Distributed Resource Allocation Optimization in 5G Virtualized Networks

Hassan Halabian

***Abstract*— The concepts of network function virtualization and end-to-end network slicing are the two promising technologies empowering 5G networks for efficient and dynamic network/service deployment and management. In this paper, we propose a resource allocation model for 5G virtualized networks in a heterogeneous cloud infrastructure. In our model, each network slice has a resource demand vector for each of its virtual network functions. We first consider a system of collaborative slices and formulate the resource allocation as a convex optimization problem, maximizing the overall system utility function. We further introduce a distributed solution for the resource allocation problem by forming a resource auction between the slices and the data centers. By using an example, we show how the selfish behavior of non-collaborative slices affects the fairness performance of the system. For a system with non-collaborative slices, we formulate a new resource allocation problem based on the notion of dominant resource fairness and propose a fully distributed scheme for solving the problem. Simulation results are provided to show the validity of the results, evaluate the convergence of the distributed solutions, show protection of collaborative slices against non-collaborative slices and compare the performance of the optimal schemes with the heuristic ones.**

***Index Terms*— 5G network function virtualization, network slicing, optimal resource allocation, algorithmic games.**

I. Introduction

NETWORK Function Virtualization (NFV) and Software-Defined Networking (SDN) are two promissing techniques used in 5G network architecture evolution to provide significant capital and operational expenditure saving by immigrating the network functions and services to cloud infrastructures [1]–[4]. NFV provides software and hardware decoupling by virtualizing the service components and network functions and running them on top of a virtualization system, i.e., virtual machines or containers [5]. On the other hand, SDN provides centralized control plane for control and management of network services and network functions [6]. These two techniques together with the concept of end-to-end (E2E) network slicing [7]–[10] enable mobile network providers to create virtualized E2E networks over cloud systems. Depending on the functional, operational and performance requirements, there have been defined a number of 5G network slices in accordance with the concept of networks as a service (NaaS), including but not limited to enhanced mobile broadband (eMBB), ultra-reliable and low-latency communication (uRLLC) and massive machine-type communications (mMTC) [11].

A virtualized network slice consists of a number of Virtual Network Functions (VNFs) distributed geographically in numerous Data Centers (DCs). Each VNF provides certain services in its slice and all the VNFs of a slice collectively provide wireless network access to the User Equipments (UEs) attached to that slice. Fig. 1 shows an illustration of network function virtulization architecture for 5G networks which provides RAN (Radio Access Network) and mobile back-haul/core function virtualization in data centers. As shown in Fig. 1, in 5G C-RAN (Cloud-Radio Access Network) architecture, communication signals are collected from the cell towers by the Remote Radio Heads (RRH) and after RF (Radio Frequency) processing they are sent to the Base Band Units (BBUs) for digital processing. BBU may itself split into Central Unit (CU) and Distributed Unit (DU) [12]. DU runs latency sensitive RAN functions while CU is supposed to run latency tolerant functions. In C-RAN architecture, some or all BBU RAN functions may be virtualized [13]. Packet level processing is done in SGW (Serving Gateway) and PGW (Packet Gateway) and mobility services are provided by MME (Mobility Management Entity). Subscriber related information processing, e.g., authentication, location, etc., is done by HSS (Home Subscriber Server) [14]. In 4G technology, these functions are implemented in dedicated hardware while in a virtulaized architecture they are placed as virtual machines/containers in DCs promoting the concept of Mobile Carrier Cloud [4]. For 5G networks, 3GPP defines a service-based network architecture in which mobile back-haul/core services are provided by virtualized network functions including (but not limited to) Access and Mobility Management Function (AMF), Session Management Function (SMF), Authentication Server Function (AUSF), User Plane Function (UPF), Unified Data Management (UDM) and Policy Control Function (PCF) [15].

Due to interdependence of VNFs in each slice, there are placement constraints for placing the VNFs in the serving DCs. For instance, due to front-haul latency and jitter constraints, there is distance limitation between the radio processing functions and the BBU functions. Furthermore, certain slices, e.g., uRLLC, require to satisfy certain QoS (Quality of Service) requirements and therefore there exist some placement constraints for the VNFs of those slices. There might also exist some placement restrictions due to administration, logistics and management concerns. In the model presented in this paper, it is assumed that the VNFs are placed in predetermined DCs called Access Data Center (ADC), Metro Data Center (MDC) and Core Data Center (CDC).

Manuscript received October 10, 2018; revised January 7, 2019; accepted January 11, 2019. Date of publication February 5, 2019; date of current version February 14, 2019.

The author is with the Huawei Canada Research Centre, Ottawa, ON K2K 3J1, Canada (e-mail: hassan.halabian@huawei.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2019.2894305

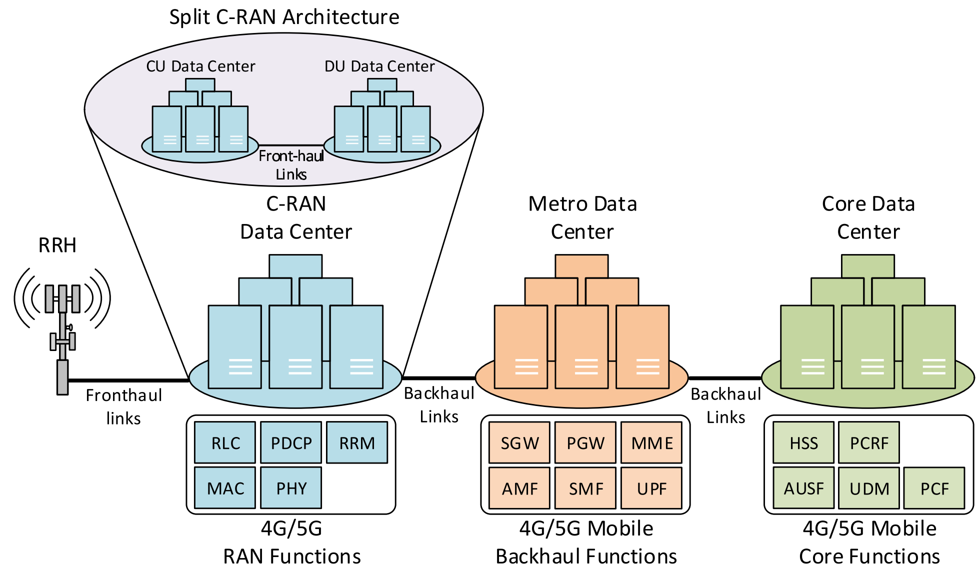

Fig. 1. 5G network fuction virtualization - C-RAN and mobile backhaul architecture.

ADC hosts VNFs processing L1/L2 access functions, e.g., RLC (Radio Link Control), PDCP (Packet Data Convergence Protocol), RRM (Radio Resource Management), MAC (Medium Access Control) and PHY (Physical). ADCs are preferably located physically close to the cell sites and RRHs due to latency considerations. MDC hosts VNFs processing traffic forwarding, classifications, admission and mobility management, etc. Examples of functions in MDC are 4G functions MME, SGW, PGW and 5G functions AMF, SMF and UPF. CDC may locate functions dealing with subscriber related information and policy enforcement and charging functions. HSS is an example of a 4G function in CDC. In 5G systems, CDC may locate functions such as UDM and AUSF. Fig. 2 shows a topology illustration of the distribution of ADC, MDC and CDC data centers in a 5G network architecture. Similar modeling had been captured for 5G packet core in the literature, e.g., in [9].

The VNFs of a single slice have heterogeneous resource requirements, i.e., CPU, memory, bandwidth and storage. For example, BBU functions are CPU intensive as they execute heavy processing DSP functions while PGW is a bandwidth intensive function as it passes the entire slice traffic. The resource requirements of slices of the same type are also different since they are serving different number of UEs. For instance a provider might run multiple Internet of Things (IoT) slices each one dedicated for a specific application. These slices might have different resource demands depending on the number of attached UEs to them and also the type of IoT application. Furthermore, the serving DCs have heterogeneous resource capacities for each of their resources. For such a system, with heterogeneous resource capacities and heterogeneous slice requirements, optimal resource allocation to the network slices is a challenging problem. Each network slice might have a different utility/revenue function not willing to share with DCs. Moreover, the DCs might not be under the same management. For such a model, a distributed resource allocation scheme is more preferable for both the slice providers and also the DCs.

In contrast with centralized solutions which are commonly based on SDN, distributed schemes are implemented by distributing the actors of the algorithms among the data centers and slice providers and also define the actions and operations such that the result of the actions of all the entities provide optimal resource allocation to the slice VNFs. This is a different approach than SDN in managing and optimizing the resource allocation in 5G VNFs which is believed to be a better approach due to the following issues with centralized solutions:

- One of the main issues of centralized solutions is lack of scalability. With the growth of the number of network slices and their VNFs and dynamic network changes, scalability becomes an important challenge for centralized solutions.
- Single-point-of-failure is another problem with centralized solutions. Any failure in the central optimizer may result in the entire resource allocation scheme to fail.
- Another drawback of centralized approaches is that the slice providers need to disclose their (possibly private) utility functions with DCs.
- Finally, centralized solutions fail to provide a global optimal resource allocation if the DCs are not under the same management and do not want to disclose their resource capacities to a third party optimizer.

On the other hand, distributed systems have a storied history on the internet. Interior gateway protocols such as OSPF [16] and IS-IS [17] are fully distributed without any central control. Distributed file sharing applications such as BitTorrent [18] have made a substantial impact in content distribution. Session initiation protocol [19] is a distributed implementation of telephony and BlockChain [20] implements a distributed trust framework. All of these technologies have the fundamental aim of improving their core performance by removing centralized entities. Moreover, distributed bandwidth allocation in networking has already appeared in the literature for both single link bandwidth allocation [21] and network wide bandwidth allocation [22], [23].

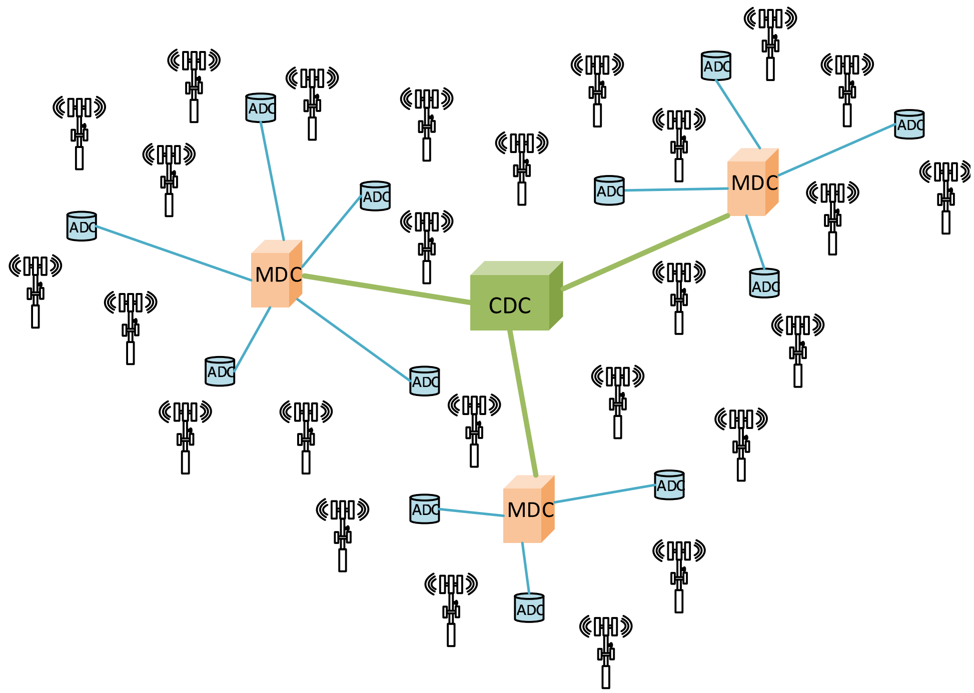

Fig. 2. A topology of ADC, MDC and CDC distribution.

Our contributions in this paper are summarized as follows: We first propose a resource allocation model for 5G virtualized networks in a heterogeneous cloud infrastructure with E2E network slices having diverse requirements and resource demands. The heterogeneity of the slice requirements is reflected in our model by considering different resource demand vectors for each function of each slice. The demand vectors for each slice specify the amount of resources required for each function to complete a network service unit in one unit of time, e.g., to serve one (or more) wireless user equipment(s). Hence, the resource volume of each slice can be specified by a scalar multiplier of its demand vectors and is called *slice thickness* in this paper. The resource allocation optimization is maximizing the total network utility as a function of the slice thicknesses with the constraints of the DCs' resource capacities. The utility functions are assumed to be strictly concave and thus the resource allocation is a convex optimization problem. Specific choices of utility functions may provide desired fairness properties, e.g., max-min, α-proportional, etc. We then introduce a distributed solution for solving the resource allocation problem by forming a resource allocation auction between the slices and the DCs. It is proven that the resource allocation game has a Nash equilibrium and also the Nash solution is the same as the solution of the centralized system optimization problem, i.e., in equilibrium the slice thicknesses are also maximizing the overall system utility function. By using an example, we show how selfish behavior of a non-collaborative slice can affect the fairness performance of the proposed solution. To fix this problem, we propose a new resource allocation formulation based in the idea of Dominant Resource Fairness (DRF) [24]. We extend the notion of DRF to be applicable in our system model and formulate a centralized resource optimization problem. Furthermore, we propose a novel fully distributed solution for solving the proposed resource allocation problem. Finally, we use simulation and numerical analysis to support the validity of the results, show the convergence of the proposed solutions, show protection of collaborative slices against non-collaborative slices and compare the performance of the optimal solutions with heuristic ones.

The rest of the paper is organized as follows. Section II presents the related research work in this domain of research. In Section III, the proposed 5G resource allocation model is introduced. Section IV formulates the global centralized system utility optimization problem for a system with collaborative slices and presents a distributed auction-based solution for which the existence of Nash equilibrium is proven. Section V provides a resource allocation problem formulation for a system with non-collaborative slices that lie about their demand vectors and presents a novel distributed scheme based on the notion of DRF that protects the collaborative slices from the non-collaborative ones. In Section VI, we present a system-level implementation of the proposed algorithms in 5G virtualized networks. Section VII, presents the simulation and numerical results. Finally, Section VIII provides the conclusions of the paper.

II. Related Work

The concept of network slicing provides flexible and dynamic provisioning of network services to vertical industries including but not limited to manufacturing, health care, media and entertainment, automotive, public safety, financial services, etc. [25]. The research in this area resulted in forming many joint projects over open source platforms such as OPNFV [26] and OpenMANO [27] for management and orchestration of wireless network functions [1], [2].

Many research activities in this area are mainly focused on radio resource virtualization [7], [8], management and orchestration of network functions [9], [10] without considering the heterogeneous demand, QoS and performance requirements of slices. The work in [28] focuses on Evolved Packet Core (EPC) virtualization and addresses the optimal placement of SGW and PGW functions in cloud carrier networks without considering the end-to-end network slice requirements.

While most of the work in VNF orchestration is dedicated to per DC orchestration of the VNFs, service-centric slice orchestration (with diverse E2E requirements) is studied far less in [29]. In [12] and [29]–[31], there have been proposed algorithms for optimal VNF resource allocation problem. Wang *et al.* [31] formulate a mixed-integer linear programming (MILP) for joint function chaining and resource allocation problem and to solve this problem they propose heuristic alternatives. Similarly Liu *et al.* [30] formulate the function chaining problem as a binary NP-hard programming problem and to solve it the authors propose heuristic approaches. In [29], complex network theory is used to obtain topological information of slices and infrastructure network and ranking the nodes for mapping VNFs to the nodes. In none of these papers, the model is comprehensive in a sense to consider the DC models and the available resources in DCs (computation, memory and bandwidth) in the problem formulation. Moreover, the objective in all of them misses the slice provider utilities, fairness and also heterogeneous resource demands of the network slices. Clayman *et al.* [32] support the idea of high-level system orchestration for dynamic management of VNFs and propose an architectural system model without proposing a technical function placement and resource allocation solution. The work in [12] formulates a MILP to derive the optimal number of VNFs to meet the performance requirements of a network slice. The authors further form a coalition game between DCs to host the slice VNFs. In [33], an algorithm is proposed that derives the optimal number and locations of vEPC's virtual instances over the federated cloud based on coalition formation game, wherein the aim is to build optimal coalitions of cloud networks to host the virtual instances of the vEPC elements. In contrast to the aforementioned references, we consider the resource allocation among a number of competing slices with diverse resource demands on a set of heterogeneous DCs. Moreover, we formulate the resource allocation with the objective of maximizing the overall system utility. Our distributed scheme forms an auction game between the slices and the DCs (in contrast to [12] where the game is between the DCs) to solve the system optimization problem.

Addad *et al.* [34] formulate a MILP optimization model that enables a cost-optimal deployment of network slices, allowing a mobile network operator to efficiently allocate the underlying layer resources according to the users' requirements. For each network slice, the proposed solution guarantees the required delay and the bandwidth, while efficiently handling the usage of underlying nodes, which leads to reduced cost. The work in [35] and [36] focuses on the problem of optimal VNF placement on federated cloud following the concept of carrier cloud. Ksentini *et al.* [35] focus on the problem of the SGW-C placement, where a trade-off is needed between reducing the SGW relocation frequency and balancing the traffic load among the underlying SGW-C VNFs. Nash Bargaining game is used to derive a fair solution. The work in [36] introduces different VNF placement algorithms for carrier cloud considering the mobility features and service usage behavioral patterns of mobile users, in addition to the mobile operators' cost. Laghrissi *et al.* [5] propose a solution that takes into account the geographic distribution of variant Edge Clouds (ECs), as well as the spatio-temporal distribution of users' requests to place VMs in optimal positions for serving the variant users while QoE is ensured and the cost is minimized. This is achieved by mapping the non-uniform signaling messages to a new uniform environment, namely, the canonical domain, whereby the placement of core functions is more feasible and efficient. Laghrissi *et al.* [37] present a tool for developing a spatio-temporal model of mobile service usage over a particular geographical area which helps to define the behavior of mobile users in terms of mobility patterns and mobile service consumption. Based on this tool, they present a benchmark of some VNF placement algorithms. In our work, in contrast to the research work described above, placement of the VNFs of all the slices are predetermined by the slice operator and the objective is to achieve the optimal resource allocation in the hosting data centers such that the overall slice utility function is maximized.

III. System Model

Consider a virtualized 5G system consisting of a set of N network slices. Each slice n is composed of a number of VNFs denoted by $\mathcal{F}^n = \{f_1^n, f_2^n, \cdots, f_M^n\}$ where M is the number of VNFs for each slice. If different network slices have different number of VNFs, we let M be the maximum. Fig. 3 shows an illustration of the system model for three slices.

In our model, there are K DCs over which the VNFs are being distributed. The available resources in each DC k are denoted by vector $\mathbf{R}_k = (R_{k,1}, R_{k,2}, \cdots, R_{k,L})$ where $R_{k,\ell}$ represents the amount of available resource ℓ on DC k. The available resources in each DC are for example CPU, memory, bandwidth and storage. Each VNF f_m^n of slice n is placed in one of the DCs. The placement of each VNF in DCs is predetermined by the operator. However, the amount of resources allocated to each VNF in each DC is unknown. The goal of the resource allocation is to determine the amount of resources allocated to each of the VNFs in each DC.

We define the notion of demand vector for the VNFs of each network slice as follows: Each VNF is associated with a demand vector which specifies an estimated amount of resources required to provide a unit of wireless service in unit of time. A unit of service might be defined for example to serve n ($n \geq 1$) attached UEs. For example a demand vector for a gNodeB VNF might be defined as $(4 \text{ CPU core}, 1\text{GB RAM}, 1\text{GB storage}, 100 \text{ Mbps BW})$. This means that for one unit of wireless service, gNodeB VNF requires 4 CPU cores, 1GB RAM, 1GB storage and

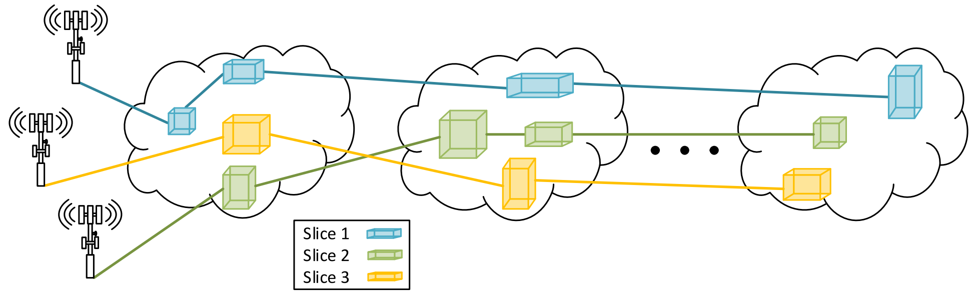

Fig. 3. System model for function placement and resource allocation.

100Mbps bandwidth. What is important here is the ratio of the elements in the demand vectors. More specifically, it says for every 4 CPU cores it needs at least 1GB RAM, 1GB storage and 100Mbps bandwidth. Less values for the RAM, storage and bandwidth will be useless. In fact, demand vectors determine the shape of the resources required by the VNFs. Another view of a demand vector is to consider it as a conceptual virtual machine with the resources specified by the elements of the demand vector as the granularity of the resource allocation to that VNF. This means that resource allocation to the VNFs is determined by scalar multiplications of VNFs' demand vectors. The demand vectors are dependent on the software implementation of the VNFs and are determined by the slice providers. These vectors are heterogeneous among the VNFs of a slice and also across the VNFs of different slices. For instance the vPGW is a very bandwidth intensive function since all wireless traffic passes through this function while the gNodeB is a very CPU intensive function as it executes very fast DSP tasks.

We assume that each network slice n is associated with a set of demand vectors for each of its VNFs denoted by $\mathbf{d}^k_{f^n_m} = (d^k_{f^n_m,1}, d^k_{f^n_m,2}, \cdots, d^k_{f^n_m,L})$ for each DC k and each VNF m of slice n. If VNF m for slice n (i.e., f^n_m) is not defined or is not going to be placed on DC k, we set $\mathbf{d}^k_{f^n_m}$ to zero vector. The set of demand vectors of each network slice is denoted by $\mathcal{D}_n = \{\mathbf{d}^k_{f^n_m} | m = 1, 2, \cdots M, k = 1, 2, \cdots, K\}$ which reflects the amount of resources for each VNF in each DC to complete a network service unit in one unit of time, e.g., to serve one (or n) wireless user equipment(s). The demand vectors are heterogeneous across the network slices and also inside each network slice. We now define the slice thickness variable v_n ($v_n > 0$) for each slice n which denotes the number of network service unis (wireless user services) that can be executed in one unit of time. We can interpret $\mathcal{D}_n$ as the slice realization for a single service and therefore $v_n \mathcal{D}_n$ represents the amount of resources allocated to slice n to support v_n service(s) in one unit of time. In other words, v_n specifies how the network slice expands or shrinks with respect to its demand vectors $\mathcal{D}_n$. By $\mathbf{v} = (v_1, v_2, \cdots, v_N)$ we denote the vector of the slice thicknesses. To have a visual view of the resource allocation problem, consider Fig. 3 again. In this figure, each cube represents the demand vector for each VNF such that each side magnifies the demand element for each resource type. The cubes of the same color are the building VNFs of a network slice. A slice thickness variable v_n determines how the cubes of the same slice expand or shrink with v_n as the expansion coefficient.

Since the demand vectors are specified by the slice providers, same slices might provide different demand vectors for their VNFs. Moreover, they may lie about their demand vectors with the purpose of getting more resources in the resource allocation process. An enhanced distributed resource allocation scheme is proposed in Section V to mitigate the misbehavior of non-collaborative slices.

We assume that slice operators have separate utility functions as a function of the amount of resources allocated to the slice's functions. Since the allocations of each slice scale with the slice thickness v_n, the slice n utility function can be denoted by $U_n(v_n)$. It is assumed that the slice utility function is increasing, strictly concave and continuously differentiable function of v_n. These assumptions on the utility functions are realistic assumptions as each slice utility/revenue will be increasing with respect to its allocated resources but the slope of utility growth decreases by increasing the allocations, e.g., due to limited number of UEs. The problem to address here is to find the slice thickness variables v_n such that the total system utility is optimized.

IV. Optimal Resource Allocation for a System With Collaborative Slices

In this section, we present the problem formulation for resource allocation in a system with collaborative slices. A collaborative slice is willing to collaborate with other slices and participate in a distributed framework for solving the resource allocation problem. The demand vectors declared by a collaborative slice is true and accurate. For such a system, we first formulate the resource allocation problem as a convex optimization problem maximizing the overall system utility function as a function of the slice thickness variables. We then present a fully distributed solution for the allocation problem based on the notion of auction theory and show the existence of a unique Nash equilibrium for the auction game. It is also shown that the Nash solution coincides with the solution of the centralized optimization problem, i.e., the auction game achieves 100% efficiency.

A. Problem Formulation

Based on the assumptions and the presented system model, we can formulate the virtualized 5G resource allocation into the following optimization problem.

Centralized System Optimization - Collaborative Slices:

$$\begin{aligned}&\underset{\mathbf{v}}{\texttt{Maximize:}}\ \sum_{n=1}^{N} U_n(v_n)\\&\texttt{Subject to:}\ \sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} \leq R_{\ell,k} \quad \forall \ell, \forall k\\&\qquad\qquad\quad v_n \geq 0 \quad \forall n\end{aligned} \tag{1}$$

The objective of Problem (1) is to maximize the overall system utility defined as the sum of the slices' utility functions. The constraints of this problem ensure that the slice thickness allocations will not violate the capacity limits of each resource in each DC. The centralized system optimization problem is a convex optimization problem in terms of the slice thickness variables v_n. This is because the objective function is concave and the constraints are linear inequalities representing a compact feasible region. Therefore, the centralized optimization problem has a unique optimum solution [38].

By choosing a proper utility function $U_n(\cdot)$, we can achieve a trade-off between efficiency and fairness which depends on the specific choice of $U_n(\cdot)$. To capture the trade-off between efficiency and fairness, one may choose $U_n(\cdot)$ from the class of α-fair utility functions [21], [22]. Specifically, by choosing $U_n(\cdot)$ such that $U'_n(x) = x^{-\alpha}$, for some fixed parameter α, the optimal solution of the centralized system optimization satisfies α-proportional fairness in terms of slice thicknesses. Thus, the α-proportional utility function is defined by $U_n(v_n) = \frac{v_n^{1-\alpha}}{1-\alpha}$ for $\alpha > 0$ and $\alpha \neq 1$. For $\alpha = 1$, we have $U_n(v_n) = \log(v_n)$ which is equal to the limiting value of $\frac{v_n^{1-\alpha}}{1-\alpha}$ when $\alpha \to 1$. The α-proportional utility function with $\alpha = 1$ provides "proportional fair" allocation and when $\alpha \to \infty$, it provides "max-min" fairness.

The centralized system problem can be solved by well-known convex optimization methods, e.g., subgradient projection and interior-point methods by a central optimizer [38], [39].

B. Distributed Resource Allocation–Auction Game Approach

Due to the problems with centralized approaches, we propose the following distributed scheme for 5G resource allocation problem. The distributed scheme is based on application of auction theory by forming an auction game between the slices and the DCs. In this scheme, the network slices bid for each of the resources of the DCs they are placing a function on. Based on the bids submitted by the network slices, the price for each resource on each DC is determined and is announced to the network slices together with their calculated thickness values. Each slice thickness will be equal to the minimum of the slice thicknesses received from all DCs. On the other hand, each slice maximizes its payoff based on the prices received from the DCs and updates its bid for the next round of the game. It is shown that Nash equilibrium exists for such an auction, i.e., there exists an equilibrium slice thickness vector and an equilibrium resource price for each of the DCs' resources such that no network slice is willing to change its bid and its allocation. Furthermore, it is shown that Nash solution for the game problem is the same as the solution of the centralized system optimization Problem (1), i.e., the Nash equilibria of the game approach will achieve the full efficiency of the system optimization. The benefits of the proposed distributed scheme are the following:

- Convergence to the system optimal solution.
- No optimization third party involved and no information sharing between the slice providers and the DCs.
- DCs do not necessarily need to be under the same management. This provides flexibility for the slice provider to choose proper DCs for placing its functions, i.e., flexibility for different business models.

Game Setup: The resource allocation game is setup in the following items:

- Each Slice n offers an amount of $w^k_{f^n_m,\ell}$ for resource ℓ of DC k which is locating the VNF f^n_m. The bidding is done for all ℓ, k and m. We define $\mathbf{w} = (w^k_{f^n_m,\ell}, \forall n, \forall m, \forall k, \forall \ell)$ as the offer matrix.
- Each DC k calculates the price of each of its resources by using the following equation.

$$p_{\ell,k} = \frac{\sum_{n,m} w^k_{f^n_m,\ell}}{R_{\ell,k}} \tag{2}$$

  We define $\mathbf{p} = (p_{\ell,k}, \forall \ell, \forall k)$ as the resource price matrix.
- Each DC k calculates a slice thickness for each slice n for each VNF f^n_m and resource ℓ as follows.

$$v^k_{f^n_m,\ell} = \frac{w^k_{f^n_m,\ell}}{p_{\ell,k} d^k_{f^n_m,\ell}} \tag{3}$$

  We call these thicknesses as the *local thicknesses* since they are calculated just based on local and insufficient information for each resource of each DC.
- The resource price on each DC as well the local thicknesses are announced to the slices. Each slice calculates the offered final thickness by

$$v_n = \min_{m,\ell,k} \{v^k_{f^n_m,\ell}\}. \tag{4}$$

- Each slice further uses the resource prices to update its thickness by maximizing its overall payoff function $G_n(v_n; \mathbf{p})$ defined as

$$G_n(v_n; \mathbf{p}) = U_n(v_n) - v_n \sum_{\ell,m,k} d^k_{f^n_m,\ell} p_{\ell,k}. \tag{5}$$

Slice n Payoff Optimization:

$$\begin{aligned}&\texttt{Maximize:}\ G_n(v_n; \mathbf{p})\\&\texttt{Subject to:}\ v_n \geq 0\end{aligned} \tag{6}$$

- Each slice then updates its offer for each resource to each DC for each of its functions by

$$w^k_{f^n_m,\ell} = v^\star_n d^k_{f^n_m,\ell} p_{\ell,k}, \tag{7}$$

  where $v^\star_n$ is the solution of the slice payoff optimization Problem (6).

The game is considered to be converged if the distance of the thickness allocation from the DCs derived from (4) and the thickness derived from the slice payoff optimization problem in (6) is less than ϵ, i.e., *for all* n, $|v_n - v_n^\star| \leq \epsilon$ where ϵ is a given parameter of the algorithm and establishes a trade-off between the solution accuracy and the convergence speed. We now formally define the Nash equilibrium for the aforementioned game.

Definition 1: The game is in Nash equilibrium if there exists a pair of offer matrix and resource price matrix $(\mathbf{w}, \mathbf{p})$ *such that the slice payoff defined in (6) is maximized and the equilibrium resource price is determined according to (2), i.e.,*

$$G_n(v_n; \mathbf{p}) \geq G_n(\bar{v}_n; \mathbf{p}) \quad \text{for any} \quad \bar{v}_n \geq 0, \quad \forall n \qquad (8)$$

$$p_{\ell,k} = \frac{\sum_{n,m} w^k_{f^n_m,\ell}}{R_{\ell,k}} \quad \forall \ell,\ \forall k. \qquad (9)$$

In the following theorem, we show that the Nash equilibrium does exist for the described game and the resulting Nash thickness vector is equal to the solution of the centralized system optimization in (1).

Theorem 1: Assume that the slice utility functions are strictly concave, increasing and continuously differentiable. Nash equilibrium exists for the resource allocation game described above, i.e., there exists a pair $(\mathbf{w}, \mathbf{p})$ *such that (8) and (9) are satisfied. Furthermore, the Nash pair* $(\mathbf{w}, \mathbf{p})$ *will result in a unique thickness vector* $\mathbf{v}$ *derived from (4) such that it also solves the centralized system optimization in (1).*

Detailed proof of the theorem is provided in the Appendix. In summary, Theorem 1 proves the following statements:

- A unique Nash equilibrium exists for the proposed auction game, i.e., there exists an equilibrium slice thickness vector and an equilibrium resource price for each of the DCs' resources such that no network slice is willing to change its bid and its allocation.
- Nash solution for the game problem is equal to the solution of the centralized system optimization problem in (1), i.e., the gap between the game result in Nash equilibrium and the solution of Problem (1) is zero. Hence, the game approach achieves the full efficiency of the system optimization.

V. Optimal Resource Allocation for a System With Non-Collaborative Slices

In this section, we first show how a non-collaborative slice who lies about its demand vectors can abuse the system and ruin the resource fairness among the slices. Using the notion of Dominant Resource Fairness (DRF) [24], we propose a new resource optimization problem and provide a novel distributed solution for solving the problem.

Example: Consider a system with two slices and one DC. Assume that the demand vector for CPU, RAM, Bandwidth and Storage is (1, 1, 1, 1) for both slices and the resource capacity of the DC for all resource types is 10. Consider a max-min fair allocation of the resources to the slices. If both slices are collaborative and provide their true demand vectors to the DC, the total allocation of the resources to these two slices will be (5, 5, 5, 5) for both slices and the thickness variables would be $v_1 = v_2 = 5$. Now, suppose that Slice 1 is non-collaborative and declares (9, 9, 9, 9) as its demand vector. Using resource allocation Problem (1), the optimal thicknesses would be $v_1 = v_2 = 1$ but the amount of resources allocated to each slice would be (9, 9, 9, 9) for Slice 1 and (1, 1, 1, 1) for Slice 2, i.e., Slice 1 grabs 9 times more resources than Slice 2. Note that the resource allocation Problem (1) is still maximizing the system utility function and provides balanced thicknesses to the slices. However, the selfish behavior of Slice 1 affects the allocation to Slice 2 and ruins the fairness performance of the system. In the following sections, we proposed a new resource allocation problem formulation based on the notion of DRF which helps us to beat this problem. We first provide a biref review of the notion of DRF presented in [24] for multi-resource systems with the assumption that all resources are aggregated at one resource pool. We will extend the definition of DRF to be applicable to our multi-DC model in Section III.

A. Dominant Resource Fairness

Consider a multi-resource system with L different resource types and resource capacity R_ℓ for each resource type ℓ. Suppose that the demand of each user (slice) n for resource r is $d_{n,\ell}$. For each user n, the dominant resource is defined as $\rho_n = \underset{\ell}{\operatorname{argmax}} \frac{d_{n,\ell}}{R_\ell}$ which is the most scarce resource type for user n. Given a feasible allocation $\boldsymbol{a} = (a_{n,\ell}, \forall n, \forall \ell)$ where $\sum_n a_{n,\ell} \leq R_\ell$, the dominant share of a user n is defined as the share of user n on its dominant resource type, i.e.,

$$z_n := \frac{a_{n,\rho_n}}{R_{\rho_n}} \qquad (10)$$

Definition 2: An allocation $\boldsymbol{a}$ *is said to be DRF if it provides max-min fairness among the* z_n *variables associated to each user, i.e., the dominant share of no user can be increased without decreasing the dominant share of another user.*

B. Problem Formulation for Non-Collaborative Slices

We extend the notion of dominant share and dominant resource fairness to the system presented in III where the resources are aggregated in multiple data centers and the services are composed of a chain of virtual functions. We define the dominant vector $\nu_n = (\rho_n, \kappa_n, \theta_n)$ for each slice n as

$$\nu_n = (\rho_n, \kappa_n, \theta_n) = \underset{\ell,k,m}{\operatorname{argmax}} \frac{d^k_{f^n_m,\ell}}{R_{k,\ell}} \qquad (11)$$

The vector ν_n determines the most scare resource of a slice n among all its virtual functions over all data centers. ρ_n denotes the resource type, κ_n denotes the DC and θ_n denotes the VNF where maximization (11) occurs.

Recall that in our model, the allocations are defined in terms of the slice thicknesses. Given a slice thickness vector $\mathbf{v}$, we define the dominant share of slice n by

$$z_n := \frac{v_n d^{\kappa_n}_{f^n_{\theta_n},\rho_n}}{R_{\kappa_n,\rho_n}}. \qquad (12)$$

In (12), the nominator is equal to the total allocation of slice n for resource ρ_n on DC κ_n for VNF θ_n. Note that the dominant vector ν_n is only dependent on the demand vectors $\mathcal{D}_n$ and the resource capacities at the data centers and therefore, the dominant share z_n is uniquely determined for each slice n independent of its allocation (thickness). We define

$$\beta_n := \frac{d^{\kappa_n}_{f^n_{\theta_n},\rho_n}}{R_{\kappa_n,\rho_n}}, \tag{13}$$

and therefore, dominant share z_n and slice thickness v_n are related by

$$z_n = v_n \beta_n. \tag{14}$$

Similar to Def. 2, we can define the dominant resource fair allocation as follows:

Definition 3: A slice thickness allocation $\mathbf{v}$ *is said to be DRF if it provides max-min fairness among the* z_n *variables associated to each slice, i.e., the dominant share of no slice* n *can be increased without decreasing the dominant share of another slice* m.

We can extend the notion of DRF in Def. 3 to more generalized fairness criteria, i.e., α-proportional fairness. In this regard, we define the following resource allocation optimization problem in which the objective is defined as the summation of the slice utility functions as a function of the dominant shares z_n.

Centralized System Optimization - Non-collaborative Slices:

$$\begin{aligned} &\underset{\mathbf{v}}{\texttt{Maximize:}} \ \sum_{n=1}^{N} U_n(z_n) \\ &\texttt{Subject to:} \ \sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} \le R_{\ell,k} \quad \forall \ell, \forall k \\ &\qquad\qquad\quad z_n = v_n\beta_n \quad \forall n \\ &\qquad\qquad\quad v_n \ge 0 \quad \forall n \end{aligned} \tag{15}$$

where β_n is defined in (13). If the utility functions are α-proportional fairness functions, i.e., $U_n(z_n) = \frac{z_n^{1-\alpha}}{1-\alpha}$ for $\alpha > 0$ and $\alpha \neq 1$, we can observe that by letting $\alpha \longrightarrow \infty$, the resource allocation Problem (15) provides max-min fairness among the dominant share variables z_n, i.e., the solution of (15) provides DRF among the network slices. Thus, the optimization Problem (15) provides broader fairness properties to the resource allocation problem.

In contrast to the optimization Problem (1), if a network slice is non-collaborative and lies about its demand vectors, it cannot compromise the system and grab more resources than other slices. More specifically, if slice n declares a larger demand vector than its real demand vector, it makes its β_n variable larger. Assuming that the system is designed to provide DRF among the slices, the optimization Problem (15) assigns the liar slice a smaller thickness v_n since under DRF it tries to balance the z_n variables among the slices.

The optimization Problem (15) can be reformatted as

$$\begin{aligned} &\underset{\mathbf{z}}{\texttt{Maximize:}} \ \sum_{n=1}^{N} U_n(z_n) \\ &\texttt{Subject to:} \ \sum_{n=1}^{N}\sum_{m=1}^{M} z_n b^k_{f^n_m,\ell} \le R_{\ell,k} \quad \forall \ell, \forall k \\ &\qquad\qquad\quad z_n \ge 0 \quad \forall n \end{aligned} \tag{16}$$

where $b^k_{f^n_m,\ell} = \frac{d^k_{f^n_m,\ell}}{\beta_n}$. This problem is similar to Problem (1) with the same utility function except the linear constraint terms are normalized with the β_n variables of the slices. Hence, this problem is also a convex optimization problem in terms of dominant share variables z_n and has a unique optimum solution [38] which can be obtained using well-known convex optimization methods, e.g., subgradient projection and interior-point methods [38].

In the following section, we propose a distributed solution similar to the one proposed for the collaborative system. The design of the distributed scheme for the non-collaborative system is tricky due to introducing the β_n variables in the optimization problem.

C. Distributed Resource Allocation for Non-Collaborative Slices

To solve Problem (16) in a distributed fashion, we cannot directly use the scheme presented in Sub-section IV-B unless the data centers as well as the slices know the β_n variables which involves sharing the resource capacities of the data centers with the slices and this conflicts the purpose. In the following section, we proposed a game setup whose Nash equilibrium solution provides the unique solution of the system optimization Problem (16). In the proposed distributed solution, the data centers only need to share the following variables with each other.

$$s_{n,k} = \max_{\ell,m} \frac{d^k_{f^n_m,\ell}}{R_{k,\ell}} \tag{17}$$

Since the data centers do not know the demand vectors of the slices on the other data centers, sharing $s_{n,k}$ variables does not disclose any information regarding the data centers capacities. The data centers then calculate the β_n for each slice n by using the following equation.

$$\beta_n = \max_k s_{n,k} \tag{18}$$

Game Setup: The resource allocation game is setup in the following items:

- Each Slice n offers an amount of $w^k_{f^n_m,\ell}$ for resource ℓ of DC k which is locating the VNF f^n_m. The bidding is done for all ℓ, k and m. We define $\mathbf{w} = (w^k_{f^n_m,\ell}, \forall n, \forall m, \forall k, \forall \ell)$ as the offer matrix.
- Each DC k calculates the price of each of its resources by using

$$p_{\ell,k} = \frac{\sum_{n,m} w^k_{f^n_m,\ell}}{R_{\ell,k}}, \tag{19}$$

and $\mathbf{p} = (p_{\ell,k}, \forall \ell, \forall k)$ is the resource price matrix.

- Each DC k calculates a local dominant share variable $z^k_{f^n_m,\ell}$ for each slice n for each VNF f^n_m and resource ℓ as follows.

$$z^k_{f^n_m,\ell} = \frac{w^k_{f^n_m,\ell}}{p_{\ell,k} b^k_{f^n_m,\ell}} \tag{20}$$

  We call these variables as the *local dominant shares* since they are calculated just based on local and insufficient information for each resource of each DC.
- In contrast to the game setup in Section IV-B, the DCs do not announce their prices to the slices. Instead, the DCs provide a normalized price for each resource type to each slice n. More specifically, the DCs send $\pi^n_{\ell,k} = \frac{p_{\ell,k}}{\beta_n}$ as the resource price of resource ℓ to slice n. Note that the DCs provide a different resource price to each slice as the slice's β_n variables are different. The reason for this normalization is that this resource unit price is used in the payoff optimization problem by the slices. Since the slices only know their own demand vectors and are not aware of the β_n variables, when using the normalized price in their payoff optimization, they instinctively incorporate it in their payoff optimization. The normalized resource prices as well the local dominant shares are announced to the slices and each slice calculates the offered final dominant share by

$$z_n = \min_{m,\ell,k} \{z^k_{f^n_m,\ell}\}. \tag{21}$$

- Each slice further uses the normalized resource prices to update its dominant share variable by maximizing its overall payoff function $G_n(z_n;\pi)$ defined as

$$G_n(z_n;\pi) = U_n(z_n) - z_n \sum_{\ell,m,k} d^k_{f^n_m,\ell} \pi^n_{\ell,k}. \tag{22}$$

  Slice n Payoff Optimization:

$$\begin{aligned} &\texttt{Maximize: } G_n(z_n;\pi) \\ &\texttt{Subject to: } z_n \geq 0 \end{aligned} \tag{23}$$

- Each slice then updates its offer for each resource to each DC for each of its functions by

$$w^k_{f^n_m,\ell} = z^\star_n d^k_{f^n_m,\ell} \pi^n_{\ell,k}, \tag{24}$$

  where $z^\star_n$ is the solution of the slice payoff optimization Problem (23).

We use a similar convergence criterion for the game as we used in Section IV-B, i.e., the game is considered to be converged if *for all* n, $|z_n - z^\star_n| \leq \epsilon$. After the game is converged, the actual slice thickness is announced by the DCs to the slices, which is calculated using (14). We now formally define the price equilibrium for non-collaborative slices game.

Definition 4: The game is in Nash equilibrium if there exists a pair of offer matrix and resource price matrix $(\mathbf{w}, \mathbf{p})$ *such that the slice payoff defined in (23) is maximized and the equilibrium resource price is determined according to (19), i.e.,*

$$G_n(z_n;\pi) \geq G_n(\bar{z}_n;\pi) \text{ for any } \bar{z}_n \geq 0, \quad \forall n \tag{25}$$

$$\pi^n_{\ell,k} = \frac{p_{\ell,k}}{\beta_n} \quad \forall \ell,\ \forall k. \tag{26}$$

$$p_{\ell,k} = \frac{\sum_{n,m} w^k_{f^n_m,\ell}}{R_{\ell,k}} \quad \forall \ell,\ \forall k. \tag{27}$$

Note that the main difference of the game for the non-collaborative slices from the collaborative slices (presented in Section IV-B) are the following:

- The DCs calculate and share variables $s_{n,k}$ to be able to calculate the β_n for each slice.
- The DCs announce a normalized resource price to each slice that might be different from the price announced to other slices.

Using the techniques presented above, in the following theorem we formally state that the Nash equilibrium does exist for the described game and the resulting Nash thickness vector is equal to the solution of the centralized system optimization in (16) which is equivalent to system optimization (15).

Theorem 2: Assume that the slice utility functions are strictly concave, increasing and continuously differentiable. Nash equilibrium exists for the resource allocation game described above, i.e., there exists a pair $(\mathbf{w}, \mathbf{p})$ *such that (25) and (27) are satisfied. Furthermore, the Nash pair* $(\mathbf{w}, \mathbf{p})$ *will result in a unique dominant share vector* $\mathbf{z}$ *derived from (21) such that it also solves the centralized system optimization in (16) which is equivalent to system optimization (15).*

Proof: The proof of the theorem is following similar steps used in the proof of Theorem 1, i.e., by taking the Lagrangian of the optimization Problem (16) and showing that the Nash conditions (25) and (27) are the same as the optimality conditions of the centralized system optimization problem. Note that since the form and the constraints of Problem (16) is similar to Problem (1) and also the form and the constraint of the payoff optimization functions in (5) and (6) are similar to (22) and (23), we can follow similar approach as used in the proof of Theorem 1. Details will be redundant and are skipped here for brevity. □

VI. A System-Level Implementation of the Algorithms

This section presents a system-level implementation of the described algorithms in Sections IV and V. Main building blocks of the system are presented as well as the necessary communication messages between different entities. The focus is to provide a high-level system description of the presented idea without providing a detailed system/software implementation which is out of the scope of this paper. In the following, we only present the system description for the non-collaborative case. The system description for the collaborative case is simpler and follows a similar setup.

A. System Blocks

The presented distributed resource allocation scheme in the previous sections is based on running separate concurrent games on all the resources of all the data centers over which the VNFs are placed. A software block called Game Controller is responsible to perform and control a game on each data center for each resource type. Since we assume the available resources are CPU, RAM, storage and bandwidth, there exist four blocks (software modules) namely CPU Game Controller, RAM Game Controller, Storage Game Controller and Bandwidth Game Controller in each data center. The game controllers of each data center are all instantiated and managed by an entity called Data Center Manager.

On the other hand, the slices need to participate in the games which are started and controlled by the data centers' game controllers. Therefore, each slice contains a separate Game Player for all the games it is participating in. All the Game Players of a slice are under the control of Slice Manager which is responsible to configure the demand vectors and instantiate the game players. Fig. 4 shows the Game Controllers, Slice Manager and its Game Players in a system with two data centers, two slices and four resource types, CPU, RAM, storage and bandwidth. The Slice Manager and its Game Players are placed in the slice provider domain which might be outside the data centers domain. The Data Center Manager and the Game Controllers are located in each data center as observed in Fig. 4. The arrow between the each Game Controller and a Player depicts a logical communication channel which carries game data and control variables and parameters, e.g., demand vectors, bid values, allocations, etc. The communication messaging can be implemented using JSON (JavaScript Object Notation) file format [40].

B. System Operation

The system operation is summarized in the following bullets:

- The system starts when at least one slice becomes activated and requests for resource allocation to accommodate its VNFs in the data centers. This is done by the Slice Manager by instantiating the required Game Players and then sending an `authenticationRequest` message to all the Data Center Managers. The `authenticationRequest` message not only is used for slice authentication but also contains the slice demand vectors for that particular data center. The Data Center Managers respond with `authenticationResponse` messages. After the authentication process is done successfully, the Data Center Manager provides the proper demand element to each Game Controller. At this point, the Game Controllers are aware of the slice demand elements $d^k_{f^n_m,\ell}$. Therefore, the Data Center Manager can calculate $s_{n,k}$ variables using Eq. (17).
- The Data Center Managers then exchange $s_{n,k}$ variables with each other.
- The Data center Managers then calculate β_n for the activated slice. At this point, all the Data Center Managers will have a common view of β_n for the new activated slice.
- The Game Players then send a `gameRequest` message to the Game Controllers. Upon receiving a `gameRequest` message, a Game Controller sends a `gameStart` message to all the active slices (i.e., the ones it has their demand element).
- The Game Players respond with a `gameStartAck` message to the Game Controllers.
- The Game Controller sends a `bidRequest` message to its active players and requests for a bid value.
- The Game Players start the first bid with an arbitrary bid value since they do not have any view of the system and the allocations. Slice bid values are sent to the Game Controllers using `bid` messages.
- Each Game Controller calculates its price using (19). It then calculates the slice local dominant shares using (20). Using the resource price and the β_n variable, each Game Controller calculates slice resource prices $\pi^n_{\ell,k} = \frac{p_{\ell,k}}{\beta_n}$. The Slice resource price together with the local dominant shares are responded back to each player in an `allocationUpdate` message.
- Upon receiving the `allocationUpdate` message from all the Game Controllers, the Slice Manager calculates offered final dominant share z_n using Eq. (21).
- Using the slice resource prices $\pi^n_{\ell,k}$, the Slice Manager also solves the payoff optimization Problem (23). Since this optimization problem is a single variable convex problem, a closed-form solution can be found offline and be used to calculate the optimal $z^\star_n$.
- The Slice Manager then checks if its games are converged by checking if $|z_n - z^\star_n| \leq \epsilon$. If it is converged, it sends a `gameConverged` message to all Game Controllers.
- Each Player also calculates a new bid value using Eq. (24) and sends a `bid` message to the Game Controllers.
- After Each Game Controller receives a `gameConverged` message from all of its players, it sends a `gameEnd` message to the players. At this time, all the games are over.

C. Computation and Communication Overheads

As mentioned in the previous sections, the DCs only need to share $s_{n,k}$ variables for each slice. Other than that, there is no communication required during the live operation of the algorithm between the data centers. Therefore, its communication overhead is almost zero as it is done just once before running the algorithm. The communication between the data centers and the slice providers are the following:

- Authentication messaging which performs authentication and demand vector provisioning. This is done just once when a slice is activated. Therefore, its communication overhead is negligible.
- Each game starts with `gameRequest`, `gameStart` and `gameStartAck` messages which occur only once at the beginning of the game. Games are also finalized by `gameConverged` and `gameEnd` messages. Therefore, we can also neglect the overhead of these messages in the system operation.
- Iterations which are the main pieces of the algorithm run, are composed of a `bid` message and an `allocationUpdate` message.

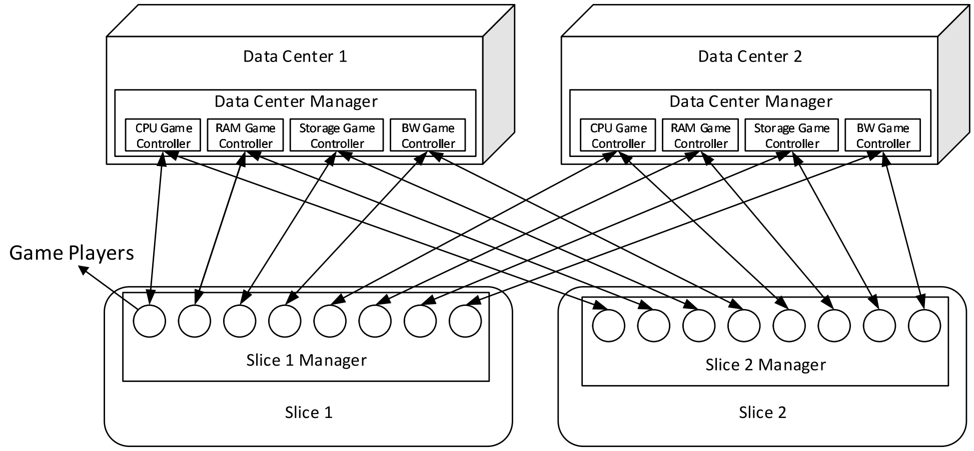

Fig. 4. System blocks: Data center manager, game controllers, slice manager and game players.

In all the cases mentioned above, since the DC-Slice messages of all the slices and data centers are concurrent, the communication overhead is independent of the number of slices and also the number of DCs.

In terms of computation overhead, each game iteration is composed of a number of basic arithmetic (multiplications and divisions) except the payoff maximization problem which needs to be solved locally by each slice in (6) and (23). Since the payoff maximization objective is a single variable concave function, it can be solved offline and a closed-form solution can be derived and be used to determine $z_n^\star$ and $v_n^\star$. Therefore, the proposed solution does not impose any significant computation overhead for convergence.

In summary, the major latency component in each iteration of the algorithm belongs to the communication overhead of `bid` and `allocationUpdate` messages which is less than a few milliseconds.

VII. Simulation Results

In this section, we present our simulation/numerical results in which we first confirm the convergence of the distributed schemes for both collaborative and non-collaborative systems to the optimal thickness vector via numerical analysis over sample system setups. We then compare the performance of different α-proportional fairness utility functions in terms of resource allocation efficiency. We also compare the performance of both collaborative and non-collaborative systems in terms of fairness and show how the solution in non-collaborative system can protect the system fairness behavior against non-collaborative slices. Finally, we compare the optimal solution with the solution of two heuristic schemes in terms of system utility and resource utilization. The results are only shown for the collaborative system in this case since the objective function in the non-collaborative is a different one. However, same heuristics can be proposed for the non-collaborative system and similar conclusions can be drawn. Hence, to avoid redundancy we skip presenting the numerical results for the non-collaborative system. Recall that the α-proportional utility function is defined by $U_n(v_n) = \frac{v_n^{1-\alpha}}{1-\alpha}$ for $\alpha > 0, \alpha \neq 1$ and $U_n(v_n) = \log(v_n)$ for $\alpha = 1$.

TABLE I
Data Centers Resource Settings

DC	CPU (cores)	RAM(GB)	BW(Gbps)	Storage(TB)
1	5000	10000	1000	2000
2	5000	5000	2000	5000
3	5000	5000	2000	10000

In all the presented results, we only used the Nash equilibrium solution of the proposed distributed schemes as we have already proven theoretically that the Nash equilibrium solution is exactly equal to the centralized optimization solution. Nevertheless, in all simulation results presented below, we solved the optimization problems using convex solver cvx [39] and confirmed that the Nash solution is equal to the centralized solution.

We consider a system consisting of three DCs and 100 network slices. Each network slice is composed of 5 VNFs. VNF 1 is placed at DC 1, VNFs 2 and 3 are placed at DC 2 and VNFs 4 and 5 are placed at DC 3 for all slices. Each DC contains 4 types of resources, CPU, memory, network bandwidth and storage. Table I shows the amount of available resources in each DC. The elements of demand vectors of all network slices, for all the functions are randomly selected as follows: for CPU, from the interval $[1\ 10]$ cores; for RAM, from the interval $[1\ 10]$ GB; for storage, from the interval $[1\ 10]$ TB and for network bandwidth, from the interval $[0.25\ 2.5]$ Gbps.

A. Convergence of the Distributed Scheme

To show the convergence of the distributed algorithm, we consider a system in which each slice has an α-proportional fair utility function where α is chosen randomly for each slice from the interval 1 to 10. Recall that we introduced ϵ in Section IV-B as the precision parameter for the convergence of the resource allocation game. In Fig. 5 and 6, we have measured the number of game iterations required for the

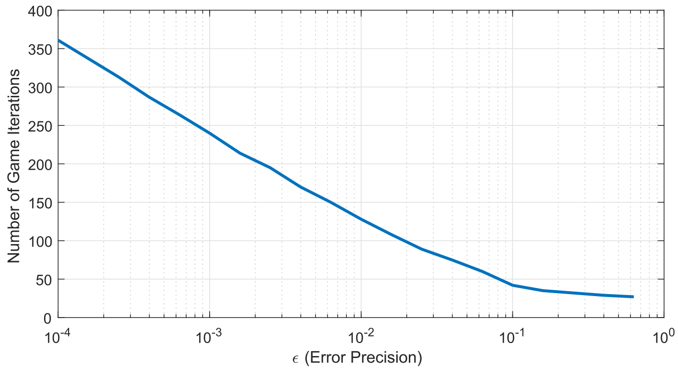

Fig. 5. Convergence of the distributed scheme for collaborative slices.

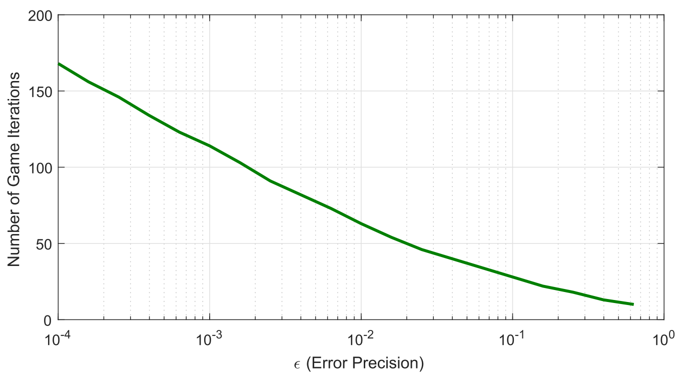

Fig. 6. Convergence of the distributed scheme for non-collborative slices.

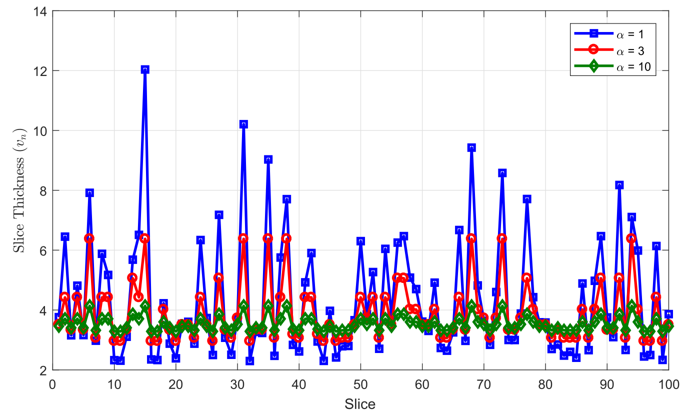

Fig. 7. Slice thickness for α-proportional fair utility functions ($\alpha = 1, 3, 10$) with collaborative slices.

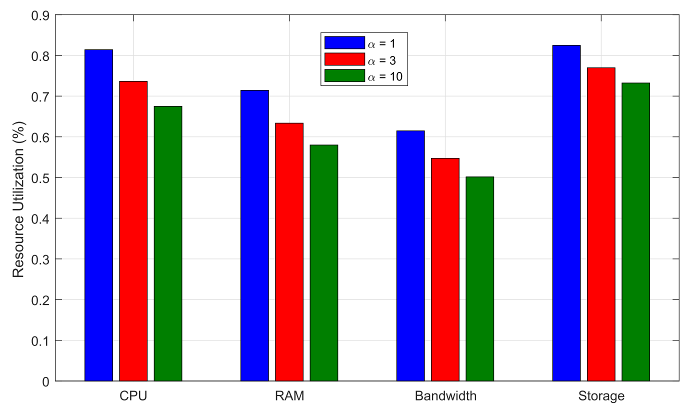

Fig. 8. Resource utilization for α-proportional fair utility functions ($\alpha = 1, 3, 10$) with collaborative slices.

system to converge with precision ϵ for sample values of ϵ ranging from 0.5 to 10^{-4} for the collaborative slices and non-collaborative slices, respectively. It is observed that for precision $\epsilon = 0.1$ which is a reasonable precision value, less than 40 iterations is enough for the system to converge. Moreover, for both systems it is observed that the required number of iterations is decreasing linearly as the precision error grows logarithmically meaning that we can achieve sophisticated precision errors efficiently by linearly increasing the number of iterations.

B. Resource Efficiency for Different α-Proportional Utilities

In this section, we compare the allocation and resource utilization under 3 different α-proportional fairness utility functions. We assume that all the network slices have the same α-proportional fairness utility with $\alpha = 1, 3, 10$.

Fig. 7 shows the slice thickness values for each slice for different values of α under the collaborative system. With $\alpha = 1$, allocations are proportionally fair, i.e., the system tries to maintain a balance between fairness and resource utilization (we see the resource utilization result later). By increasing the α parameter, the fairness behavior of the system tends to max-min fair allocation where resource utilization is ignored and the objective is maximizing the minimum allocation among the network slices. Fig. 8 shows the resource utilization for each α. Note that the resource utilization is measured for each resource per DC as the ratio of the total allocation of each resource and the available amount of that resource in the DC and then it is averaged over the three DCs. It is observed that with $\alpha = 10$, the thickness allocations are almost equal for every slice while the resource utilization efficiency is the least. However, with $\alpha = 1$ there are fluctuations in the thickness allocations and the resource utilization is the maximum for all types of resources.

Fig. 9 shows the slice thicknesses under the system with non-collaborative slices. In contrast with the collaborative system, even with $\alpha = 10$ the resource allocation solution does not try to balance the thicknesses among the competing slices. The reason is that the objective function in the resource allocation problem for non-collaborative slices is not designed to provide fairness over the slice thicknesses. Instead it tries to provide fairness over the dominant resource share of each slice. Fig. 10 presents the result under non-collaborative system for the dominant resource share. It is observed that with $\alpha = 10$, the resource allocation optimization provides max-min fairness over the slice dominant shares. Using a smaller α in the utility functions provides better resource utilization as observed in Fig. 11 and poorer fairness among the slice dominant shares.

C. Protection of Collaborative Slices Against Non-Collaborative Slices

In this scenario, we consider the same setup as in the previous scenarios, except we assume that the last 20 slices

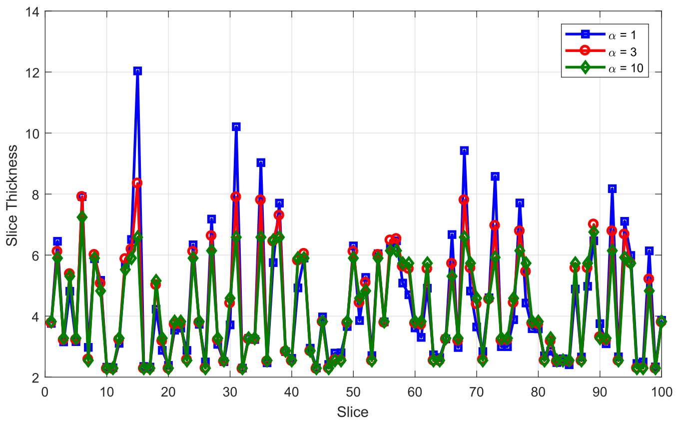

Fig. 9. Slice thickness for α-proportional fair utility functions ($\alpha = 1, 3, 10$) with non-collaborative slices.

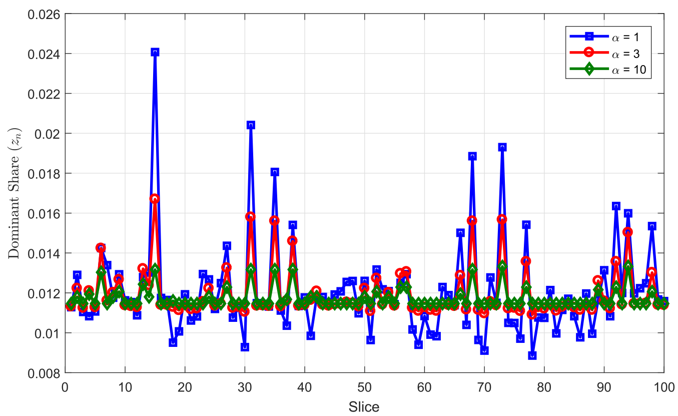

Fig. 10. Dominant shares for α-proportional fair utility functions ($\alpha = 1, 3, 10$) with non-collaborative slices.

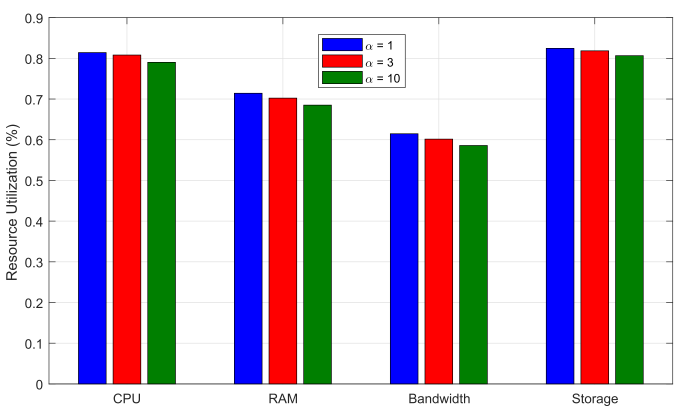

Fig. 11. Resource utilization for α-proportional fair utility functions ($\alpha = 1, 3, 10$) with non-collaborative slices.

are non-collaborative and lie about their demand vectors and declare them 20 times larger than the actual demand vectors. For this scenario, we assume that $\alpha = 10$ which means that the system is willing to provide max-min fairness among the slices. While the resource allocation Problem (1) proposed for collaborative slices tries to balance the slice thicknesses (see Fig. 7) under max-min fairness, it does not provide fairness over the actual total allocation as observed in Fig. 12. This is

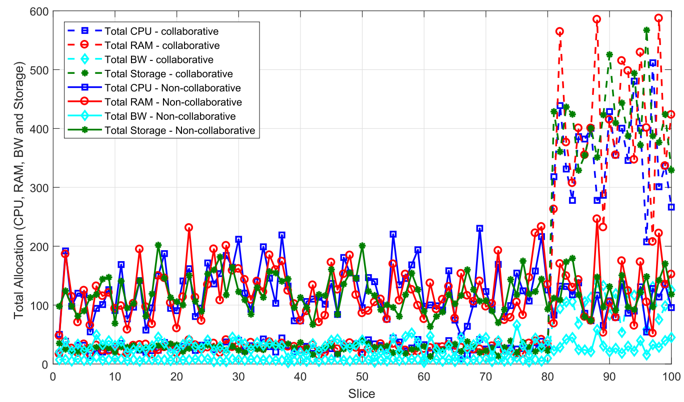

Fig. 12. Resource allocation for max-min fair utility functions with 20 non-collaborative slices.

due to the fact that the optimization (1) ignores the demand vectors in its objective function. In contrast, the resource optimization Problem (15) involves the demand vectors both in the constraints as well as the objective function (the β_n variables) to provide fairness over the dominant shares of the slices. If a slice lies about its demand vectors and provides larger demand vectors to the DCs, this makes its β_n variable larger which means that the resource allocation solution results in smaller slice thickness for that slice. In other words, a non-collaborative slice cannot grab more resources by lying about its demand vectors as it penalizes itself.

D. Comparison With Heuristic Sub-Optimal Schemes

In this section, we compare the performance of the optimal distributed scheme under the collaborative system with two heuristic schemes. The first scheme allocates the resources of each DC uniformly among its VNFs. In this scheme, not all resources allocated to a network slice are useful for it. The effective utilized resources for each slice depends on its allocations as well as its demand vectors. We call this allocation as the uniform allocation. The second scheme allocates the available resources to the VNFs based on the demand vectors of the VNFs, i.e., the allocations are weighted based on the elements of the demand vectors of the VNFs for each resource type and for each DC. The resultant allocation for this scheme is such that all network slices will get the same slice thickness. This allocation is called the demand-weighted scheme. Note that in both heuristic schemes, the maximal information needed for resource allocation is the demand vectors and they operate independently of the utility functions. This assumption is due to the fact that slice providers are reluctant to share their private information with DCs.

For this scenario, we again assume that each network slice has an α-proportional utility function with a randomly selected α from 1 to 10 for each network slice. Fig. 13 compares the objective value of the resource optimization Problem (1) for the optimal distributed scheme and the two heuristics. Fig. 14 shows the resource utilization comparison among the optimal and the heuristic approaches. It is observed that the optimal allocation results in the maximum overall system

Fig. 13. Objective values - optimal and heuristic schemes.

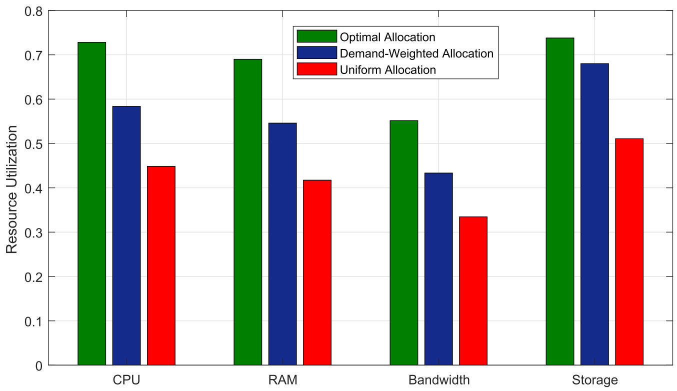

Fig. 14. Resource utilization comparison for the optimal and the heuristic schemes.

utility and also outperforms the heuristic ones in terms of resource utilization.

VIII. Conclusions

We have introduced a model of resource allocation for 5G networks incorporating the notions of network function virtualization and end-to-end network slicing. We formulated the optimal resource allocation as a convex problem with the objective to maximize the overall system utility function as a function of the slice resource allocations indicated by slice thickness variables. We introduced a distributed auction-based approach for collaborative slices to solve the system optimization problem and showed theoretically that the auction game has a unique Nash equilibrium and also it converges to the global optimal system solution. By using an example we showed how selfish behavior of non-collaborative slices impacts the resource fairness among the slices. We then formulated a resource allocation problem based on the notion of DRF and proposed a distributed solution to solve it. Simulation results were provided to evaluate the performance of the distributed schemes in terms of convergence and resource utilization for different utility functions and protection of collaborative slices against non-collaborative ones. We also compared the performance of the optimal schemes with two heuristic approaches.

Appendix
Proof of Theorem 1

Proof: The proof follows by taking the Lagrangian of the optimization problem and showing that the Nash conditions (8) and (9) are the same as the optimality conditions of the centralized system optimization problem as used in [21]. Since the centralized system optimization problem is strictly feasible (at least $\mathbf{v} = 0$ is in the feasible region) then Slater condition guarantees that the strong duality for this problem holds [38]. Also since the objective function is strictly concave, increasing and continuously differentiable and the feasible region is compact, the solution is unique [38]. The Lagrangian form for this problem is given by

$$L(\mathbf{v};\lambda) = \sum_{n=1}^{N} U_n(v_n) - \sum_{\ell,k} \lambda_{\ell,k} \left(\sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} - R_{\ell,k} \right) \tag{28}$$

where λ is the matrix of Lagrangian variables. Assuming that $\mathbf{v}^*$ is the optimal vector of slice thicknesses for Problem (1), KKT (Karush-Kuhn-Tucker) conditions ensure that there exist Lagrange multipliers $\lambda_{\ell,k}$ such that the following conditions (primal and dual feasibility, complementary slackness and vanishing of the gradient of the Lagrangian) are hold [38].

$$\sum_{n=1}^{N}\sum_{m=1}^{M} v^*_n d^k_{f^n_m,\ell} \leq R_{\ell,k}, \quad \forall \ell,\ \forall k \tag{29}$$

$$v^*_n \geq 0, \quad \forall n \tag{30}$$

$$\lambda_{\ell,k} \geq 0 \quad \forall \ell,\ \forall k \tag{31}$$

$$\lambda_{\ell,k} \left(\sum_{n=1}^{N}\sum_{m=1}^{M} v^*_n d^k_{f^n_m,\ell} - R_{\ell,k} \right) = 0 \quad \forall \ell,\ \forall k \tag{32}$$

$$v^*_n > 0 \Rightarrow U'_n(v^*_n) = \sum_{\ell,m,k} \lambda_{\ell,k} d^k_{f^n_m,\ell} \quad \forall n \tag{33}$$

$$v^*_n = 0 \Rightarrow U'_n(v^*_n) \leq \sum_{\ell,m,k} \lambda_{\ell,k} d^k_{f^n_m,\ell} \quad \forall n \tag{34}$$

The Nash equilibrium is at a point where given the resource prices, the payoff of all the slices are maximized, i.e., for all n,

$$G'_n(v_n;\mathbf{p}) = U'_n(v_n) - \sum_{\ell,m,k} p_{\ell,k} d^k_{f^n_m,\ell} \leq 0,\ v_n \geq 0 \tag{35}$$

$$v_n \left(G'_n(v_n;\mathbf{p}) \right) = 0 \Rightarrow v_n \left(U'_n(v_n) - \sum_{\ell,m,k} p_{\ell,k} d^k_{f^n_m,\ell} \right) = 0 \tag{36}$$

On the other hand, the equilibrium resource prices satisfy either of the following equations:

$$\sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} = R_{\ell,k}$$

$$\text{or } \sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} \leq R_{\ell,k} \text{ and } p_{\ell,k} = 0 \tag{37}$$

The first equation says that with the current price and thicknesses the capacity $R_{\ell,k}$ is totally used up. The second one says that if the capacity is not used up (i.e., there is no competitive demand for it in the auction) its price tends to zero. Note that in equilibrium, since the thickness v_n is determined from (4), the overall demand for some of resources on some DCs might not be fully demanded, i.e., $\sum_{n=1}^{N}\sum_{m=1}^{M} v_n^* d^k_{f^n_m,\ell} \le R_{\ell,k}$ and thus for those resources, the price $p_{\ell,k}$ tends to zero. Conditions (37) can be summarized as

$$p_{\ell,k}\left(\sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} - R_{\ell,k}\right) = 0,$$
$$p_{\ell,k} \ge 0, \quad \sum_{n=1}^{N}\sum_{m=1}^{M} v_n d^k_{f^n_m,\ell} \le R_{\ell,k}. \tag{38}$$

We observe that Conditions (29) – (34) are similar to Conditions (35) – (38). Since the primal and dual centralized system optimization have unique solutions $\mathbf{v}^*$ and λ, by letting $p_{\ell,k} = \lambda_{\ell,k}$, we observe that the equilibrium price matrix $\mathbf{p}$ does exist and is unique. We now show that the Nash equilibrium point in the resource allocation game is the same as the optimal point in the centralized system optimization problem. From (6) at the equilibrium, we know that

$$U_n(v_n) - v_n \sum_{\ell,m,k} d^k_{f^n_m,\ell} p_{\ell,k} \ge U_n(v_n^*) - v_n^* \sum_{\ell,m,k} d^k_{f^n_m,\ell} p_{\ell,k} \tag{39}$$

By summing over all slices,

$$\sum_n U_n(v_n) \ge \sum_n U_n(v_n^*) + \sum_{n,\ell,m,k} v_n d^k_{f^n_m,\ell} p_{\ell,k} - \sum_{n,\ell,m,k} v_n^* d^k_{f^n_m,\ell} p_{\ell,k} \tag{40}$$

From (38), we have

$$\sum_{n,\ell,m,k} v_n d^k_{f^n_m,\ell} p_{\ell,k} = \sum_{\ell,k} p_{\ell,k} R_{\ell,k} \tag{41}$$

From the constraints of Problem (1), we also have

$$\sum_{n,\ell,m,k} v_n^* d^k_{f^n_m,\ell} p_{\ell,k} \le \sum_{\ell,k} p_{\ell,k} R_{\ell,k} \tag{42}$$

Using (40) – (42), we observe that

$$\sum_n U_n(v_n) \ge \sum_n U_n(v_n^*) \tag{43}$$

Since, $\mathbf{v}^*$ is the global optimal point for the centralized system Problem (1), we must have $\sum_n U_n(v_n) = \sum_n U_n(v_n^*)$ and since the solution is unique, we have $\mathbf{v} = \mathbf{v}^*$. □

References

[1] *Network Functions Virtualization (NFV); Architectural Framework*, ETSI, Sophia Antipolis, France, Oct. 2013.
[2] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.
[3] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.
[4] T. Taleb, "Toward carrier cloud: Potential, challenges, and solutions," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 80–91, Jun. 2014.
[5] A. Laghrissi, T. Taleb, and M. Bagaa, "Conformal mapping for optimal network slice planning based on canonical domains," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 519–528, Mar. 2018, doi: 10.1109/JSAC.2018.2815436.
[6] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end QoS based on multi-paths routing using SDN technology," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.
[7] N. Nikaein *et al.*, "Network store: Exploring slicing in future 5G networks," in *Proc. Int. Workshop Mobility Evolving Internet Archit.*, Paris, France, Sep. 2015, pp. 8–13.
[8] X. Costa-Pèrez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 27–35, Jul. 2013.
[9] P. Rost *et al.*, "Mobile network architecture evolution toward 5G," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 84–91, May 2016.
[10] K. Samdanis, X. C. Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 32–39, Jul. 2016.
[11] *IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, document M Series 2083, International Telecommunication Union, Sep. 2015.
[12] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 469–484, Mar. 2018, doi: 10.1109/JSAC.2018.2815398.
[13] *C-RAN: The Road Towards Green RAN-V2.5*, China Mobile, Hong Kong, Oct. 2011.
[14] T. Taleb *et al.*, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Netw.*, vol. 29, no. 2, pp. 78–88, Mar. 2015.
[15] *Technical Specification Group Services and Systems Aspects; System Architecture for the 5G System; Stage 2, Releases 15*, document 3GPP TS 23.501, V15.2.0, 3GPP, Jun. 2018.
[16] J. Moy, *OSPF Version 2*, document STD 2328, Internet Requests for Comments, RFC Editor, Apr. 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2328.txt
[17] *Intermediate System to Intermediate System Intra-Domain Routeing Information Exchange Protocol for Use in Conjunction With the Protocol for Providing the Connectionless-Mode Network Service*, Standard 10589, International Organization for Standardization, 2002.
[18] B. Cohen. *The Bittorrent Protocol Specification*. Accessed: Oct. 2018. [Online]. Available: http://www.bittorrent.org/
[19] J. Rosenberg *et al.*, *SIP: Session Initiation Protocol*, document RFC 3261, RFC Editor, Internet Requests for Comments, Jun. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3261.txt
[20] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[21] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, Jan./Feb. 1997.
[22] 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.
[23] R. Johari and J. N. Tsitsiklis, "Efficiency loss in a network resource allocation game," *Math. Oper. Res.*, vol. 29, no. 3, pp. 407–435, 2004.
[24] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in *Proc. 8th USENIX Conf. Netw. Syst. Design Implement.*, Boston, MA, USA, Mar. 2011, pp. 1–24.
[25] *5G Network Slicing For Vertical Industries*, Ericsson, Stockholm, Sweden, Sep. 2017.
[26] Linux Foundation. *OPNFV*. [Online]. Available: https://www.opnfv.org
[27] ETSI. *Open Source Mano*. [Online]. Available: http://osm.etsi.org
[28] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE WCNC*, Istanbul, Turkey, Apr. 2014, pp. 2402–2407.
[29] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," *IEEE Access*, vol. 6, pp. 19691–19701, Apr. 2018, doi: 10.1109/ACCESS.2018.2822398.
[30] J. Liu, Y. Li, Y. Zhang, L. Su, and D. Jin, "Improve service chaining performance with optimized middlebox placement," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 560–573, Jul. 2017.
[31] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, Nov. 2016.

[32] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in *Proc. Netw. Oper. Manage. Symp. (NOMS)*, Krakow, Poland, May 2014, pp. 1–9.

[33] M. Bagaa, T. Taleb, A. Laghrissi, and A. Ksentini, "Efficient virtual evolved packet core deployment across multiple cloud domains," in *Proc. IEEE WCNC*, Barcelona, Spain, Apr. 2018, pp. 1–6.

[34] R. A. Addad, T. Taleb, M. Bagaa, D. Dutra, and H. Flinck, "Towards modeling cross-domain network slices for 5G," in *Proc. IEEE Globecom*, Abu Dhabi, UAE, Dec. 2018, pp. 1–7.

[35] A. Ksentini, M. Bagaa, and T. Taleb, "On using SDN in 5G: The controller placement problem," in *Proc. IEEE Globecom*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[36] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5G network infrastructure," in *Proc. IEEE ICC*, London, U.K., Jun. 2015, pp. 3879–3884.

[37] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards edge slicing: VNF Placement algorithms for a dynamic & realistic edge cloud environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[39] M. Grant and S. Boyd. (Mar. 2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: http://cvxr.com/cvx

[40] JSON. JavaScript Object Notation. [Online]. Available: https://www.json.org/

Hassan Halabian received the B.Sc. and M.Sc. degrees in electrical engineering from Isfahan University of Technology in 2005 and 2008, respectively, and the Ph.D. degree in electrical and computer engineering from Carleton University, Ottawa, in 2012. He was a MITACS Post-Doctoral Fellow with Carleton University from 2012 to 2013. From 2013 to 2016, he was a System Design Engineer with Ericsson Canada, Ottawa. In 2016, he joined the 5G Research Team, Huawei Canada Research Centre, where he is currently a Senior Research Engineer. His research interests include stochastic network optimization, optimal resource allocation, and network function virtualization and distributed systems. He was a recipient of a number of fellowships and awards, including the MITACS Elevate Strategic Post-Doctoral Fellowship, the Ontario Graduate Scholarship, the Queen Elizabeth II Scholarship in Science and Technology, the Ontario Graduate Scholarships in Science and Technology, and the Carleton Dean of Graduate Studies Academic Excellence Scholarship.