Joint Resource and Admission Management for Slice-enabled Networks

Sina Ebrahimi*, Abulfazl Zakeri*, Behzad Akbari*, Nader Mokari*

Faculty of Electrical and Computer Engineering (ECE)
Tarbiat Modares University
Tehran, Iran
{sina.ebrahimi, abolfazl.zakeri, b.akbari, nader.mokari}@modares.ac.ir

Abstract—Network slicing is a crucial part of the 5G networks that communication service providers (CSPs) seek to deploy. By exploiting three main enabling technologies, namely, software-defined networking (SDN), network function virtualization (NFV), and network slicing, communication services can be served to the end-users in an efficient, scalable, and flexible manner. To adopt these technologies, what is highly important is how to allocate the resources and admit the customers of the CSPs based on the predefined criteria and available resources. In this regard, we propose a novel joint resource and admission management algorithm for slice-enabled networks. In the proposed algorithm, our target is to minimize the network cost of the CSP subject to the slice requests received from the tenants corresponding to the virtual machines and virtual links constraints. Our performance evaluation of the proposed method shows its efficiency in managing CSP's resources.

Index Terms—Network Slicing, 5G, Resource Allocation, Admission Control, CSP, Tenants, Slice Requests, NFV.

I. INTRODUCTION

With the rise of the fourth industrial revolution in recent years, telecommunication industry leaders e.g. international telecommunication union (ITU) have tried to open the room for vertical industries by innovations in defining new services through the fifth generation of mobile networks (5G) paradigm.ITU has defined some stringent contradicting requirements on reliability, latency, and throughput for 5G [1].

To satisfy the aforementioned diverse requirements of IMT-2020 [1], the next generation mobile networks alliance (NGMN) has come up with the network slicing concept in which multiple logical networks could be deployed on a single physical network [2], [3]. The main target of network slicing for a communication service providers (CSPs) is to prepare an intelligent and flexible network which can support three (or more) conflicting types of services such as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (URLLC) [1]. The main goal of this paper is allocating the resources of each CSP to the received slice requests of its tenants by leveraging network function virtualization (NFV) and software-defined networking (SDN).

A. Related Works

Herein, we divide the related works into two major categories in terms of network architecture and resource allocation in network slicing.

1) Network Architecture: In [4], the authors introduce a network slicing architecture towards 5G. The authors in [5], discuss the evolution of the network sharing concept towards an architecture of network slicing. They propose a slice broker architecture to manage the CSPs slices which belong to different tenants. In [6]–[8], SDN, NFV, and orchestration are introduced as the most relevant technologies to elevate the architecture of network slicing. In this paper, the high-level architecture is inspired by the infrastructure sharing aspect of network slicing with an eye on QoS of different tenants [6], [8]. Moreover, the multi-tenancy framework in this paper is inspired by [8].

2) Resource Allocation in Network Slicing: [9] is a comprehensive survey about resource allocation in network slicing. In [10], the problem of network slice broker is formulated in order to provide a joint internet of things (IoT)/slice broker orchestration scheme. This approach, known as "Slice as a Service" (SlaaS) [10], inherently improves the resource utilization rate. An admission and allocation problem of slice requests in order to maximize the revenue of CSP and satisfy the tenants' service requirements is discussed in [11]. In [12], the authors take the SlaaS approach and develop an online genetic-based optimizer that approaches toward the ideal slicing strategy with the maximized long-term network utility.

In [13], the authors introduce a resource allocation problem by adapting a matching game for a mobile virtual network operator, who buys physical resources from the CSP and bundle them into virtual resources called slices. An integer linear program for offline mobile network slice embedding, focusing on resource allocation and virtual node and link mapping is proposed in [14], to maximize the weighted sum of all embedded slices in the physical network. In [15], the authors study the network slice dimensioning problem with resource pricing policy by exploring the relationship of the CSP (which is the slice provider) and the tenants (which are slice customers). Moreover, a network slicing framework, including admission control, resource allocation, and user dropping is presented in [16], which use a game-theoretic approach to solve the problem.

Although there are some works in the literature about resource allocation in network slicing, there are still many challenges in this area that are not completely solved. For

example, Resource pricing, working on end-to-end latency, examining the roles of the new players, and providing admission control mechanisms are some important challenges about resource allocation for network slicing. Resource allocation frameworks for network slicing have mostly failed to consider the relationship of the CSP with its tenants, e.g., third-parties, and vertical industries thoroughly. Moreover, despite envisioning slice request-based architectures in [12] and [15], they do not propose any admission control mechanisms to ensure the feasibility of their problems.

B. Contributions

In this paper, we propose a new SlaaS framework for a CSP which intends to allocate its own resources to the slice requests received from its tenants with formulating a novel integre linear programming (ILP) problem. The main contributions of this paper are presented as follows:

• Our proposed joint resource allocation (JRA) method minimizes the power consumption cost of the cloud nodes alongside the bandwidth consumption cost of links. This way, our proposed method prefers less turned-on cloud nodes.

• Our system model is more realistic than the related works, because tenants' slice requests affect the CSP's network and normally this impact is neglected in the literature. This framework allows the tenants to orchestrate their own slice and serve their own customers[1].

• To solve the proposed optimization problem, we devise two different methods and analyze them from different aspects. Moreover, to overcome the infeasibility of these methods, which is because of the limited amount of physical resources, we propose a novel admission control mechanism. Our proposed joint admission control mechanism (AC-JRA) gains about 46% on average in comparison to the disjoint admission control mechanism (AC-DRA).

C. Organization

The rest of this paper is organized as follows. Section II introduces the system model and categorizes the constraints of the problem. In Section III, we first describe the formulation of the proposed joint optimization problem, and then linearize one of its constraints. Later on Section III, we propose an admission control mechanism. Moreover, a disjoint formulation of the problem is presented at the end of Section III. The simulation results are presented in Section IV. Finally, we conclude the paper in Section V.

II. SYSTEM MODEL

In this paper, we suppose a single CSP and multiple tenants, where the CSP serves the slice requests of these tenants. A typical illustration of the considered system model is depicted in Fig. 1. The details of our system model are as follows.

[1]Most tenants such as MVNOs or vertical industries (e.g., automotive and device manufacturing industry companies) know the requirements of their requested network and instantiate their own VNFs on their slice.

A. CSP's Network

We consider the CSP's network as a graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of cloud nodes (physical servers/forwarding devices)[2] and $\mathcal{L} = [l_{u,u'}]$ is the set of physical links, in which $l_{u,u'} = 1$, if cloud nodes u and u' are connected, otherwise is 0. We also assume that the CSP's cloud nodes are all NFV-enabled and the connectivity of all these cloud nodes is controlled by a logically-centralized SDN controller (Fig. 1).

We indicate the total resource capacity of cloud node $n \in \mathcal{N}$ as $\mathbf{r}_n = [r_n^{\text{Com}} \ r_n^{\text{Mem}} \ r_n^{\text{Sto}}]$, where r_n^{Com}, r_n^{Mem}, and r_n^{Sto} indicate the computational, memory, and storage capacities of n, respectively. Each physical link $l_{n,n'} \in \mathcal{L}$ has the limited bandwidth of $BW^{l_{n,n'}}$. Moreover, since the cloud nodes are distributed geographically, there is a considerable propagation delay between them denoted by $\tau^{l_{n,n'}}$.
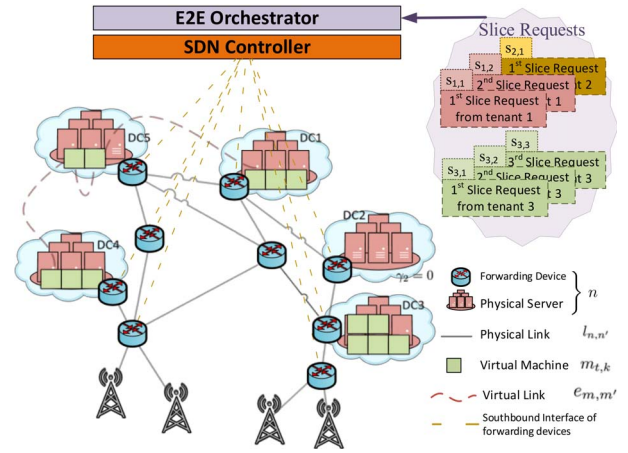

Fig. 1: Network architecture of the CSP in our system model.

B. Slice Requests

We consider $\mathcal{T} = \{1, \ldots, T\}$ as the set of all tenants and $\mathcal{K}_\text{t} = \{1, \ldots, K_t\}$ as the set of tenant t's slices in CSP's network. K_t is the number of slice requests of tenant t. Each tenant t requests its kth specific slice denoted by $s_{t,k} \in \mathcal{S}$ ($\mathcal{S}$ is the set all slice requests) with a specified set of virtual machines (VMs) and virtual links (VLs) indicated by $s_{t,k} = (\mathcal{M}_{t,k}, \mathcal{E}_{t,k})$ where $\mathcal{M}_{t,k}$ and $\mathcal{E}_{t,k}$ are the set of $s_{t,k}$'s requested VMs and VLs, respectively. We denote $\phi_{m_{t,k}} = [\phi_{m_{t,k}}^{\text{Com}} \ \phi_{m_{t,k}}^{\text{Mem}} \ \phi_{m_{t,k}}^{\text{Sto}}]$, $m_{t,k} \in \mathcal{M}_{t,k}$ as the vector of the requested computing, memory and storage resources for the mth VM in $s_{t,k}$.

Hereafter, we drop subscript t, k from m and m' for the sake of readability and convenience in the rest of this paper. We denote $\boldsymbol{\chi}^{e_{m,m'}} = [\varpi^{e_{m,m'}}, \tau_{\max}^{e_{m,m'}}], \forall m, m' \in \mathcal{M}_{t,k}$ as the vector of requested VLs between VM m and m', where $\varpi^{e_{m,m'}}$ is the requested data rate between m and m' in $s_{t,k}$ and $\tau_{\max}^{e_{m,m'}}$ is the maximum tolerable delay between its two connected VMs m and m'. After $s_{t,k}$ arrives at the CSP's E2E orchestrator, if there were enough resources left in the

[2]We use the terms node, physical node, data center (DC) and cloud node interchangeably.

network, it creates the slice templates and then calculates the cost based on the required resources. Then, the CSP creates $s_{t,k}$ by utilizing the specific cloud nodes and physical links that establish connections between those nodes[3]. This relationship between the CSP and its tenants is similar to the business model introduced by 5GPPP [17].

C. Allocating Resources of the Cloud Nodes to VMs

The CSP creates VM m by utilizing existing cloud node resources. We define the decision variable ξ_m^n, which is set to 1 if VM m is created on n by hypervisor for slice $s_{t,k}$, otherwise is 0.

To ensure that the sum of allocated VM resources in a cloud node do not exceed its resources, we have the following constraint:

$$\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}_t}\sum_{m\in\mathcal{M}_{t,k}} \xi_m^n \cdot \phi_m \preceq \mathbf{r}_n \qquad \forall n \in \mathcal{N}. \qquad (1)$$

D. Power Usage of the Cloud Nodes

We define the variable γ_n which is set to 1 if node n is on, otherwise is 0. To verify which nodes are turned on, we have the following constraint as [18]:

$$\xi_m^n \leq \gamma_n, \quad \forall n \in \mathcal{N}, \ \forall m \in \mathcal{M}_{t,k}, \forall t \in \mathcal{T}, \ \forall k \in \mathcal{K}_t. \quad (2)$$

We define the power consumed by node n inspired by [19]–[21]: $P_n = (P_n^{\max} - P_n^{\mathrm{idle}})U_n^{\mathrm{Com}} + \gamma_n P_n^{\mathrm{idle}}$,

where P_n^{idle} and $P_n^{\max}$ are the average power values when node n is idle and is fully utilized, respectively. Moreover, we define the CPU utilization of node n as follows [19], [20]:

$$U_n^{\mathrm{Com}} = \frac{\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}_t}\sum_{m\in\mathcal{M}_{t,k}} \xi_m^n \cdot \phi_m^{\mathrm{Com}}}{r_n^{\mathrm{Com}}}$$

E. Determining the Physical Path for Virtual Links

There may be multiple physical paths between two cloud nodes. Therefore, we should choose the best physical path for each virtual link. We denote $\mathcal{B}_{n,n'} = \{1, \cdots, b, \cdots, B_{n,n'}\}$ as the set of total possible physical paths, where $B_{n,n'}$ is the total number of physical paths between cloud nodes n and n'. We also denote the bth physical path between cloud nodes n and n' as $p_{n,n'}^b$. We define the decision variable $\pi_{p_{n,n'}^b}^{e_{m,m'}}, m \neq m'$ where it is set to 1 if VL $e_{m,m'}$ is sent over path $p_{n,n'}^b$, otherwise is set to 0. To determine whether physical link $l_{u,u'}$ contributes in physical path $p_{n,n'}^b$, the binary indicator $I_{p_{n,n'}^b}^{l_{u,u'}}$ is defined. Moreover, the set of all physical links that contribute in a path $p_{n,n'}^b$ is defined as $\mathcal{L}_{p_{n,n'}^b} = \left\{ l_{u,u'} \in \mathcal{L} \ \middle| \ I_{p_{n,n'}^b}^{l_{u,u'}} = 1 \right\}$.

To ensure that the path between nodes n and n' is established for the connectivity of VMs m and m', we introduce the following constraints:

$$\sum_{n\in\mathcal{N}}\sum_{n'\in\mathcal{N}}\sum_{b\in\mathcal{B}_{nn'}} \pi_{p_{n,n'}^b}^{e_{m,m'}} = 1, \qquad (3)$$

$$\pi_{p_{n,n'}^b}^{e_{m,m'}} = \xi_m^n \xi_{m'}^{n'}, \qquad (4)$$
$$\forall e_{m,m'} \in \mathcal{E}_{t,k}, \forall t \in \mathcal{T}, \ \forall k \in \mathcal{K}_t, n \neq n'.$$

Constraint (4) ensures that the right physical path is chosen between cloud nodes n and n', which are the hosts of VMs m and m'.

F. Bandwidth Limitations

The aggregated rates of all VLs belonging to all of the slices that pass a physical link should not exceed its bandwidth. This can be ensured if

$$\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}_t}\sum_{e_{m,m'}\in\mathcal{E}_{t,k}}\sum_{n\in\mathcal{N}}\sum_{n'\in\mathcal{N}}\sum_{b\in\mathcal{B}_{nn'}} I_{p_{n,n'}^b}^{l_{u,u'}} \pi_{p_{n,n'}^b}^{e_{m,m'}} \varpi^{e_{m,m'}} \leq$$
$$BW^{l_{u,u'}}, \quad \forall l_{u,u'} \in \mathcal{L}_{p_{n,n'}^b}, n \neq n' \quad (5)$$

holds. We calculate the overall traffic of all VLs belonging to all the slices that pass through these physical links and determine the cost of passing this traffic. The overall cost of bandwidth consumption is calculated as

$$\beta = \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}_t}\sum_{e_{m,m'}\in\mathcal{E}_{t,k}}\sum_{b\in\mathcal{B}_{nn'}}\sum_{l\in\mathcal{L}_{p_{n,n'}^b}} I_{p_{n,n'}^b}^{l_{u,u'}} \pi_{p_{n,n'}^b}^{e_{m,m'}} \psi^{l_{u,u'}} \varpi^{e_{m,m'}},$$

where $\psi^{l_{u,u'}}$ is the cost of transmitting 1bps traffic on $l_{u,u'}$.

Remark 1. *For the scenario in which m and m' are mapped to the same cloud node ($n = n'$), we assume that we have the physical connection between these VMs with much higher data rate and negligible delay.*

G. Maximum Tolerable Delay of Virtual Links

To ensure that we can guarantee tenant's maximum tolerable propagation delay ($\tau_{\max}^{e_{m,m'}}$) between its VLs, we have the following constraint:

$$\sum_{l_{u,u'}\in\mathcal{L}} I_{p_{n,n'}^b}^{l_{u,u'}} \pi_{p_{n,n'}^b}^{e_{m,m'}} \tau^{l_{u,u'}} \leq \tau_{\max}^{e_{m,m'}}, \forall e_{m,m'} \in \mathcal{E}_{t,k},$$
$$\forall t \in \mathcal{T}, \forall k \in \mathcal{K}_t. \quad (6)$$

It is noteworthy that because the CSP does not have information on the applications running on top of tenants' VMs and how much data will arrive at each VM, we cannot calculate the execution delay in this case. Also, since the amount of data passing the physical links is unknown and the CSP only guarantees the requested data rates of the tenants, the transmission delay (neither on the radio nor the network side) cannot be computed either. To make the paper more readable, Table I provides the main notation used in this paper.

III. THE PROPOSED RESOURCE ALLOCATION AND ADMISSION CONTROL OF SLICE REQUESTS

In this section, we first formulate the joint resource allocation problem (JRA) and then linearize one of its constraints. Then, in order to make sure the aforementioned problem would not be infeasible, we propose an admission control mechanism for the received slice requests which is solved before the proposed JRA. Moreover, we introduce disjoint VM allocation

TABLE I: Main Notation

Notation	Definition
$\mathcal{T}, \mathcal{K}_t$	Sets of tenants and tenant t's slices
$s_{t,k}$	The kth slice of tenant t
$\mathcal{M}_{t,k}, \mathcal{E}_{t,k}$	Sets of $s_{t,k}$'s requested VMs and VLs
$\phi_{m_{t,k}}$	The vector of requested virtual resource capacities for $m_{t,k}$
$\mathbf{r}_n$	The vector of physical resource capacities of cloud node n
$\boldsymbol{\chi}^{e_{m,m'}}$	The vector of requested VLs between $m_{t,k}$ and $m'_{t,k}$
$\varpi^{e_{m,m'}}$	Tenant t's requested data rate between $m_{t,k}$ and $m'_{t,k}$
$\tau^{e_{m,m'}}_{\max}$	Tenant t's requested maximum tolerable propagation delay between $m_{t,k}$ and $m'_{t,k}$
$\mathcal{N}, \mathcal{L}$	Sets of CSP's cloud nodes/physical servers and physical links
$l_{n,n'}$	Physical link between nodes n and n'
$BW^{l_{n,n'}}$	The bandwidth of the link $l_{n,n'}$
$\psi^{l_{u,u'}}$	The cost of transmitting 1 bps on the link $l_{u,u'}$
$I^{l_{u,u'}}_{p^b_{n,n'}}$	Indicator determining that physical link $l_{u,u'}$ contributes in the bth path between n and n'
γ_n	Decision variable is set to 1 if cloud node n is switched on
ξ^n_m	Decision variable is set to 1 if VM $m_{t,k}$ is embedded on cloud node n
$\pi^{e_{m,m'}}_{p^b_{n,n'}}$	Decision variable is set to 1 if VL $e_{m,m'}$ is mapped on the bth path between n and n'
Υ, ζ	Cost weight functions for balancing the power usage cost of all cloud nodes and the bandwidth consumption of VLs
C_{Total}	Total cost of CSP's network

(DMA) and disjoint VL allocation (DLA) problems, i.e., DRA with their corresponding admission control mechanisms to evaluate our JRA method at the end of this section.

A. The Proposed Joint Resource Allocation Problem

In this subsection, we formulate the proposed joint VM and VL placement. Aiming to minimize the total network cost for the CSP, we define the total cost function based on the minimization of active cloud nodes and reducing the cost of bandwidth consumption. The overall cost of resource allocation consists of two components: 1) bandwidth consumption cost, which relates to the power consumption of forwarding devices as well as the specific operational expenditure (OPEX) on diverse physical links, 2) the cost of consumed power of all turned-on cloud nodes. Therefore, the overall cost function can be stated as

$$C_{\text{Total}}(\boldsymbol{\pi}, \boldsymbol{\gamma}, \boldsymbol{\xi}) = \zeta\beta + \Upsilon \sum_{n \in \mathcal{N}} P_n, \qquad (7)$$

where $\boldsymbol{\pi}, \boldsymbol{\gamma}, \boldsymbol{\xi}$ are the sets of all $\pi^{e_{m,m'}}_{p^b_{n,n'}}$, γ_n, ξ^n_m, respectively. Moreover, ζ and Υ are scaling factors for translating bandwidth and power consumption into cost. Hence, the joint VM and VL placement optimization problem for the management of the requested slices by the CSP can be written as

$$
\begin{aligned}
\min_{\boldsymbol{\pi}, \boldsymbol{\gamma}, \boldsymbol{\xi}} \quad & C_{\text{Total}}(\boldsymbol{\pi}, \boldsymbol{\gamma}, \boldsymbol{\xi}) \\
\text{subject to} \quad & \text{C1:} \quad (1) \\
& \text{C2:} \quad \sum_{n \in \mathcal{N}} \xi^n_m = 1, \quad \forall t, k, m \\
& \text{C3:} \quad (2) \\
& \text{C4:} \quad (3) \\
& \text{C5:} \quad (4) \qquad\qquad\qquad (8) \\
& \text{C6:} \quad (5) \\
& \text{C7:} \quad (6) \\
& \text{C8:} \quad \gamma_n \in \{0,1\}, \quad \forall n \\
& \text{C9:} \quad \xi^n_m \in \{0,1\}, \quad \forall m, n \\
& \text{C10:} \quad \pi^{e_{m,m'}}_{p^b_{n,n'}} \in \{0,1\}, \quad \forall e, b, n.
\end{aligned}
$$

Constraint C1 makes sure that all VMs are hosted by their corresponding cloud nodes without violation in computational, memory, and storage capacities of the cloud nodes, respectively. Constraint C2 guarantees that each VM will be placed only in one cloud node. Constraint C3 ensures that we host VMs on turned-on cloud nodes. C4 and C5 make sure that each VL is only traversing one physical path and its path is between the cloud nodes hosting its corresponding VMs. Moreover, constraints C6 and C7 are related to bandwidth and delay limitations, respectively. Finally, constraints C8-C10 assure that decision variables of the problem are binary.

B. Solution to the JRA problem

The introduced joint problem (8) is an integer non-linear programming problem due to the constraint C5. Since there are effective ILP solvers i.e., MOSEK, we should linearize C5 first. Influenced by [22], we replace the $\xi^n_m \xi^{n'}_{m'}$ with an auxiliary variable $\theta^{n,n'}_{m,m'}$. Furthermore, to ensure that the equality $\theta^{n,n'}_{m,m'} = \xi^n_m \xi^{n'}_{m'}$ holds, we replace C5 in problem (8) with inequalities below:

$$
\begin{aligned}
\text{C5-a:} \quad & \pi^{e_{m,m'}}_{p^b_{n,n'}} = \theta^{n,n'}_{m,m'}, \\
\text{C5-b:} \quad & \theta^{n,n'}_{m,m'} \le \xi^n_m + 1 - \xi^{n'}_{m'}, \\
\text{C5-c:} \quad & \xi^n_m \le \theta^{n,n'}_{m,m'} + 1 - \xi^{n'}_{m'}, \\
\text{C5-d:} \quad & \theta^{n,n'}_{m,m'} \le \xi^{n'}_{m'}, \forall e_{m,m'} \in \mathcal{E}_{t,k}, \forall m, m' \in \mathcal{M}_{t,k}, \\
& \forall n, n' \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}_t, n \ne n'.
\end{aligned}
$$
$$(9)$$

By applying these constraints to (8) instead of C5, we make sure that both ξ^n_m and $\xi^{n'}_{m'}$ are not 0 when $\pi^{e_{m,m'}}_{p^b_{n,n'}}$ is set to 1.

C. Admission Control Mechanism for the Proposed Joint Resource Allocation

In this paper, we apply a novel admission control mechanism in order to determine which slice requests are making the joint cloud nodes and links infeasible. Since resources in the cloud nodes and the links between them are limited (constraints (1), (5), (6)), if the virtual resources demanded by the slice requests are more than the whole network's capacity, the problem would be infeasible and no slice request can be served by the network. Whenever the admission control problem is solved and the sum of all elastic variables (defined in Table II) of the problem is more than zero, we figure out that the original problem (i.e., (8)) would be infeasible. In each round of our heuristic method (i.e., Alg. 1), if the sum of elastic variables (i.e., the objective function of (10)) is not zero, we will find out which slice request is demanding more resources and reject it. We repeat this until the problem (8) becomes feasible and then we solve it using ILP solvers.

In order to find the slice requests which cause infeasibility, we use the elasticization approach in which, the corresponding constraints are elasticized by defining some elastic variables

TABLE II: Definition of the elastic variables in (10)

Variable(s)	Defined to elasticize	Const.
$\boldsymbol{\sigma}^{VM}$	Computing, Memory, and Storage capacity of the cloud nodes	(1)
σ^{BW}	Bandwidth of the physical links	(5)
σ^{τ}	Maximum tolerable latency of VLs	(6)

that extend the bounds on constraints [23], [24]. The problem of AC-JRA can be written as

$$\min_{\boldsymbol{\sigma}}(\boldsymbol{\sigma}^{VM} + \sigma^{BW} + \sigma^{\tau})$$

subject to

C1-a: $\sum_t \sum_k \sum_m \xi_m^n \cdot \phi_m \preceq \mathbf{r}_n + \boldsymbol{\sigma}_n^{VM}$,

C6-a: $\sum_t \sum_k \sum_{e_{m,m'}} \sum_n \sum_{n'} \sum_b I_{p_{n,n'}^b}^{l_{u,u'}} \pi_{p_{n,n'}^b}^{e_{m,m'}} \varpi^{e_{m,m'}}$
$$\leq BW^{l_{u,u'}} + \sigma_{l_{u,u'}}^{BW},$$

C7-a: $\sum_{l_{u,u'}} I_{p_{n,n'}^b}^{l_{u,u'}} \pi_{p_{n,n'}^b}^{e_{m,m'}} \tau^{l_{u,u'}} \leq \tau_{\max}^{e_{m,m'}} + \sigma_{s_{t,k}}^{\tau}$,

C2, C3, C4, C5-a, C5-b, C5-c, C5-d, C8, C9, C10.
(10)

In (10), the value of σ_n^{VM} represents the difference of the sum of allocated computing, memory, and storage resources from a feasible fulfilled set of these resources in node n. The value of $\sigma_{l_{u,u'}}^{BW}$ represents the deviation of the sum of all requested data rates of VLs from the actual bandwidth of the physical link $l_{u,u'}$. Also, the value of $\sigma_{s_{t,k}}^{\tau}$ shows the difference of one (or more) VL(s) of the slice request $s_{t,k}$ from the actual propagation delay(s) of physical links. Alg. 1 provides broader information about our admission control mechanism.

D. Summary of the Proposed Joint Method

Herein, we summarize our proposed joint method which consists of AC-JRA and JRA for our network slicing framework. The proposed method can be illustrated in Fig. 2.
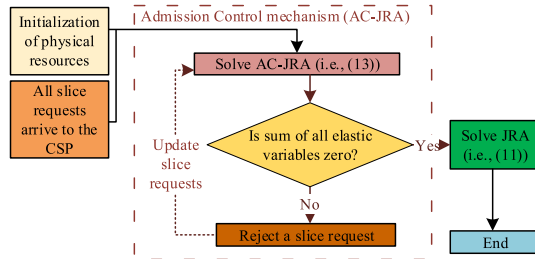

Fig. 2: Simplified flowchart of JRA.

E. Disjoint Resource Allocation of Cloud Nodes and Links

In order to evaluate the proposed JRA problem (Fig. 2), we advise a disjoint resource allocation (DRA) algorithm as a baseline of comparison. In this method, we divide both problems of admission control and resource allocation into nodes and links subproblems. The proposed admission control for DMA (AC-DMA) can be written as

$$\min_{\boldsymbol{\sigma}} \quad (\boldsymbol{\sigma}^{VM})$$
$$\text{subject to} \quad \text{C1-a, C2, C3, C8, C9.}$$
(11)

Afterwards, we solve the DMA subproblem which can be stated as

$$\min_{\boldsymbol{\gamma},\boldsymbol{\xi}} \quad \Upsilon(\sum_{n\in\mathcal{N}} P_n)$$
$$\text{subject to} \quad \text{C1, C2, C3, C8, C9.}$$
(12)

Algorithm 1: Joint admission control (i.e., AC-JRA) of the received slice requests (Solving (10))

Input: $\mathcal{N}, \mathcal{L}, \mathbf{r}, \mathbf{BW}, \boldsymbol{\psi}, \boldsymbol{\tau}, \mathbf{I},$
$\quad\quad \mathcal{S}, \mathcal{M}, \mathcal{E}, \boldsymbol{\phi}, \boldsymbol{\varpi}, \boldsymbol{\tau}_{\mathbf{max}}$

1 **repeat**
2 $\quad$ Solve (10) according to $\mathcal{S}$
3 $\quad$ **if** $\sum \boldsymbol{\sigma}^{VM} \neq 0$ **then**
4 $\quad\quad$ **if** $\sum_{n\in\mathcal{N}} \sigma_n^{VM,Com} \neq 0$ **then**
5 $\quad\quad\quad$ **for** $t \leq T$ and $k \leq K_t$ **do**
6 $\quad\quad\quad\quad$ $SumUsedCom(s_{t,k}) = \sum_{m\in\mathcal{M}_{t,k}} \phi_{m_{t,k}}^{\text{Com}}$
7 $\quad\quad\quad$ $s_{t,k}^{\star} = \arg\max_{s_{t,k}}(SumUsedCom(s_{t,k}))$
8 $\quad\quad$ **else if** $\sum_{n\in\mathcal{N}} \sigma_n^{VM,Mem} \neq 0$ **then**
9 $\quad\quad\quad$ **for** $t \leq T$ and $k \leq K_t$ **do**
10 $\quad\quad\quad\quad$ $SumUsedMem(s_{t,k}) = \sum_{m\in\mathcal{M}_{t,k}} \phi_{m_{t,k}}^{\text{Mem}}$
11 $\quad\quad\quad$ $s_{t,k}^{\star} = \arg\max_{s_{t,k}}(SumUsedMem(s_{t,k}))$
12 $\quad\quad$ **else if** $\sum_{n\in\mathcal{N}} \sigma_n^{VM,Sto} \neq 0$ **then**
13 $\quad\quad\quad$ **for** $t \leq T$ and $k \leq K_t$ **do**
14 $\quad\quad\quad\quad$ $SumUsedSto(s_{t,k}) = \sum_{m\in\mathcal{M}_{t,k}} \phi_{m_{t,k}}^{\text{Sto}}$
15 $\quad\quad\quad$ $s_{t,k}^{\star} = \arg\max_{s_{t,k}}(SumUsedSto(s_{t,k}))$
16 $\quad\quad$ set all requested resources of $s_{t,k}^{\star}$ to zero and reject this slice request
17 $\quad$ **else**
18 $\quad\quad$ **if** $\sum_{l_{u,u'}\in\mathcal{L}} \sigma_{l_{u,u'}}^{BW} \neq 0$ **then**
19 $\quad\quad\quad$ **for** $t \leq T$ and $k \leq K_t$ **do**
20 $\quad\quad\quad\quad$ $SumUsedRate(s_{t,k}) = \sum_{e_{m,m'}\in\mathcal{E}_{t,k}} \varpi^{e_{m,m'}}$
21 $\quad\quad\quad$ $s_{t,k}^{\star} = \arg\max_{s_{t,k}}(SumUsedRate(s_{t,k}))$
22 $\quad\quad$ **else if** $\sum_{l_{u,u'}\in\mathcal{L}} \sigma_{s_{t,k}}^{\tau} \neq 0$ **then**
23 $\quad\quad\quad$ $s_{t,k}^{\star} = \arg\max_{s_{t,k}} \sigma_{s_{t,k}}^{\tau}$
24 $\quad\quad$ set all requested resources of $s_{t,k}^{\star}$ to zero and reject this slice request
25 $\quad$ Update $\mathcal{S}$
26 **until** $\sum(\boldsymbol{\sigma}^{VM} + \sigma^{BW} + \sigma^{\tau}) = 0$
Output: $\mathcal{S}$

After solving DMA subproblem, we solve the proposed admission control for DLA (AC-DLA) knowing $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ from (12):

$$\min_{\boldsymbol{\sigma}} \quad (\sigma^{BW} + \sigma^{\tau})$$
$$\text{subject to} \quad \text{C4, C5, C6-a, C7-a, C10.}$$
(13)

After the proposed AC-DLA, the last phase is to allocate link-related resources (also known as DLA subproblem) (e.g., BW and τ) which can be stated as:

$$\min_{\boldsymbol{\pi}} \quad C_{\text{Total}}(\boldsymbol{\pi}, \boldsymbol{\gamma}, \boldsymbol{\xi})$$
$$\text{subject to} \quad \text{C4, C5, C6, C7, C10.}$$
(14)

The admission control mechanisms used in (11) (i.e., AC-DMA) and (13) (i.e., AC-DLA) are alike Alg. 1.

IV. EXPERIMENTAL ANALYSIS

In this section, we evaluate the performance of the proposed joint (i.e., JRA) and disjoint (i.e., DMA and DLA) methods alongside their admission control mechanisms (i.e., AC-JRA vs. AC-DMA and AC-DLA).

A. Simulation Environment

Herein, we explain the simulation environment used for the results in Section VI. Using a random topology with 4 cloud nodes, we initialize the CSP's physical network. The values of the parameters related to the cloud nodes and requested VM resources (e.g., computational, memory, and storage resources) are presented at the bottom of Table III. Moreover, the top of

TABLE III: Simulation Parameters

CSP's Resource	In/Out	min	max
$BW^{l_{n,n'}}$	Intra	10^7	10^7
(Kbps)	Inter	9×10^4	1.9×10^5
ψ	Intra	1	1
($)	Inter	$10^{-3}BW$	$10^{-2}BW$
$\tau^{l_{n,n'}}$	Intra	0	0
(ms)	Inter	10^{-1}	4
Requested Resource			
$\varpi^{e_{m,m'}}$ (Kbps)	Between VMs	10^4	1.1×10^5
$\tau_{\max}^{e_{m,m'}}$ (ms)	Between VMs	5	14
Other Parameters			
$|T|, |K|, |N|$, and $|M|$	1-16, 1, 4, and 3		
r_n^{Com}, r_n^{Mem}, and r_n^{Sto}	7000 MHz, 800 GBs, and 2000 GBs		
$\phi_m^{Com}, \phi_m^{Mem}$, and ϕ_m^{Sto}	1000 MHz, 64 GBs, and 120 GBs		
Υ and ζ	1 and 9×10^{-5}		

Table III displays the values of the parameters related to the CSP's physical links (e.g., propagation delay and bandwidth of links). Moreover, the weights of node and link parts of the objective function (7) are set to $\Upsilon = 1$ and $\zeta = 9 \times 10^{-5}$ experimentally in order to balance the effects of these parts to each other.

For evaluating the effect of the number of slice requests on CSP's network, we increase the number of tenants from 1 to 16, keeping $k_t = 1$ in all circumstances (all tenants have only one slice request). We also keep the number of VMs in all slice requests the same ($|M| = 3$).

The values of the parameters related to the VL requests of each slice request is shown in the middle rows of Table III. Note that the parameters that do not have the same max and min value presented in Table III are set randomly between the aforementioned values.

It is worth mentioning that we use MOSEK solver in order to solve problems (8), (10), (11), (12), (13), and (14) [25], [26]. Moreover, all simulation steps (including initialization, admission control mechanisms, and solving ILP problems with MOSEK toolbox) have been implemented in MATLAB software [27] which is widely used to solve resource allocation problems [28], [29].

B. Simulation Results

The overall cost (according to (7)) of the CSP for serving the slice requests is shown in Fig. 3(a). The JRA method (i.e., Fig. 2) is winning the race because the cost is growing smoothly using this method.

To analyze the performance of the admission control mechanisms (problem (10) (i.e., AC-JRA) vs. (11) (i.e., AC-DMA) and (13) (i.e., AC-DLA)), Fig. 3(b) displays the acceptance ratio of the two methods. Hereby, the average difference between the acceptance ratio of JRA vs. DRA is 0.46, indicating the advantage of the JRA method. It is worth noting that the DRA (i.e., Subsection III-E) method, which is the intersection of AC-DMA and AC-DLA, cannot accept any slice request after the number of them reaches to 7. This is the reason that the overall cost in Fig. 3(a) decreases dramatically, only showing the cost of subproblem (12). Moreover, Fig. 3(c) shows the number of the rejected slice requests in the two aforementioned methods.
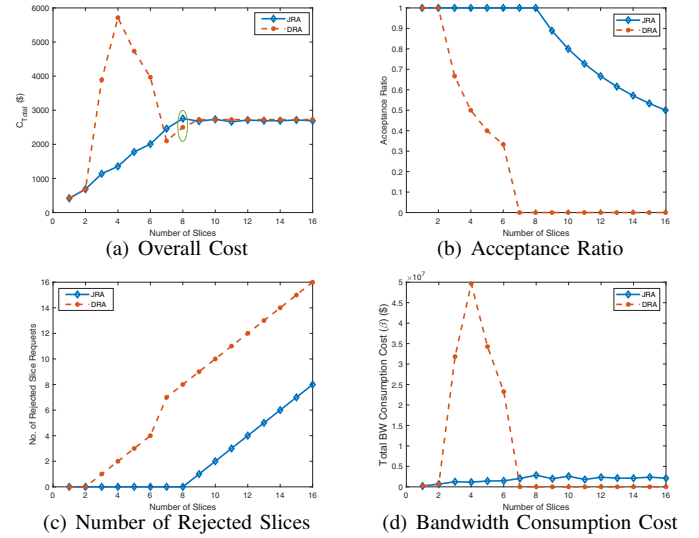

(a) Overall Cost

(b) Acceptance Ratio

(c) Number of Rejected Slices

(d) Bandwidth Consumption Cost

Fig. 3: Comparison of the (a) overall cost, (b) acceptance ratio, (c) number of rejected slices, and (d) bandwidth consumption cost (β) between the proposed methods.

Fig. 3(d) shows the superiority of the JRA method against the DRA method in terms of the bandwidth consumption cost. The 0 value in the DRA method indicates that after the CSP gets 7 slice requests, the AC-DLA subproblem (i.e., (13)) is getting infeasible anyway and there will be no bandwidth consumption cost in (14). According to Fig. 3(b), after getting 7 slice requests, computing resources are fulfilled and the CSP cannot accept any more slice requests, but in the DRA method, although after having 7 slice requests, some requests will be accepted in AC-DMA subproblem (i.e., (11)), the acceptance ratio of the AC-DLA subproblem (i.e., (13)) suddenly drops to 0%. The main reason of this drop-off is that the links and nodes resources are related to each other and when we solve them in different phases, after getting some requests, subproblem (13) will always be infeasible, thus leading to the cost of 0 according to Fig. 3(d).

V. CONCLUSION

In this paper, we considered a network slicing problem where the CSP should allocate its resources according to the slice requests of its different tenants. An SDN/NFV-enabled network with some cloud nodes was considered and allocating the computing and bandwidth resources to different slices of CSP's tenants was the main problem. Because of the limited amount of resources, we performed an admission control mechanism before solving the joint problem to reject the slice requests which make the joint problem infeasible. To evaluate the joint method (i.e., Fig. 2), we considered a new disjoint formulation (i.e., DRA method) of the original problem and the simulation results were all voted for our joint formulation. Incorporating the arrival time of each slice request into JRA and DRA can be regarded as a future line of research. Moreover, applying this system model in a discrete event simulator (e.g. NS3) will be done in our future work.

REFERENCES

[1] M. Series, "IMT vision–framework and overall objectives of the future development of IMT for 2020 and beyond," *ITU Recommendation*, pp. 2083–0, 2015.

[2] NGMN Alliance, "5G white paper," 2015.

[3] NGMN Alliance, "Description of network slicing concept," 2016.

[4] P. Rost, A. Banchs, I. Berberana, M. Breitbach, M. Doll, H. Droste, C. Mannweiler, M. A. Puente, K. Samdanis, and B. Sayadi, "Mobile network architecture evolution toward 5G," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 84–91, 2016.

[5] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[6] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[7] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[8] 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 Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

[9] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," *IEEE Network (Early Access)*, pp. 1–8, 2019.

[10] V. Sciancalepore, F. Cirillo, and X. Costa-Perez, "Slice as a service (SlaaS) optimal IoT slice resources orchestration," in *Proc. of IEEE Global Communications Conference (GLOBECOMM 2017)*, pp. 1–7, IEEE, 2017.

[11] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in *Proc. of IEEE International Conference on Computer Communications (INFOCOM 2017)*, pp. 1–9, IEEE, 2017.

[12] B. Han, J. Lianghai, and H. D. Schotten, "Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5G networks," *IEEE Access*, vol. 6, pp. 33137–33147, 2018.

[13] T. LeAnh, N. H. Tran, D. T. Ngo, and C. S. Hong, "Resource allocation for virtualized wireless networks with backhaul constraints," *IEEE Communications Letters*, vol. 21, no. 1, pp. 148–151, 2016.

[14] A. Fendt, S. Lohmuller, L. C. Schmelz, and B. Bauer, "A network slice resource allocation and optimization model for end-to-end mobile networks," in *Proc. of IEEE 5G World Forum (5GWF'18)*, pp. 262–267, IEEE, 2018.

[15] 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. of IEEE Global Communications Conference (GLOBECOMM 2017)*, pp. 1–6, IEEE, 2017.

[16] P. Caballero, A. Banchs, G. De Veciana, X. Costa-Pérez, and A. Azcorra, "Network slicing for guaranteed rate services: Admission control and resource allocation games," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6419–6432, 2018.

[17] 5GPPP Architecture Working Group, "View on 5G architecture, Version 2.0," *White Paper*, 2017.

[18] A. Zakeri, N. Gholipoor, M. R. Javan, and N. Mokari, "Energy cost minimization by joint radio and NFV resource allocation: E2E QoS framework," *arXiv preprint arXiv:1907.06212*, 2019.

[19] Y. Gao, H. Guan, Z. Qi, B. Wang, and L. Liu, "Quality of service aware power management for virtualized data centers," *Journal of Systems Architecture*, vol. 59, no. 4-5, pp. 245–259, 2013.

[20] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[21] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[22] R. A. Addad, T. Taleb, M. Bagaa, D. L. C. Dutra, and H. Flinck, "Towards modeling cross-domain network slices for 5G," in *Proc. of IEEE Global Communications Conference (GLOBECOM 2018)*, pp. 1–7, IEEE, 2018.

[23] J. W. Chinneck, *Feasibility and Infeasibility in Optimization*. Algorithms and Computational Methods, Springer, 2008.

[24] M. Tajallifar, S. Ebrahimi, M. R. Javan, and N. Mokari, "QoS-aware joint power allocation and task offloading in an MEC/NFV-enabled C-RAN network," *arXiv preprint arXiv:1912.00187*, 2019.

[25] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.

[26] CVX Research, Inc., "CVX: Matlab software for disciplined convex programming, Version 2.0," 2012.

[27] MathWorks, *MATLAB*, 2019.

[28] D. Sattar and A. Matrawy, "Optimal slice allocation in 5G core networks," *IEEE Networking Letters*, vol. 1, no. 2, pp. 48–51, 2019.

[29] 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 Transactions on Network and Service Management*, vol. 16, no. 1, pp. 374–388, 2019.