresponding SFC as well as its resource requirements and utilizing the open interface of network slicing. In this way, network slicing can provide full flexibility for network operators and service providers in a cost-efficient way, and thus has been widely deemed as a fundamental technology for 5G by industry leaders and standardization bodies [3], [4], [7].

Although network slicing brings substantial benefits for provisioning services, there remain some outstanding issues to be addressed. How to efficiently map the SFC to the substrate network has been considered in some work [14]–[17]. The commonality is jointly optimizing *network function placement and flow routing*, so as to minimize network cost and improve network throughput [15], fairness [16], etc. However, existing work does not consider the NSP and NSCs as independent operational entities, *i.e.*, they lack a business model for the NSP and NSCs to maximize their profits. If the service provided by a slice has certain elasticity [18], [19], NSCs can adjust resource demand according to the resource prices, in order to maximize its profit. According to the demand of slices, the NSP can adjust the prices, so as to maximize its profit and improve resource utilization of the substrate network. Therefore, network slices should be carefully dimensioned by appropriately allocating resources to slices, according to the resource availability and the economic mechanism in the network. This inspires us to deeply explore resource pricing in network slice dimensioning.

To the best of our knowledge, this is the prior work that investigates network slicing in conjunction with resource pricing. We consider the coexisted network slices with elastic traffic and ordered service function chains. The slice dimensioning problem is considered in the form of Stackelberg pricing game [20], [21], where the NSCs adjust their resource demand according to the price offered by the NSP, in order to maximize their profit. As the demand of slices depends on the prices, the NSP might first determine a price that can also maximize its own profit. We formulate the optimization problem for the NSP and NSCs that maximize their respective profit (denoted as NSPP and NSCP). We analyze how to find the optimal price for NSP by *backward induction* under the scenario where multiple users share a single link. However, finding the optimal solution is intractable in a complex network setting with numerous constraints on capacity, flow routing, and VNF placement. In this case, we cannot determine the closed-form expression of the demand curve for individual slices and thus the optimal resource price.

In order to perform slice dimensioning efficiently, we propose a two-stage resource pricing algorithm. The first stage aims at maximizing the NSP's profit by searching prices based on resource cost, while the second stage aims to maximize the social welfare of the network so that both the network resource utilization and the profit of NSCs can be improved. In the second stage, the price is determined with the dual variables. All slices are jointly optimized for social welfare (the joint-Network Slice Customers Problem, joint-NSCP in brief), making the problem large-scale, while the variables from each slice are coupled together in the capacity constraints so that the problem cannot be decomposed directly. To cope with this issue, we solve joint-NSCP via its dual form, decompose it on the per-slice basis based on the dual-ADMM method [22]–[24], and finally obtain a price that further improves the NSP's profit. The numerical results show that the proposed pricing algorithm can efficiently solve the slice dimensioning problem and strikes a tradeoff between optimizing the NSP's profit and other metrics including network social welfare, resource utilization, and the profit of NSCs.

The rest of the paper is organized as follows. We discuss the related work in Section II and present the resource and cost model for network slicing in Section III. Next, we formulate the NSCP and NSPP in Section IV and propose the resource pricing algorithm in Section V. Then we present the numerical results and discussions in Section VI. Finally, we conclude this paper in Section VII.

II. RELATED WORK

With the advent of 5G, network slicing enabled wireless network architectures have been proposed by industry leaders and standardization bodies [3], [4], [7]. For instance, Huawei has proposed the MyNET/SONAC platform to realize network slicing for future wireless network [7]. In the MyNET architecture, the Slice Provider builds both control and user plane slices on network infrastructures. Once a Slice Customer requests for a slice, the Slice Provider will create a user plane slice, and associate it with shared control slices or dedicated control slices. An *et al.* [4] study the interaction between user equipments (UEs) and network slices, such as discovering and selecting appropriate slices.

In recent days, the research work on network slicing is mainly focused on the resource allocation aspect, *i.e.*, service function chain deployment and flow routing. Li *et al.* [13] propose a general optimization framework, which can deal with various objectives for SFC deployment, such as minimizing delay, load balancing, etc. Also, the framework can incorporate other technique problems, such as flow routing, so as to improve the flexibility. In [14]–[16] and [25], an SFC is decomposed into segments, so that the order of the function chain can be ensured when performing optimization. Jang *et al.* [15] consider that a service node can deploy multiple VNF instances with fixed capacity and resource consumption as [14]. Then they jointly optimize the flow routing and the VNF placement problem, so as to achieve throughput maximization and energy saving. Zhang *et al.* [16] allow flows to be routed on multi-paths, while a flow's traffic can only be processed by one VNF instance of a specific type, thus avoiding cooperation overhead. With the fixed deployment of VNF instances, Yu *et al.* [26] prove the NP-hardness of the QoS routing problem with specified function chain and then propose an FPTAS algorithm. Usually, the constraints on VNF placement introduce integer design variables, leading the optimization problem into NP-hardness [15], [16]. Therefore, relaxing and rounding techniques and heuristic methods are usually adopted to efficiently solve the problems [14]–[16].

On the other hand, there is also related work on un-ordered service function chains [27]–[29]. Cohen *et al.* [28] consider the cost optimization in VNF placement, which assumes the fixed flow routing and ignore the link resource constraints in the network. Lin *et al.* [27] formulate the end-to-end flow routing and VNF placement problem as a mixed integer program, as they optimize the number of VNF instances that have constant capacities and they enforce single path routing. In [29], we consider the processing resource requirement is proportional to data rate, which is different from [14], [15] and [27]. We have studied the slice dimensioning problem that utilizes pre-calculated multi-paths and optimizes VNF placement and flow routing to maximize the profit of slice provider and slice customers. Although the un-ordered function chain is easier to handle, realistic services usually specify the processing sequence of VNFs according to the service logic, and thus the ordered service chain would be more realistic for most services.

Network slicing is similar to Virtual Network Embedding (VNE) [30]–[32], whereas there are several key differences [9]. The topology of virtual networks in VNE is explicitly known as the virtual link/node resource demand. In network slicing, however, we indeed only know the function chains and the end-to-end service requests [25]. In addition, the endpoints of service requests in network slicing are bound to access points or forwarding nodes, while there might be no such requirement for VNE requests. Besides, most work on VNE is focused on maximizing the acceptance ratio of VNE requests with static resource allocation schemes, with little attention dedicated to the dynamic scaling of virtual networks [9], [33]. Therefore, it is interesting to investigate the difference between VNE and network slicing, in order to design new solutions for network slicing.

The aforementioned work is discussed in the scope of a single operator, which owns or purchases network infrastructures to build network slices (or virtual networks). Therefore, the operator can control the resource allocation to achieve its objective, *e.g.*, minimizing the operational cost or maximizing profit. References [19] and [34]–[36] are focused on maximizing network utility of elastic traffic, and [33] aims to maximize the social welfare of the network, *i.e.*, the user utility minus the resource cost. The network utility related methods provide the insight of exploiting economic mechanism to facilitate resource allocation to individual users/services, so as to achieve fairness and avoid congestion, etc. Different from those work, NSP and NSCs in this paper are independent operators in a cloud-based market, *i.e.*, NSCs carry their services by purchasing or leasing network slices from the NSP who manages the network infrastructures. Therefore, the economic mechanism between NSP and NSCs should be investigated when dimensioning slices, so as to improve the profit level and the network profitability of NSP/NSCs and network resource utilization.

III. SYSTEM MODEL FOR NETWORK SLICING

We consider a scenario where multiple network slices are built upon the infrastructure of an SDN/NFV-enabled network, as shown in Figure 1. There are geographically distributed access nodes (BS1∼BS6) with heterogeneous radio technologies, forwarding nodes (FR1∼FR8), and data centers (DC1∼DC5). VNF instances, including Access Functions (AF1∼AF2) and Network Functions (NF1∼NF4, and Controller), are created at the NFV-capable nodes, such as access nodes, edge-computing servers, and data centers. Enabled by

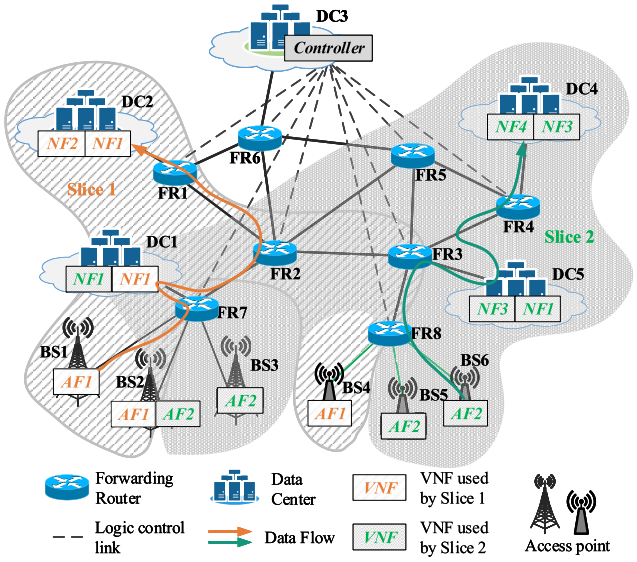


FIGURE 1. Illustration of network slicing.

virtualization technology, VNF instances of the same type can be deployed on multiple physical nodes so as to get close to end users and achieve load balancing, and one physical node can instantiate multiple VNF instances. In the example of Figure 1, two network slices are instantiated on the infrastructure. In a slice, a data flow should be processed by a chain of AFs and NFs. For example, the data flow from BS1 to DC2 in Slice 1 is processed by AF1, NF1 and NF2.

The main focus of this paper is the resource allocation between slices, *i.e.*, *slice dimensioning*. We consider that the network slice provider (NSP) that manages the network infrastructure and provides network slices for network slice customers (NSCs) to carry their services. The network infrastructure is represented as a directed graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N} = \{i | i = 1, \ldots, N\}$ is the node set, $\mathcal{L} = \{e | e = 1, \ldots, L\}$ is the link set (we also use node pair to denote a link, *i.e.*, $e = (i, j)$). The node set can be partitioned into the forwarding node set $\mathcal{N}_R$ and the VNF-capable node set $\mathcal{N}_V$. The VNF-capable node is connected to the proximal forwarding node with high capacity low latency link. Especially, some VNF-capable nodes are co-located with forwarding nodes. We use $\boldsymbol{C} = (C_e : e \in \mathcal{L})$ to denote the bandwidth of links, and the links between VNF-capable nodes and forwarding nodes are assumed to have infinite capacity. The processing capacity of NFV-capable nodes means a collection of computing resources, including CPU cores, memory, and storage etc. [13], [37], [38]. We use $\boldsymbol{V} = (V_i : i \in \mathcal{N}_V)$ to denote the processing capacity of nodes, and the processing capacity of forwarding nodes is zero.

As shown in Figure 1, a set of network slices, denoted by $\mathcal{S}$, are running on the network. The resource description of slice s is denoted by $G^s(\mathcal{N}^s, \mathcal{L}^s, \boldsymbol{w}^s, \boldsymbol{c}^s, \boldsymbol{v}^s)$, where $\mathcal{N}^s$ and $\mathcal{L}^s$ denote the subset of nodes and links used by slice s, and $\boldsymbol{w}^s$, $\boldsymbol{c}^s$, and $\boldsymbol{v}^s$ respectively denote the capacity of virtual nodes, links and VNF instances that allocated to slice s. Suppose that there are F_s flows in slice s, denoted by $\mathcal{F}_s = \{f | f = f_1, \ldots, f_{F_s}\}$. A data flow f in the slice is defined by a tuple (s_f, t_f, r_f), where forwarding nodes s_f, t_f are the source and destination of the flow, and r_f is the data rate.

Data flows in slice s should be processed by the SFC of the slice, which is an ordered sequence of VNFs [15], as shown in Figure 1. Suppose that the VNFs supported by the network are set $\Pi = \{\pi | \pi = 1, 2, \ldots, M\}$. The SFC of slice s is denoted by $\Pi^s = (\pi_1, \pi_2, \ldots, \pi_{M_s})$, which has M_s VNFs from the VNF set Π. To model resource consumption of a SFC, we partition flow f into $(M_s + 1)$ segments, denoted by (f, π_m), $m = 0, 1, \ldots, M_s$ (π_0 is a dummy VNF at source node). The first segment (f, π_0) starts from the source node s_f of flow f and terminates at a VNF instance of type π_1, while the last segment (f, π_{M_s}) starts from a VNF instance of type π_{M_s} and terminates at the destination node of flow f. The other segments (f, π_m) start from a VNF instance of type π_m and terminate at a VNF instance of type π_{m+1}. According to the forwarding rule, a forwarding node decides if a packet from (f, π_m) should be sent to one of the VNF-capable nodes connected to it or directly sent to the next forwarding node,

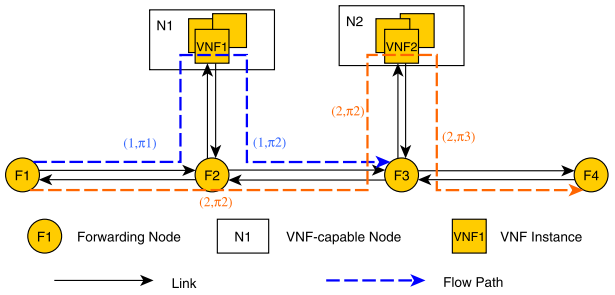


FIGURE 2. Forwarding flows via forwarding nodes and VNF-capable nodes ($(1, \pi_1)$ and $(1, \pi_1)$ are the two segments for flow 1, $(2, \pi_2)$ and $(2, \pi_3)$ are the two segments of flow 2).

as shown in Figure2. When a packet has been processed by the VNF instance on the VNF-capable node and sent back to the forwarding node, it goes into the next segment of the flow, and then be sent to the next forwarding node.

The SFC resource consumption includes link bandwidth and VNF processing capacity. Let $x_e^s(f, \pi_m)$ denote the bandwidth demand on link e of the mth segment of flow f. Then the demand of all flows in slice s on link e is expressed as

$$x_e^s = \sum_f \sum_m x_e^s(f, \pi_m). \tag{1}$$

The *edge-flow* variables $\boldsymbol{x}$ should satisfy the flow conservation law on all nodes. Firstly, at the intermediate forwarding nodes, the incoming traffic equals to the outgoing traffic for each flow segment, *i.e.*,

$$\sum_{e:(j,i)} x_e^s(f, \pi) - \sum_{e:(i,j)} x_e^s(f, \pi) = 0, \quad \forall i \in \mathcal{N}_R^s,\ f \in \mathcal{F}^s,\ \pi \in \Pi^s. \tag{2}$$

Secondly, at source s_f, the outgoing traffic of the first segment (f, π_0) equals to the flow data rate, *i.e.*,

$$\sum_{e:(i,j)} x_e^s(f, \pi_0) - \sum_{e:(j,i)} x_e^s(f, \pi_0) = r_f, \quad \forall f \in \mathcal{F}^s,\ i = s_f. \tag{3}$$

Similar constraints at destination t_f for the last flow segment (f, π_{M_s}) is given by,

$$\sum_{e:(j,i)} x_e^s(f, \pi_{M_s}) - \sum_{e:(i,j)} x_e^s(f, \pi_0) = r_f, \quad \forall f \in \mathcal{F}^s,\ i = t_f. \tag{4}$$

Note that the source and destination nodes may also serve as forwarding nodes for intermediate segments of the flow (corresponding to the Constraint (2)). Lastly, when data of segment (f, π_m) transits a VNF-capable node, it should be processed by type π_{m+1} VNF instance on that node and then transform to the data of next segment (f, π_{m+1}). Thus, the flow constraint at the VNF-capable node is expressed as

$$\sum_{e:(j,i)} x_e^s(f, \pi_m) - \sum_{e:(i,j)} x_e^s(f, \pi_{m+1}) = 0, \quad \forall i \in \mathcal{N}_V^s, \quad f \in \mathcal{F}^s, \quad \pi_m \in \Pi^s \backslash \pi_{M_s}. \tag{5}$$

In order to guarantee the latency requirement of a slice (SFC), we first calculate a group of candidate paths that satisfy its delay budget for each flow. The latency of SFC primarily consists of the link propagation and queueing delay and the VNF processing delay. We have analyzed the VNF processing delay in [39], in which we exploit the parallelization capability of the general processing platform and model the processing delay of VNF instances. According to that work, with the fixed number of VNFs of the SFC in a slice, we can predict the total processing delay of the SFC. Therefore, giving the latency requirement and the predicted SFC processing delay, we can derive the delay budget for the candidate paths. The flow traffic is only allowed to route on the candidate paths, thus guaranteeing the latency requirements.

Let $z_i^s(\pi)$ denote the capacity demand of slice s for VNF type π on VNF-capable node i, which is given by the demand of all flows on this VNF type,

$$z_i^s(\pi) = \sum_{f \in \mathcal{F}^s} z_i^s(f, \pi), \quad \forall \pi \in \Pi^s,\ i \in \mathcal{N}_V^s, \tag{6}$$

where $z_i^s(f, \pi)$ is the capacity demand of a single flow segment (f, π). Thus, the total processing capacity demand of slice s on node i is given by

$$z_i^s = \sum_{\pi \in \Pi^s} z_i^s(\pi). \tag{7}$$

The VNF-capable nodes (*e.g.* data centers) can dynamically manage the VNF instances [40], so that the processing capacity requirement is scaled according to flow data rate. It is also commonly assumed that the computing resource consumption of VNF instances is proportional to the flow data rate [16], [27]. Specifically, one unit of flow data rate requires α_π units of resources for VNF type π (the processing efficiency). As a result, the processing resource requirements of flow f for VNF type π_m on node i is expressed as

$$z_i^s(f, \pi_m) \geq \alpha_{\pi_m} \sum_{e:(j,i)} x_e^s(f, \pi_{m-1}), \quad \forall f \in \mathcal{F}^s,\ \pi_m \in \Pi^s, \tag{8}$$

where the right-hand side is the incoming traffic rate of flow segment (f, π_{m-1}) at the VNF-capable node.

IV. PROBLEM FORMULATION

In this paper, we investigate the network slice dimensioning problem, *i.e.*, how to efficiently allocate resources to individual slices, including determining flow routing and VNF instance placement, assisted by resource pricing method. The problem is modeled as a Stackelberg pricing game [21], where the NSP first sets resource prices, and then the NSCs determine the resource demand of their slices. Specifically, giving the network resource information and service requests from NSCs, in order to maximize NSP's profit and improve resource utilization, the NSP tries to price the resources, so that the elastic demand of slices can be regularized by maximizing the profit of individual NSCs.

The profit of an NSC corresponds to the surplus between the user utility $U_s(\cdot)$ and the payment for resource consumption [19], [41]. For elastic traffic, such as multi-media services, the typical utility function reflects the diminishing marginal profit as the increasing of resources [18], [42]. Besides, in order to achieve proportional fairness, the utility function is logarithmic [21]. Hence, we define the user utility in slice s as

$$U_s(r) = w_s \log(1 + r), \tag{9}$$

where the weight w_s reflects the level of QoS (*e.g.* data rate) in slice s. On the other hand, the payment for resource consumption of slice s is given by

$$\phi_s(\boldsymbol{x}^s, \boldsymbol{z}^s, \boldsymbol{\rho}) = \sum_{i \in \mathcal{N}_V^s} \rho_i z_i^s + \sum_{e \in \mathcal{L}_s} \rho_e x_e^s, \tag{10}$$

where the total capacity demand of slice s, z_i^s and x_e^s are given by (1) and (7), and the price ρ_i for processing resources and the price ρ_e for bandwidth are set by the NSP.

We formulate the network slice customer's problem (NSCP) to maximize the profit of NSC, *i.e.*,

$$\begin{aligned} \max_{(\boldsymbol{z}^s, \boldsymbol{x}^s) \geq 0}\ & Q_S \triangleq \sum_{f \in \mathcal{F}^s} U_s(r_f) - \phi_s(\boldsymbol{x}^s, \boldsymbol{z}^s, \boldsymbol{\rho}) \\ \text{s.t. Constraints}&(1) \sim (8), \end{aligned} \tag{11}$$

where the objective function is the total user utility minus the payment to the NSP, Constraints (2)∼(5) impose the flow conservation law at the forwarding nodes and VNF-capable nodes, and (8) represents the processing resource requirement of data flows on VNF-capable nodes. Note that link and node capacity constraints are not implicitly expressed, as the resource demand is confined by resource prices. Since the utility function is concave and the resource payment is a linear function, the NSCP is a convex optimization problem. With the resource cost in the objective function, the possible routing loops that satisfy the flow conservation law (2)∼(5) can also be eliminated.

In our model, we have no constraint on flow splitting at forwarding nodes, which might introduce higher controlling overhead than that of the single path routing, especially for VNF processing cooperation. Fortunately, with linear resource pricing given by (10), the flows usually are not split.

Lemma 1: Giving the solution of NSCP, the traffic of individual flows is routed on the minimum cost paths.

Proof: cf. Appendix A. Actually, it is practically rare to result in multiple equal-cost paths, since each path has different constituent nodes and links with different offered prices. Therefore, flow splitting is avoided.

On the other hand, the NSP's profit $Q_N(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\rho})$ is the resource payment from all the network slices minus the total physical resource cost, *i.e.*,

$$\begin{aligned} Q_N &= \sum_{s \in \mathcal{S}} \phi_s(\boldsymbol{x}^s, \boldsymbol{z}^s, \boldsymbol{\rho}) - \left(\sum_{i \in \mathcal{N}_V} \varphi_i z_i + \sum_{e \in \mathcal{L}} \varphi_e x_e \right) \\ &= \sum_{s \in \mathcal{S}} \left(\sum_{i \in \mathcal{N}_V^s} (\rho_i - \varphi_i) z_i^s + \sum_{e \in \mathcal{L}^s} (\rho_e - \varphi_e) x_e^s \right). \end{aligned} \tag{12}$$

where φ_i and φ_e are respectively the unit cost for node and link resources, and $z_i = \sum_s z_i^s$, and $x_e = \sum_s x_e^s$ is the aggregated demand of processing capacity and bandwidth respectively. Hence, the network slice provider's problem (NSPP) with the aim to maximize profit is formulated as

$$\max_{(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\rho})\geq 0} \quad Q_N(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\rho}) \tag{13}$$

$$\text{s.t. } z_i \leq V_i, \quad \forall i \in \mathcal{N}_V, \tag{13.1}$$

$$x_e \leq C_e, \quad \forall e \in \mathcal{L}, \tag{13.2}$$

where (13.1) and (13.2) are respectively the node and link capacity constraint. Since the resource demand (z_i^s and x_e^s) is the function of resource price $\boldsymbol{\rho}$, the NSP needs to find the optimal prices that maximize its profit, while the resource demand does not exceed the capacity of network infrastructure.

We illustrate the NSP profit optimization with a simple example: n users (representing n slices) sharing a single link with capacity C. The profit of user j is given by

$$Q_j = w_j \log\left(1 + x_j\right) - \rho x_j, \tag{14}$$

with the data rate $x_j^* = \left[\frac{w_j}{\rho} - 1\right]^+$ to maximize the user's profit. By backward induction, to achieve the maximal profit for NSP, we need to solve the problem

$$\max_{\rho\geq 0} \ (\rho - \varphi)\, x$$
$$\text{s.t.} \quad x = \sum_{i=1}^{n} x_j^* = \sum_{i=1}^{n} \left[\frac{w_i}{\rho} - 1\right]^+,$$
$$x \leq C.$$

Since x is a piecewise function of ρ, we cannot determine the convexity of the objective function, and thus cannot solve the problem directly. Instead, we can divide the feasible region of ρ into sub-intervals and investigating the local optimal solution on each interval. Let the user weight be given by ascending order, *i.e.*, $w_0 = \varphi \leq w_1 \leq w_2 \leq \ldots \leq w_n$ (the user whose weight is less than φ will be dropped since it contributes no profit to the link). In the price interval $\left[w_{j-1},\ w_j\right]$, users whose weight is not greater than w_{j-1} are dropped first, and thus the problem is rewritten as

$$\max_{w_{j-1}\leq\rho\leq w_j} \ (\rho - \varphi)\, x$$
$$\text{s.t. } x = \sum_{i=j}^{n} \left(\frac{w_i}{\rho} - 1\right) \leq C,$$

which is a convex optimization problem and thus could be solved readily. Finally, we compare the objective function value of each interval and determine the optimal price ρ^* from local optimal solutions ρ_j^*.

Under a general network setting, we need to consider the flow reservation law and the VNF placement for individual flows. It is intractable to explicitly express the slice's optimal demand ($x(\boldsymbol{\rho})$ and $z(\boldsymbol{\rho})$) based on the offered prices, not to mention solving the NSPP. From the above example, we know that the NSPP's objective function is even not convex, and thus it is difficult to find the global optimal solution. Without the knowledge of the demand curve of each slice, the NSP cannot determine the price to stay at a local optimal solution (*i.e.*, an equilibrium of the pricing game). Hence, we need to devise a resource pricing algorithm that helps NSP to obtain a near maximal profit.

V. RESOURCE PRICING ALGORITHM FOR NETWORK SLICES DIMENSIONING

The main objective of the pricing algorithm is to maximize the profit of NSP, which is difficult to be found. Therefore, we first perform a heuristic line search to determine a base price for NSP that has maximal profit in the search direction. On the other hand, we try to maximize the social welfare of the network, so that network resources can be efficiently utilized, and thus the profit of NSP can be further improved.

Lemma 2: The data rate of individual data flows is non-increasing with the offered price, i.e., giving the offered price $\boldsymbol{\rho}^+ \succcurlyeq \boldsymbol{\rho}$, the flow data rate ${r_f}^+ \leq r_f$.

Proof: According to (26), the optimal data rate for flow f is ${r_f}^* = \left[\frac{w_s}{\rho_f} - 1\right]^+$, where ρ_f is the unit routing cost for the flow. When we increase the offered price, *i.e.*, $\boldsymbol{\rho}^+ \succcurlyeq \boldsymbol{\rho}$, we have ${\rho_f}^+ \geq \rho_f$ for individual flows, and thus ${r_f}^+ \leq r_f$, which completes the proof. ■

It is easy to see that the traffic demand of individual slices declines with the price increasing. If raising price does not change the minimal-cost path of the flow, the resource demand of the flow is also non-increasing with the offered price, as the demand is proportional to the flow data rate. On the other hand, adjusting price might direct the traffic of a flow to a longer path (*e.g.* more hops on that path) that requires more resource, although the traffic demand is declining. However, the demand increase is only a local phenomenon. As the price continues to increase, the traffic and resource demand will both tends to zero, and thus the profit of NSP will first increase with the rising of price and then decrease with the declining of demand.

In this paper, we first perform a line search of resource price given as Algorithm 1, which will not change the minimal-cost path of flows, so that the resource demand is gradually decreasing. The start price is given as the resource cost $\boldsymbol{\varphi}$ of the substrate network. The trial price is doubled until the profit of NSP Q_k starts declining. Then we perform a tri-section procedure to locate the resource price that maximizes the profit NSP in the search direction. Note that the search results might induce capacity violation as the capacity constraint is not considered when solving NSCPs separately.

Then in the second stage, we try to optimize the social welfare of the network. Giving the base price $\tilde{\boldsymbol{\rho}}$ of the search result, we analyze the joint slice dimensioning problem for NSCs (joint-NSCP), which is simply the combination of individual NSCPs with capacity constraints, giving by

$$\max_{(\boldsymbol{x},\boldsymbol{z})\geq 0} \sum_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}^s} \left\{U_s(r_f) - \left(\sum_{i\in\mathcal{N}_V^s} \tilde{\rho}_i z_i^s + \sum_{e\in\mathcal{L}^s} \tilde{\rho}_e x_e^s\right)\right\}$$
$$\text{s.t.} \quad (1)\sim(8), \quad \forall s \in \mathcal{S}, \text{ and } (13.1),\ (13.2), \tag{15}$$

Algorithm 1 : Line Search of Resource Prices

Require: resource cost vector $\boldsymbol{\varphi}$.
Ensure: resource price $\tilde{\boldsymbol{\rho}}$.
1: Initialize: $\boldsymbol{\rho}_0 \leftarrow \boldsymbol{\varphi}$, $Q_0 = -\infty$.
2: **repeat**
3: $\quad \boldsymbol{\rho}_k \leftarrow 2\boldsymbol{\rho}_{k-1}$.
4: $\quad$ Solve NSCP for each slice, obtain Q_k.
5: **until** $Q_k \leq Q_{k-1}$
6: *tri-section* between $[\rho_{k-2}, \rho_k]$ for $\tilde{\boldsymbol{\rho}}$ that maximizes Q.

where the objective function is the social welfare defined as the total utility from all slices minus the resource payment from slices. Note that the resource payment instead of resource cost of the network is used here, resulting in a trade-off between maximizing the profit of NSP and maximizing the network social welfare. By solving this problem, we aim at deriving the final resource price that further improves the profit of NSP and optimize the network resource utilization.

As the base price $\tilde{\boldsymbol{\rho}}$ cannot guarantee that the capacity constraint is satisfied, we investigate the dual problem of (15) to derive the final price from the dual variables of the capacity constraints. The partial Lagrangian of problem (15) is obtained by converting the constraints (13.1), (13.2), *i.e.*,

$$\begin{aligned} L(\boldsymbol{x}, z, \boldsymbol{\lambda}) &= \sum_{s\in\mathcal{S}} Q_S - \sum_{i\in\mathcal{N}_V} \lambda_i \left(z_i - V_i\right) - \sum_{e\in\mathcal{L}} \lambda_e \left(x_e - C_e\right) \\ &= \sum_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}^s} \Big\{ U_s\big(r_f\big) - \sum_{i\in\mathcal{N}_V} \Big(\left(\lambda_i + \tilde{\rho}_i\right) z_i^s - \frac{\lambda_i V_i}{|\mathcal{S}|} \Big) \\ &\quad - \sum_{e\in\mathcal{L}} \Big(\Big(\lambda_e + \tilde{\rho}_e\Big) x_e^s - \frac{\lambda_e C_e}{|\mathcal{S}|} \Big) \Big\} \\ &= \sum_{s\in\mathcal{S}} L_s(\boldsymbol{x}^s, z^s, \boldsymbol{\lambda}). \end{aligned} \tag{16}$$

Thus, the dual objective function is given by

$$D\left(\boldsymbol{\lambda}\right) = \max_{(\boldsymbol{x},z)\in\boldsymbol{R}_n^+\cap\boldsymbol{\mathcal{X}}} L(\boldsymbol{x}, z, \boldsymbol{\lambda}) = \sum_{s\in\mathcal{S}} D_s(\boldsymbol{\lambda}),$$

where $\boldsymbol{R}_n^+$ is the nonnegative quadrant of the domain, $\boldsymbol{\mathcal{X}}$ is the feasible region given by (1)$\sim$ (8), and

$$D_s\left(\boldsymbol{\lambda}\right) = \max_{(\boldsymbol{x}^s,z^s)\in\boldsymbol{R}_{n_s}^+\cap\boldsymbol{\mathcal{X}}^s} L_s(\boldsymbol{x}^s, z^s, \boldsymbol{\lambda}). \tag{17}$$

with the sub-domain of $\boldsymbol{R}_{n_s}^+ \cap \boldsymbol{\mathcal{X}}^s$ corresponding to each slice. Thus, the dual problem is given by

$$\min_{\boldsymbol{\lambda}\geq 0} D\left(\boldsymbol{\lambda}\right). \tag{18}$$

Note that $D_s(\boldsymbol{\lambda})$ has the same form as NSCP. Especially, giving the dual variables $\boldsymbol{\lambda}$, the resource prices in NSCP are equivalently given by

$$\rho_i = \lambda_i + \tilde{\rho}_i, \;\; \rho_e = \lambda_e + \tilde{\rho}_e, \quad \forall i \in \mathcal{N}_V, \; e \in \mathcal{L}. \tag{19}$$

According to the strong duality and max-min property [43], the dual and primal problems have zero optimality gap, and the optimal solution is reached when $(\boldsymbol{x}, z) = (\boldsymbol{x}^*, z^*)$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$. Therefore, we can solve the problem (15) via the dual problem (18), obtaining the prices (19) that ensures a feasible resource allocation.

Since $D_s(\boldsymbol{\lambda})$ is the result of a maximization problem, we cannot solve problem (18) directly. Usually, this kind of problem can be solved by dual decomposition [24]. However, it is not effective for problem (18). As we shown in Lemma 1, the flow traffic is only routed on the minimal-cost path. If we update the dual variables (resource price) in dual decomposition procedure, the minimal-cost path is likely to be changed, which causes huge demand variation even with small changes on the dual variables. The demand variations in turn lead to drastic changes of dual variables. Therefore, the dual decomposition method tends to fall into fluctuation in our problem. To address this issue, we apply the ADMM to the dual problem (18), *i.e.*, the method of dual-ADMM [22], [23].

Duplicating auxiliary variables $\boldsymbol{\gamma}_s$ from dual variables $\boldsymbol{\lambda}$, we obtain the ADMM form of problem (18), as

$$\min_{\boldsymbol{\gamma}_s\geq 0} \sum_{s\in\mathcal{S}} D_s\left(\boldsymbol{\gamma}_s\right), \quad \text{s.t. } \boldsymbol{\lambda} - \boldsymbol{\gamma}_s = 0, \; s \in \mathcal{S}, \tag{20}$$

The augmented Lagrangian of (20) is given by

$$\mathcal{L}_\sigma\left(\boldsymbol{\gamma}_s, \boldsymbol{\lambda}\right) = \sum_{s\in\mathcal{S}} \Big\{ D_s\big(\boldsymbol{\gamma}_s\big) + \boldsymbol{q}_s^T(\boldsymbol{\lambda} - \boldsymbol{\gamma}_s) + \frac{\sigma}{2} \left\|\boldsymbol{\lambda} - \boldsymbol{\gamma}_s\right\|_2^2 \Big\},$$

where $\boldsymbol{q}_s$ are the multipliers for constraints $\boldsymbol{\lambda} - \boldsymbol{\gamma}_s = 0$ and σ is the coefficient for the quadratic penalty. Then problem (20) is solved following the iteration procedure of ADMM, given as Algorithm 2.

Algorithm 2 : Price Update to Maximize Social Welfare

Require: base price $\tilde{\boldsymbol{\rho}}$, resource capacity $\boldsymbol{V}$.
Ensure: resource price $\boldsymbol{\rho}$, resource allocation $\boldsymbol{V}_s$.
1: initialize: $\boldsymbol{\gamma}_s^{(1)} \leftarrow 0, \boldsymbol{q}_s^{(1)} \leftarrow 0, \forall s \in \mathcal{S}$.
2: **for** $k = 1, 2, \ldots, K$ **do**
3: $\quad$ global dual variables (resource price) update:

$$\boldsymbol{\lambda}^{(k+1)} \leftarrow \operatorname*{argmin}_{\boldsymbol{\lambda}} \sum_{s\in\mathcal{S}} \boldsymbol{\lambda}^T \boldsymbol{q}_s^{(k)} + \frac{\sigma}{2} \sum_{s\in\mathcal{S}} \left\|\boldsymbol{\lambda} - \boldsymbol{\gamma}_s^{(k)}\right\|_2^2.$$

4: $\quad$ slice local variables (resource demand) update:

$$\boldsymbol{\gamma}_s^{(k+1)} \leftarrow \operatorname*{argmin}_{\boldsymbol{\gamma}_s} D_s\big(\boldsymbol{\gamma}_s\big) - \boldsymbol{\gamma}_s^T \boldsymbol{q}_s^{(k)} + \frac{\sigma}{2} \left\|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\gamma}_s\right\|_2^2.$$

5: $\quad$ multipliers update (resource re-allocation):

$$\boldsymbol{q}_s^{(k+1)} \leftarrow \boldsymbol{q}_s^{(k)} + \sigma\big(\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\gamma}_s^{(k+1)}\big).$$

6: **end for**
7: set price as $\boldsymbol{\rho} \leftarrow \tilde{\boldsymbol{\rho}} + \boldsymbol{\lambda}^{(K+1)}$.
8: set resource capacity as $\boldsymbol{V}_s \leftarrow \frac{\boldsymbol{V}}{|\mathcal{S}|} - \boldsymbol{q}_s^{(k+1)}$.

In each iteration of the algorithm, the first step calculates the resource price, which can be explicitly represented as

$$\boldsymbol{\lambda}^{(k+1)} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\boldsymbol{\gamma}_s^{(k)} - \frac{1}{\sigma|\mathcal{S}|}\sum_{s\in\mathcal{S}}\boldsymbol{q}_s^{(k)}.$$

Then in the second step, the price is broadcast to each slice to compute the local variables $\boldsymbol{\gamma}_s^{(k+1)}$, according to Lemma 3.

Lemma 3: Giving the primal problem (15) has non-empty solution set, $\boldsymbol{\gamma}_s^{(k+1)}$ is given by

$$\boldsymbol{\gamma}_s^{(k+1)} = \left[\boldsymbol{\lambda}^{(k+1)} + \frac{1}{\sigma}\left(c_s\left(\boldsymbol{u}_s^{(k+1)}\right) + \boldsymbol{q}_s^{(k)}\right)\right]^+, \tag{21}$$

where $c_s(\boldsymbol{u}_s)$ is the negative residual capacity, given by

$$c_s(\boldsymbol{u}_s) = \begin{bmatrix} \left(z_i^s - \dfrac{V_i}{|\mathcal{S}|}\right)_{\forall i\in\mathcal{N}_V} \\ \left(x_e^s - \dfrac{C_e}{|\mathcal{S}|}\right)_{\forall e\in\mathcal{L}} \end{bmatrix}, \tag{22}$$

and $\boldsymbol{u}_s^{(k+1)} = \left(\boldsymbol{x}^{(k+1)}, \boldsymbol{z}^{(k+1)}\right)$ is a solution of the following maximization problem,

$$\max_{\boldsymbol{u}\in\boldsymbol{R}_{n_s}^+\cap\boldsymbol{\mathcal{X}}_s} Q_s(\boldsymbol{u},\tilde{\boldsymbol{\rho}}) - \frac{\sigma}{2}\left\|\left[\boldsymbol{\lambda}^{(k+1)} + \frac{1}{\sigma}\left(c_s(\boldsymbol{u}) + \boldsymbol{q}_s^{(k)}\right)\right]^+\right\|_2^2. \tag{23}$$

Proof: cf. Appendix A. According to Lemma 3, the item $\boldsymbol{q}_s^{(k)}$ in $c_s\left(\boldsymbol{u}_s^{(k+1)}\right)+\boldsymbol{q}_s^{(k)}$ actually balances the capacity violation of each slice [22]. Specifically, if $\boldsymbol{q}_s^{(k)}$ is positive, the corresponding capacity in (22) (*i.e.*, $\frac{V_i}{|\mathcal{S}|}$ and $\frac{C_e}{|\mathcal{S}|}$) is balanced out by $\boldsymbol{q}_s^{(k)}$. Otherwise, if $\boldsymbol{q}_s^{(k)}$ is negative, the resource amount that can be used by the slice without a penalty is increased. On the other hand, the L_2 norm of $\boldsymbol{\gamma}_s^{(k+1)}$ is the penalty in problem (23), which consists of two parts, *i.e.*, the broadcast resource price and the resource capacity violation weighted by coefficient $\frac{1}{\sigma}$. Therefore, $\boldsymbol{\gamma}_s^{(k+1)}$ is also an indicator of resource demand of slice s. Hence, in the last step of the iteration, the resource demand information is also retrieved by updating the multipliers with $\boldsymbol{\gamma}_s^{(k+1)}$. Base on the above analysis, the computation of the broadcasting price in the first step is actually based on the resource demand of all slices.

In Algorithm 2, the L_2-norm penalty enforces flow traffic to be distributed on multiple paths with different base prices, thus addressing the convergence issue that exists in dual decomposition. At the same time, the network social welfare is improved as more resource can be utilized by flows compared with single path routing in Algorithm 1. Hence, the profit of NSP is further improved with the price also being increased.

Remark: (1) We set a fixed number of iterations for Algorithm 2 ($K = 30$ in our experiments), as the ADMM algorithm usually converges slowly to the optimal solution and we only need a close optimal solution. (2) In this situation, the resource allocation might slightly violate the capacity constraints. So, we need to scale down the resource allocation to obtain a feasible solution. (3) As a result of Algorithm 2, the flows might be split, which is not desirable for *mice-flows* that have low data rate and short lifetime due to the coordinate overhead. Therefore, redundant paths should be removed from the solution according to the specification of slice on allowed number of paths. The resources of redundant paths can be re-allocated to remained paths to improve the profit of NSP and the network resource utilization.

Finally, we briefly discuss the computational complexity of the proposed resource pricing algorithm. The number of iterations of the loop in Algorithm 1 (Lines 2∼5) is bounded by $N_1 = \lceil \log_2 \max_{s,p} \frac{w_s}{\varphi_p} \rceil$, where w_s is the weight of slice s and φ_p is the cost of the path in that slice. The number of iterations for tri-section in Algorithm 1 (Line 6) is given by $N_2 = \log_{\frac{3}{2}} \frac{|\rho_k - \rho_{k-2}|}{\epsilon}$, where ϵ is the stop criterion for tri-section. Given the weight of slices and the stop criteria, N_1, N_2 are constant values. The number of iterations in Algorithm 2 is fixed as K. The two algorithms need to solve subproblems, *i.e.*, (11) and (23) respectively in a distributed manner. The subproblems are convex and thus have polynomial time complexity, denoted by $\mathcal{O}(n^a)$, where n is related to the number of variables and constraints in the sub-problem, and a is a constant number. In summary, the proposed pricing algorithm has the time complexity of $(N_1+N_2+K)\cdot\mathcal{O}(n^a)$, and thus is still classified into $\mathcal{O}(n^a)$ polynomial algorithms.

VI. PERFORMANCE EVALUATION

In this section, we use numerical results to demonstrate the benefits of the proposed resource pricing algorithm for network slice dimensioning (represented by "*DualPricing*" in the following). We use the usage-based pricing method in [29] (represented by "*UsageBased*") and two heuristic resource pricing methods as comparison references. The first heuristic also searches prices as Algorithm 1, while the difference is that it will try a higher price if there are resource capacity violations, instead of performing Algorithm 2 (represented by "*Searching*"). The second one sets a fixed price for each resource based on the resource cost (set as 5 times of the resource cost in the experiments) as a baseline (represented by "*FixPrice*"). When resource capacity is violated with the fixed price, the *FixPrice* method partitions resources to individual slices proportionating to their demands.

A. NETWORK CONFIGURATION

The experiments are conducted in a network with 15-node topology shown in Figure 3. The data centers are co-located with some forwarding nodes. All links have a capacity of 1000*Mbps* and cost of 0.05 monetary unit, while the data centers respectively have capacities of $[12, 12, 12, 18, 9, 6] \times 10^3$ processing units and the cost of $[42, 42, 42, 28, 56, 83] \times 10^{-3}$ monetary unit, inverse to the capacity [44].

There are three types of slices for different services, *i.e.*, Content Cache, VPN Access, and Video Chat. Based on [45]–[48], we summarize the specifications of the SFC of

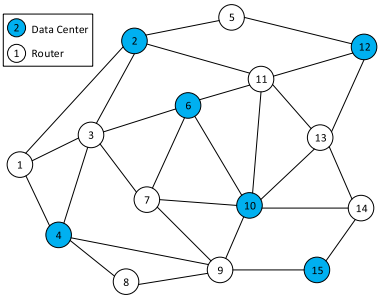


FIGURE 3. Substrate network topology.

TABLE 1. Service function chain of services.

Service	Service Function Chain
Content Cache	(Cache)–Firewall–NAT–(Content)
VPN Access	(UE)–Firewall–Encipher–IPS–NAT–(Server)
Video Chat	(UE)–Firewall–IPS–Transcoder–(UE)

TABLE 2. Slice parameters.

Parameters	Content Cache	VPN Access	Video Chat
User utility weight	10	50	300
Number of paths	1	2	3
Number of flows	75	30	4

TABLE 3. VNF processing efficiency.

Parameters	Firewall	NAT	Encipher	IPS	Transcoder
α_π	1	0.5	2.5	2	1.5

the services in Table 1 and the corresponding slice's parameters are given in Table 2. The data flows in each slice are generated with randomly selected end nodes. The processing efficiency parameters of VNFs used by slices are specified in Table 3.

B. NUMERICAL RESULTS

To demonstrate the performance of our proposed resource pricing algorithm, we conduct experiments in four scenarios. In the first three scenarios, we have one type of slices, *i.e.*, Content Cache, VPN Access, and Video Chat respectively, in the network. In the last scenario, we have equal number of the three types of slices in the network. In the experiments, the total number of slices ranges from 3 to 24. The performance metrics investigated include: the network social welfare, the resource utilization, the profit of NSP, and the profit of NSCs.

Figure 4 shows the network social welfare of the compared methods under the four scenarios. We can observe that the social welfare of our proposed *DualPricing* method is higher than that of the *UsageBased* method and the *Searching* price method. As the *Searching* method does not consider social welfare, when the demand increases with the number of slices, the *Searching* method simply raises the price, which limits the resource demand of slices. Instead, our *DualPricing* method uses the quadratic penalty to direct partial traffic to alternative paths, so that more network resource can be utilized and thus the social welfare is improved. The reason why the *UsageBased* method obtains lower social welfare is that it adopts the quadratic pricing policy [29], which tends to more flow split in order to achieve load balancing. As we limit the flow split in slices, the part of the traffic demand is dropped from the solution of the *UsageBased* method, hence resulting in lower social welfare. The baseline *FixPrice* method actually adopts a relatively lower price and the demand of slices is not much limited by the price, so that it can achieve social welfare a little higher than that of the *DualPricing* method when traffic is light. When the number of slices increases and there exist resource contentions, the *FixPrice* method partitions obtains lower social welfare.

Figure 5 shows the network resource utilization of the compared methods under the four scenarios. Figure 5 verifies that the network resource utilization is correlated with the social welfare that indicates the benefit the network produces by consuming resources. The exception is the *Searching* method whose resource utilization first increases and then declines. This method increases prices of all resources, which also discourages those flows that are not using the congested resource, leading to the decline of resource utilization.

Next, we compare the profit of NSP in Figure 6 for the four methods under different scenarios. The profit of the proposed *DualPricing* method and the *Searching* method has close profit and much higher than that of the other two methods. In some cases (Figure 6a), the proposed *DualPricing* method's profit is lower than that of the *Searching* method. The reason

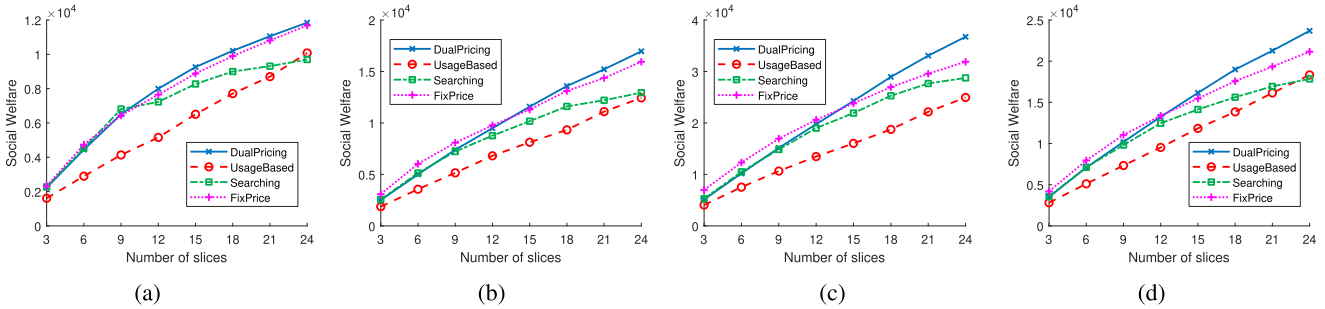


FIGURE 4. Network social welfare under different constituent slices. (a) Content cache slices. (b) VPN access slices. (c) Video chat slices. (d) Three types of slices.

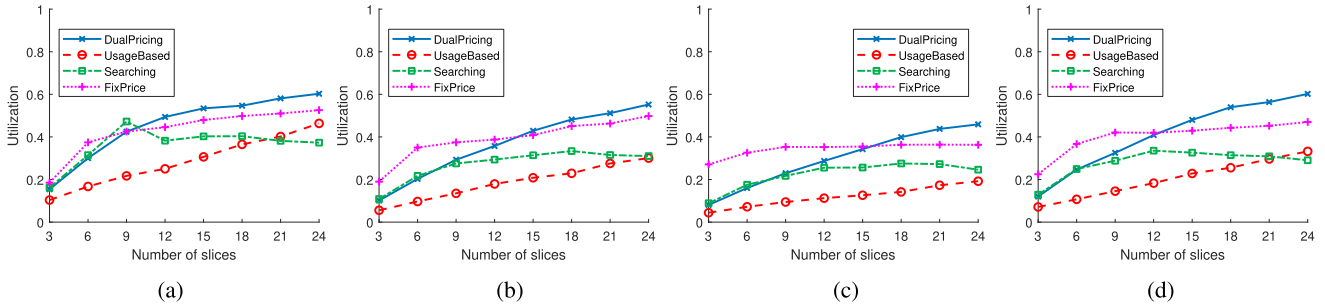


FIGURE 5. Network resource utilization under different slice combinations. (a) Content cache slices. (b) VPN access slices. (c) Video chat slices. (d) Three types of slices.

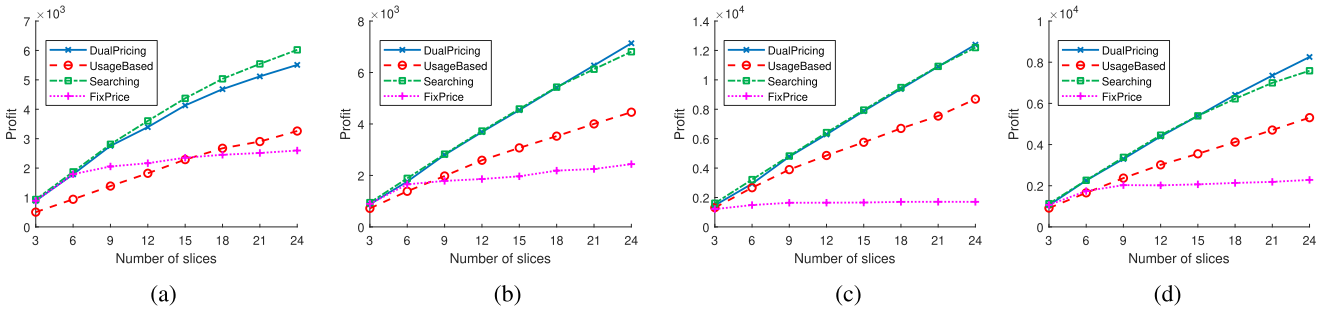


FIGURE 6. Network slice provider's profit under different slice combinations. (a) Content Cache slices. (b) VPN Access slices. (c) Video Chat slices. (d) Three types of slices.

is also related to flow split, as we drop partial split flow in the solution of the *DualPricing* method, especially for the Content Cache slices, where we can only keep one path for each flow. Observing the trends of the curves, we can see that as the resource becomes scarce when more slices are added to the network, the profit form the proposed *DualPricing* method keeps increasing, while the profit growth of the *Searching* method gradually slows down. As we have analyzed for the resource utilization, although the resource price increase, the resources are not fully utilized with the *Searching* method, and thus the profit increases slowly. Influenced by flow split, the *UsageBased* method achieves lower profit similar to its performance on social welfare. For the *FixPrice* methods, as the price is fixed and relatively low, the resources are quickly saturated by elastic slices, the achieved profit tends to a constant and is much than that of other methods.

Finally, we illustrate the profit of NSCs for each type of services under the compared methods. Here, we only show the fourth scenario as it is more realistic, as different type of slices coexist in one network. The average profit of each type of slices is given by Figure 7. The profit of an elastic slice is closely related to its user utility weight and the number of flows in the slice. From the *profit*-axis, we can see that the Video Chat slices have a much higher profit than that of the other two types of slices, as the Video Chat slices have very high utility weight, although it only has several flows in each slice. On the other hand, the *log*-based utility guarantees proportional fairness for all slices, so that even the weight of the Content Cache slices is relatively low, it can

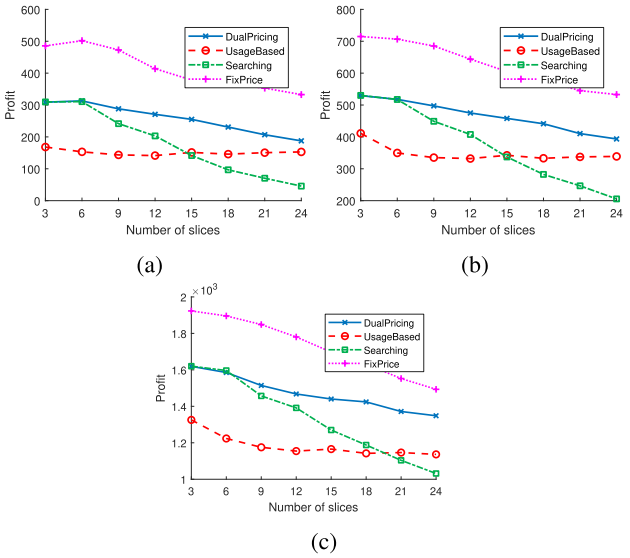


FIGURE 7. Network slice customer's profit. (a) Content cache slices. (b) VPN access slices. (c) Video chat slices.

still obtain resources and gain profit. Comparing the four methods, we find that the *FixPrice* method makes the profit of NSCs much higher. Comparing with the profit of NSP shown in Fig. 6, it is easy to find that the profit loss of NSP has been transferred to the profit of NSCs, making this pricing method impractical to be adopted by NSP. Instead, the proposed

DualPricing method has a better balance between maximizing the profit of NSP and maximizing the profit of NSCs, so that both sides are willing to adopt this pricing policy. For the *Searching* method, its progressive pricing policy is on the contrary of the *FixPrice* method, which cannot be accepted by NSCs. Besides, the performance of the *UsageBased* method has lower profit for NSCs and the NSP, it is totally outperformed by our proposed algorithm, as it is originally designed for un-ordered service function chain and does not consider the flow split issue.

VII. CONCLUSIONS

In this paper, we have investigated resource pricing for dimensioning network slices with elastic traffic and ordered service function chains. We have developed an optimization framework in the form of Stackelberg pricing game and have found that both optimize the profit of NSCs and the profit of the NSP is intractable in general network settings. Hence, we proposed a resource pricing algorithm that seeks a tradeoff between maximizing NSP's profit and the network social welfare. The numerical results show that with the proposed pricing algorithm, the profit of the NSP is higher than other methods in most cases, while the profit of NSCs is acceptable. In addition, the proposed method can achieve higher network social welfare and better resource utilization. Therefore, our proposed method can efficiently dimension elastic network slices.

APPENDIX A
PROOF OF LEMMA I: ROUTING ON MINIMUM-COST PATH

Proof: A feasible path of a flow connects the source and destination of the flow, with the required VNFs placed on some nodes on that path. The set of feasible paths of flow f is given by $\mathcal{P}_f$. Thus, NSCP can be reformulated as

$$\max_{r_p \geq 0} \sum_{f \in \mathcal{F}^s} \left\{ w_s \log\left(1 + \sum_{p \in \mathcal{P}_f} r_p\right) - \sum_{p \in \mathcal{P}_f} \rho_p r_p \right\}, \tag{24}$$

where the price of path ρ_p is determined by the prices of the constituent links and VNFs of the path. The optimality condition for (24) should include,

$$\begin{cases} \dfrac{w_s}{1 + \sum_{p \in \mathcal{P}_f} r_p} - \rho_p - \lambda_p = 0 \\ \lambda_p r_p = 0, \quad \forall p \in \mathcal{P}_f,\ f \in \mathcal{F}_s. \end{cases} \tag{25}$$

Considering a path with data rate $r_p > 0$, *i.e.*, $\lambda_p = 0$ due to complementary slackness, we have $\sum_{p \in \mathcal{P}_f} r_p = \frac{w_s}{\rho_p} - 1,\ \forall p \in \mathcal{P}_f (\rho_p < w_s)$. Hence for two paths $p, \tilde{p} \in \mathcal{P}_f$ if $r_p, r_{\tilde{p}} > 0$, we have $\rho_p = \rho_{\tilde{p}}$, *i.e.*, the traffic is routed on equal-cost paths. Thus, the problem (24) can be rewritten as

$$\max_{r_f \geq 0} \sum_{f \in \mathcal{F}_s} \left\{ w_s \log\left(1 + r_f\right) - \rho_f r_f \right\}, \tag{26}$$

where $r_f = \sum_{p \in \mathcal{P}_f} r_p$ is the flow rate. It is obvious that $\rho_f = \min_{p \in \mathcal{P}_f} \rho_p$ for maximizing the profit, *i.e.*, the traffic is routed on the minimum-cost paths, which completes the proof. ■

APPENDIX B
PROOF OF LEMMA II: COMPUTING THE LOCAL SOLUTION

Proof: To obtain the value of $\gamma_s^{(k)}$ (we drop the superscript k and subscript s for simplicity), we need to solve the problem

$$\begin{aligned} &\min_{\boldsymbol{\gamma}} \left\{ D(\boldsymbol{\gamma}) - \boldsymbol{\gamma}^T \boldsymbol{q} + \frac{\sigma}{2} \left\| \boldsymbol{\lambda} - \boldsymbol{\gamma} \right\|_2^2 \right\} \\ &= \min_{\boldsymbol{\gamma}} \sup_{\boldsymbol{u}} \left\{ Q(\boldsymbol{u}) - \boldsymbol{\gamma}^T \left(c(\boldsymbol{u}) + \boldsymbol{q} \right) + \frac{\sigma}{2} \left\| \boldsymbol{\lambda} - \boldsymbol{\gamma} \right\|_2^2 \right\} \\ &= \min_{\boldsymbol{\gamma}} \sup_{\boldsymbol{u}} \hat{L}(\boldsymbol{u}, \boldsymbol{\gamma}), \quad \boldsymbol{u} = (\boldsymbol{x}^s, \boldsymbol{z}^s). \end{aligned} \tag{27}$$

Since $Q(\boldsymbol{u})$ is concave, $\hat{L}(\cdot, \boldsymbol{\gamma})$ is a concave function for $\boldsymbol{u}$ with fixed $\boldsymbol{\gamma}$. If any component of $\boldsymbol{u}$ goes to $+\infty$, the value of $-\hat{L}(\cdot, \boldsymbol{\gamma})$ will go to $+\infty$. Thus, for any sublevel sets $S_\alpha = \left\{ \boldsymbol{u} \middle| -\hat{L}(\cdot, \boldsymbol{\gamma}) \leq \alpha \right\},\ \alpha < +\infty$, there is an upper bound for each component of $\boldsymbol{u}$. In addition, $\boldsymbol{u}$ is also lower bounded by $\mathbf{0}$ in our problem. Hence, all the non-empty sublevel sets are bounded. Therefore, the recession cone of S_α is empty, and thus there is no common direction of recession for all sublevel sets, and hence no common direction of recession for all functions $-\hat{L}(\cdot, \boldsymbol{\gamma}), \boldsymbol{\gamma} \geq 0$ [49]. Similarly, there is no recession direction for all functions $\hat{L}(\boldsymbol{u}, \cdot)$. Therefore, according to [49, Ths. 37.3 and 37.6], $\hat{L}(\boldsymbol{u}, \boldsymbol{\gamma})$ has a saddle point $(\bar{\boldsymbol{u}}, \bar{\boldsymbol{\gamma}})$ with finite value, and the *strong max-min property* holds, *i.e.*,

$$\begin{aligned} &\min_{\boldsymbol{\gamma}} \sup_{\boldsymbol{u}} \hat{L}(\boldsymbol{u}, \boldsymbol{\gamma}) \\ &= \hat{L}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{\gamma}}) = \max_{\boldsymbol{u}} \inf_{\boldsymbol{\gamma}} \hat{L}(\boldsymbol{u}, \boldsymbol{\gamma}) \\ &= \max_{\boldsymbol{u}} \min_{\boldsymbol{\gamma}} \left\{ Q(\boldsymbol{u}) - \boldsymbol{\gamma}^T \left(c(\boldsymbol{u}) + \boldsymbol{q}^{(k)} \right) + \frac{\sigma}{2} \left\| \boldsymbol{\lambda} - \boldsymbol{\gamma} \right\|_2^2 \right\}. \end{aligned} \tag{28}$$

With a fixed value of $\boldsymbol{u}$, the inner minimization problem of (28) can be solved with the optimal solution,

$$\boldsymbol{\gamma}^* = \left[\boldsymbol{\lambda} + \frac{1}{\sigma} \left(c(\boldsymbol{u}) + \boldsymbol{q} \right) \right]^+, \tag{29}$$

i.e., the equation (21). Substituting the optimal solution $\boldsymbol{\gamma}^*$ into (28) to replace the inner minimization problem and simplify it, we have the exact form of (23) in Lemma 3. Then we solve problem (23) and obtain the primal variables $\boldsymbol{u}_s^{(k+1)}$. Finally, we substitute $\boldsymbol{u}_s^{(k+1)}$ into (21) and obtain the value of $\boldsymbol{\gamma}_s^{(k+1)}$, which completes the proof. ■

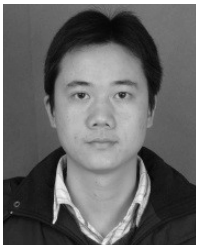
REFERENCES

[1] *5G White Paper*, NGMN Alliance 5G Initiative Team, Frankfurt, Germany, 2014.
[2] *5G Systems*, Ericsson AB, Stockholm, Sweden, 2015.
[3] M. Iwamura, "NGMN view on 5G architecture," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, Scotland, May 2015, pp. 1–5.
[4] X. An *et al.*, "On end to end network slicing for 5G communication systems," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 4, Apr. 2017, Art. no. e3058.
[5] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Trans. Netw. Serv. Manage.*, vol. 13, no. 3, pp. 462–476, Sep. 2016.
[6] I. F. Akyildiz, S.-C. Lin, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Comput. Netw.*, vol. 93, pp. 66–79, Dec. 2015.

[7] H. Zhang *et al.*, "5G wireless network: MyNET and SONAC," *IEEE Netw.*, vol. 29, no. 4, pp. 14–23, Jul./Aug. 2015.
[8] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Commun. Mag.*, vol. 54, no. 4, pp. 84–91, Apr. 2016.
[9] 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.
[10] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.
[11] J. Garay, J. Matias, J. Unzilla, and E. Jacob, "Service description in the NFV revolution: Trends, challenges and a way forward," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 68–74, Mar. 2016.
[12] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.
[13] Y. Li, F. Zheng, M. Chen, and D. Jin, "A unified control and optimization framework for dynamical service chaining in software-defined NFV system," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 15–23, Dec. 2015.
[14] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.
[15] 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.
[16] N. Zhang, Y.-F. Liu, H. Farmanbar, T.-H. Chang, M. Hong, and Z.-Q. Luo, "Network slicing for service-oriented networks under resource constraints," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, Nov. 2017.
[17] R. Wen *et al.*, "On robustness of network slicing for next-generation mobile networks," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 430–444, Jan. 2019.
[18] J. Huang and L. Gao, *Wireless Network Pricing*, vol. 6, no. 2, J. Walrand, Ed. San Rafael, CA, USA: Morgan & Claypool, 2013, pp. 1–176.
[19] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, 1997.
[20] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds., *Algorithmic Game Theory* (Lecture Notes in Computer Science). New York, NY, USA: Cambridge Univ. Press, 2007.
[21] T. Başar and R. Srikant, "A Stackelberg network game with a large number of followers," *J. Optim. Theory Appl.*, vol. 115, no. 3, pp. 479–490, 2002.
[22] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Comput. Optim. Appl.*, vol. 1, no. 1, pp. 93–111, Oct. 1992.
[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[24] S. Boyd, L. Xiao, and A. Mutapcic, "Notes on decomposition methods," Appl. Note EE392o, 2003, pp. 1–6.
[25] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., Apr. 2015, pp. 1–9.
[26] R. Yu, G. Xue, and X. Zhang, "QoS-aware and reliable traffic steering for service function chaining in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2522–2531, Nov. 2017.
[27] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Optimal network function virtualization realizing end-to-end requests," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.
[28] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, Apr./May 2015, pp. 1346–1354.
[29] 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. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.
[30] R. Mijumbi, J. Serrat, J.-L. Gorricho, and R. Boutaba, "A path generation approach to embedding of virtual networks," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 3, pp. 334–348, Sep. 2015.
[31] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quarter, 2013.
[32] R. Wen, G. Feng, W. Tan, R. Ni, S. Qin, and G. Wang, "Protocol function block mapping of software defined protocol for 5G mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1651–1665, Jul. 2018.
[33] T. Ghazar and N. Samaan, "Pricing utility-based virtual networks," *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 2, pp. 119–132, Jun. 2013.
[34] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.
[35] M. Saad, A. Leon-Garcia, and W. Yu, "Optimal network rate allocation under end-to-end quality-of-service requirements," *IEEE Trans. Netw. Service Manage.*, vol. 4, no. 3, pp. 40–49, Dec. 2007.
[36] S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.
[37] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.
[38] R. Riggio, A. Bradai, D. Harutyunyan, T. Rasheed, and T. Ahmed, "Scheduling wireless virtual networks functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 240–252, Jun. 2016.
[39] M. Liu, G. Feng, J. Zhou, and S. Qin, "Joint two-tier network function parallelization on multicore platform," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–7.
[40] F. Z. Yousaf and T. Taleb, "Fine-grained resource-aware virtual network function management for 5G carrier cloud," *IEEE Netw. Mag.*, vol. 30, no. 2, pp. 110–115, Mar. 2016.
[41] X. Wang and H. Schulzrinne, "Pricing network resources for adaptive applications," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 506–519, Jun. 2006.
[42] C. J. Van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatiotemporal model of the human visual system," *Proc. SPIE*, vol. 2668, pp. 450–461, Mar. 1996.
[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[44] H. Farmanbar and H. Zhang, "Cross-layer traffic engineering for software-defined radio access networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3411–3416.
[45] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017,
[46] W. Haeffner, J. Napper, M. Stiemerling, D. R. Lopez, and J. Uttaro, *Service Function Chaining Use Cases in Mobile Networks*, Internet Draft, Vodafone, Düsseldorf, Germany, 2018, pp. 1–26. [Online]. Available: https://tools.ietf.org/html/draft-ietf-sfc-use-case-mobility-08
[47] J. Duan, C. Wu, F. Le, A. X. Liu, and Y. Peng, "Dynamic scaling of virtualized, distributed service chains: A case study of IMS," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2501–2511, Nov. 2017.
[48] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.
[49] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.

GANG WANG received the B.E. degree in communication engineering from the University of Electronic Science and Technology of China, where he is currently pursuing the Ph.D. degree with the National Key Laboratory of Science and Technology on Communications. His research interests include resource allocation in communication networks, especially traffic engineering, software-defined networking, and network virtualization technologies.

GANG FENG (M'01–SM'06) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong, in 1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University, in 2000, as an Assistant Professor and was promoted as an Associate Professor in 2005. He is currently a Professor with the National Laboratory of Communications, UESTC. He has extensive research experience and has published widely in computer networking and wireless networking research. His research interests include resource management in wireless networks and next-generation cellular networks.



SHUANG QIN received the B.E. degree in electronic information science and technology and the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China (UESTC), in 2006 and 2012, respectively.

He is currently an Associate Professor with National Key Laboratory of Science and Technology on Communications, UESTC. His research interests include cooperative communication in wireless networks, data transmission in opportunistic networks, and green communication in heterogeneous networks.

RUIHAN WEN received the B.E. degree in communication engineering from Jilin University, in 2010, and the M.E. degree in electronic information science and technology from the University of Electronic Science and Technology of China (UESTC), in 2013, where she is currently pursuing the Ph.D. degree with the National Key Laboratory of Science and Technology on Communications.

Her research interests include resource management and network virtualization technology in future networks.

SANSHAN SUN received the B.E. degree in electrical engineering and automation from Sichuan Normal University, in 2006, and the M.E. degree in communication and information systems from the Chongqing University of Posts and Telecommunications, in 2009. He is currently pursuing the Ph.D. degree with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China. His research interests include resource management and game theory.