Distributed Resource Allocation for Network Slicing of Bandwidth and Computational Resource

Anqi Huang*, Yingyu Li*, Yong Xiao*, Xiaohu Ge*, Sumei Sun†, Han-Chieh Chao‡
*School of Electronic Information and Communications, Huazhong Univ. of Science & Technology, China
† Institute for Infocomm Research, Singapore
‡ School of Electrical Engineering, National Dong Hwa University, Taiwan

***Abstract*—Network slicing has been considered as one of the key enablers for 5G to support diversified services and application scenarios. This paper studies the distributed network slicing utilizing both the spectrum resource offered by communication network and computational resources of a coexisting fog computing network. We propose a novel distributed framework based on a new control plane entity, regional orchestrator, which can be deployed between base stations and fog nodes to coordinate and control their bandwidth and computational resources. We propose a distributed resource allocation algorithm based on Alternating Direction Method of Multipliers with Partial Variable Splitting (DistADMM-PVS). We prove that DistADMM-PVS minimizes the average latency of the entire network and at the same time guarantee satisfactory latency performance for every supported type of service. Simulation results show that DistADMM-PVS converges much faster than some other existing algorithms. In addition, the joint network slicing with both bandwidth and computational resources offers around 15% overall latency reduction compared to network slicing with only a single resource.**

***Index Terms*—Network slicing, resource allocation, distributed optimization, ADMM**

I. Introduction

It is commonly believed that 5G will be much more than a simple upgrade of physical performance metrics such as throughput and capacity [1] [2]. It will represent a fundamental transformation from the traditional data-oriented architecture towards a more flexible and service-oriented architecture. The *service-based architecture* (SBA) has been introduced by 3GPP as the key enabler for supporting a plethora of different services with diverse requirements on a common set of physical network resources. The core idea is to use software-defined networking (SDN) and network functions virtualization (NFV) to virtualize the network elements into network functions [3], each of which consists of a functional building block utilizing various resources offered by the network. Each type of services can then be instantiated by a series of network function sets, called *network slice* [4]. Network slicing has been considered as the foundation of 5G SBA to match with diversified service requirements and application scenarios.

To support emerging computationally intensive applications, create new business opportunities and increase revenues, fog computing has recently been promoted by both industry and standardization institutions as one of the key components in 5G [5]. Compared to massive-scale cloud data centers that are typically built in remote areas, fog computing consists of a large number of small computing servers, commonly referred to as fog nodes, that can offload computationally intensive tasks closer to end user equipments (UEs) [6]. Fog computing networks can be deployed by cloud service providers such as Amazon and Microsoft. It can also be implemented by mobile network operators (MNOs) within their network infrastructure.

Recently, network slicing utilizing both communication and computational resources has attracted significant interest. Allowing each slice to be supported by both resources can further improve the overall UE experience, balance resource utilization across different network elements, and open doorways for newly emerging services with stringent latency and computational requirements [7]. In spite of its great promise, allocating resources for multiple network slices with different resources introduces many new challenges. First, different resources are typically managed by different service providers. Therefore, exchanging and sharing proprietary information such as resource availability and traffic dynamics between them are generally impossible. Second, both fog computing and communication network infrastructure can be distributed in a wide geographical area and centralized coordination and management may result in intolerable coordination delay and excessive communication overhead. Finally, each UE can simultaneously request multiple types of services with different features offered by different resources. How to design an optimal algorithm that can quickly and accurately allocate various combination of different resources to support multiple network slices remains an open problem.

In this paper, we investigate the distributed network slicing for a 5G system consisting of a set of base stations (BSs) offering wireless communication services and a coexisting fog computing network performing computationally-intensive tasks. We consider joint resource allocation of both bandwidth of BSs and processing power of fog nodes for supporting multiple network slices. We focus on reducing the overall latency experienced by end UEs that include both communication delay in wireless links connecting UEs and BSs and queuing delay at fog nodes. A distributed resource allocation algorithm has been proposed. We prove that the proposed algorithm can minimize the average latency of the entire network and at the same time guarantee satisfactory performance for each supported type of service. The main contributions of this paper are summarized as follows:

1) A novel distributed framework based on regional orchestrator (RO) has been proposed for supporting distributed network slicing in a large network.
2) A distributed optimization algorithm based on Alternating

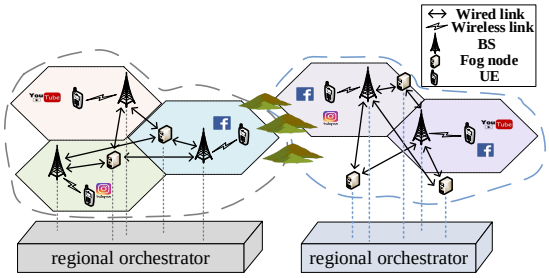

Fig. 1. Distributed network slicing model

Direction Method of Multipliers with Partial Variable Splitting (DistADMM-PVS) has been proposed to distributedly coordinate the resource allocation of both bandwidth of BSs and computational resources of fog nodes without requiring exchanging any proprietary information between BSs and fog nodes. We prove that the proposed algorithm can converge to the global optimal solution at a rate of $O(1/t)$.

3) Simulation and detailed performance analysis have been presented under various practical scenarios. Our result shows that joint slicing utilizing both bandwidth and computational resources offers around 15% overall latency reduction compared to network slicing with only a single resource.

The rest of the paper is organized as follows. Related works are reviewed in Section II. In Section III, we present the system model. The network slicing architecture and RO are introduced in Section IV. Distributed optimization algorithm is presented in Section V. We present the simulation result in Section VI and conclude the paper in Section VII.

II. Related Work

One of the main challenges for network slicing is how to quickly and effectively isolate and distribute resources according to the specific requirement of each service. In [8], the authors studied the allocation of the radio resources for network slicing. A prioritized admission mechanism was proposed to improve the resource utilization and increase UE's service experience. Network slicing has been studied for dynamic resource demand and availability in mobile environment in [9]. In [10], the authors proposed a network slicing architecture utilizing spectrum resources in both licensed and unlicensed bands.

Recently, network slicing utilizing computational resource has attracted significant interest [11] [12]. In [11], the authors proposed a distributed joint optimization algorithm for the allocation of fog computing resources and applied it to improve the performance of Internet-of-Things (IoT) system. In [12], the authors proposed a computational resource allocation scheme based on double-matching for fog computing networks.

III. System Model

We consider a network system consisting of a set $\mathcal{F} = \{1 \cdots F\}$ of F fog nodes and a set $\mathcal{S} = \{1 \cdots S\}$ of S BSs as illustrated in Figure 1. In a cellular network, each BS offers services in an exclusive coverage area. Suppose each UE can request at most N types of services. Let $\mathcal{N} = \{1 \cdots N\}$ be the set of all supported types of services. For each type of service, we assume that there exists a minimum volume of data, called task unit that can be transmitted by BSs and processed by fog nodes. For example, in video or audio processing service, each video or audio clip consists of a number of video or audio data units for transporting and processing. Let d_n be the data size of each task unit of service type n. Each BS s has been allocated with a fixed bandwidth, labeled as β_s for all $s \in \mathcal{S}$, and each fog node f can process at most μ_f task units per second for all $f \in \mathcal{F}$. Suppose the arrival rate k_{sn} of the nth service task unit at BS s follows a Poisson distribution, $k_{sn} \sim P(\lambda_{sn})$, where λ_{sn} is the expected number of received task units.

In this paper, we consider joint resource allocation for multiple network slices. Network slicing employs a network virtulization approach that virtualizes physical resources into Virtual Network Functions (VNFs). Each VNF can be further divided into smaller components to be placed in a common software container, so the network functionality can be quickly released and reused by different service instances. We refer to the smallest component that can be used in the VNFs for network slices as a slice unit. Each network slice can consist of many slice units. Different slice units are decoupled from each other. So each network slice can be launched and dynamically scaled without affecting other ongoing services.

In this paper, we focus on minimizing the service response time for each type of service, which can consist of both communication delay for task unit transportation from UEs to fog nodes as well as the queuing delay at fog nodes. Let us first consider the communication delay. Note that in many practical networks, BSs and fog nodes can be connected with wireline or fiber which typically offers much higher data rate than the wireless links between UEs and BSs. Therefore, in this paper, we follow a commonly adopted setting and ignore the communication delay between BSs and fog nodes. For a given bandwidth $0 \leq b_{sn} < \beta_s$ allocated by BS s for service type n, follow the commonly adopted setting, the communication delay for transporting each unit of task can be written as

$$p_{sn} = \frac{d_n}{b_{sn} \cdot \log(1 + h_{sn} \frac{w_{sn}}{\sigma_{sn}})}, \tag{1}$$

where w_{sn} is the required transmission power to transmit the task units for service type n from the UEs to BS s, σ_{sn} is the additive noise level received at BS s, and h_{sn} is the channel gain between BS s and the associated UE for service type n.

Queuing delay at the fog node can be affected by the processing power of fog nodes and task arrival rate. Suppose the maximum processing power allocated by fog nodes to process the nth type of service offered by BS s for its associated UEs is μ_{sn}. We follow a commonly adopted setting and assume the task units processed by fog nodes can be modeled as M/M/1 queuing [13]. We can write the queuing delay of the nth type of service offered to UEs in the coverage area of BS s as

$$q_{sn} = \frac{1}{\mu_{sn} - \lambda_{sn}}. \tag{2}$$

By combining (1) and (2), the overall service response time of the nth type of service offered by BS s can be written as

$$t_{sn} = p_{sn} + q_{sn}. \tag{3}$$

IV. Distributed Network Slicing Architecture

As mentioned earlier, two of the main challenges for quickly and accurately allocating resources across both communication and computing networks are:

1) Physical resources can be arbitrarily deployed over a large geographical area and therefore a centralized resource management and control architecture may result in intolerably high latency and excessive communication overhead.
2) Communication network infrastructure and fog computing networks can be owned by different service providers. Therefore, proprietory information cannot be shared between these two systems.

The above challenges cannot be addressed by simply extending the existing centralized SDN control plane frameworks such as OpenFlow [14] into a distributed setting. Actually, it has been observed by many existing works that OpenFlow-based SDN controller has been designed to mainly focus on managing the routing of data traffics and establishing and maintaining interconnections between virtual mesh networks [15]. It can be used to maintain network connection and service continuity even in mobile environment. It however cannot be applied to manage the computational resources of fog computing networks. Also, OpenFlow relies on a centralized SDN controller to manage network resources and can only establish static paths to each SDN switch.

In this paper, we proposed a distributed network slicing architecture based on a new control plane entity, RO, deployed between communication network and fog computing network to support the fine-grained control of resources across both network systems. In this architecture, the total coverage area has been divided into a set of sub-regions, each of which consists of a limited number of closely located BSs and fog nodes that can be connected with high-speed local wireline links. A RO can then be deployed in each sub-region to control a set of VNFs composed of locally available communication and computational resource units. Each RO can only control and instantiate network slicing with local VNFs within each sub-region. Each BS will query the RO with the resource requests whenever it receives a service task request from UEs. The RO will then coordinate with the service requesting BS and neighboring fog nodes to create the corresponding network slices. The RO will also supervise the path reservation and routing of the service traffic between BSs and fog nodes. Two or more ROs can coordinate with each other and jointly adjust the volume and distribution of their local VNFs if two or more neighboring sub-regions experience unbalanced traffic loads. Our proposed architecture is illustrated in Figure 1.

In 3GPP's network slicing framework, a certain amount of resource must be reserved and isolated for each supported type of service, so there always exist available resources whenever a service request has been received. In this paper, we consider the RO implemented in 3GPP's framework [16] [17]. For a limited time duration, each RO must first reserve a certain amount of computational resources at fog nodes, which is given by λ_{sn}, and a limited bandwidth, which is given by b_0 for task units of service type n at BS s. During this time, the reserved resources will be utilized to support the requested service instances, so we have the following constraints:

$$b_{sn} > b_0, \tag{4a}$$

$$\mu_{sn} > \lambda_{sn}. \tag{4b}$$

In this paper, we focus on the resource allocation and network slicing within a specific time duration in which the maximum bandwidth of a set of local BSs and a given amount of processing power of local fog nodes have been reserved for a set of supported types of services. The dynamic resources allocation and network slicing will be left for our future research. We consider the following constraints:

1) *Bandwidth constraint*: Let β_s be the total bandwidth reserved by each BS s. In other words, the total bandwidth that can be allocated by BS s to all upcoming service tasks cannot exceed β_s. Generally speaking, the RO needs to reserve sufficient resources without knowing the exact number of task units which will be arrived in the future. It is however possible for the RO to estimate the possible number of arrival task units according to the empirical probability distribution of the task arrival rate. In this way, the RO can reserve sufficient resource to support the performance-guaranteed services for the majority of possible tasks with a certain level of confidence. More specifically, we define the confidence level $\theta = \Pr(k_{sn} \leq \theta_{sn})$ as the possibility that the number of type n service task units arrived at BS s is below a certain threshold number θ_{sn}. For example, $\theta = 0.9$ means that the RO wants to reserve resources to meet the demands of all UEs with 90% confidence. We can observe that θ is equivalent to the Cumulative Distribution Function (CDF) of task arrival rate k_{sn}. We can therefore write $\theta_{sn} = CDF_k^{-1}(\theta, \lambda_{sn})$ where $(\cdot)^{-1}$ is the inverse function. We can then have the following constraint for the bandwidth allocated by BS s to a set $\mathcal{N}$ of all supported types of services

$$\sum_{n \in \mathcal{N}} \theta_{sn} \cdot b_{sn} \leq \beta_s. \tag{5}$$

2) *Computational resource constraint*: Suppose the total computational resource γ reserved to all fog nodes in a sub-region is limited. The sum of the computational resources allocated to all the services cannot exceed γ. We then have the following computational resources constraint

$$\sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} \mu_{sn} \leq \gamma. \tag{6}$$

In addition, we assume each supported type of service n has a maximum tolerable latency, labeled as $\overline{T_n}$, i.e., we have $t_{sn} \leq \overline{T_n}, \forall s \in \mathcal{S}, n \in \mathcal{N}$.

Our proposed architecture is general and flexible. It can be applied to network slicing utilizing multiple resources across a wide geographical area. In this paper, we consider network slicing architecture to leverage the RO. BSs and fog nodes can coordinate with the RO to dynamically allocate the bandwidth and the computational resources. In summary, we focus on designing a distributed algorithm to optimize the following

problem

$$\min_{\{\boldsymbol{b}\}\{\boldsymbol{\mu}\}} \sum_{s\in\mathcal{S}}\sum_{n\in\mathcal{N}} t_{sn} \tag{7a}$$

$$\text{s.t.} \quad b_{sn} > b_0, \forall s \in \mathcal{S}, n \in \mathcal{N}, \tag{7b}$$

$$\mu_{sn} > \lambda_{sn}, \forall s \in \mathcal{S}, n \in \mathcal{N}, \tag{7c}$$

$$t_{sn} \leq \overline{T_n}, \forall s \in \mathcal{S}, n \in \mathcal{N}, \tag{7d}$$

$$\sum_{n\in\mathcal{N}} \theta_{sn} \cdot b_{sn} \leq \beta_s, \forall s \in \mathcal{S}, n \in \mathcal{N}, \tag{7e}$$

$$\sum_{s\in\mathcal{S}}\sum_{n\in\mathcal{N}} \mu_{sn} \leq \gamma, \forall s \in \mathcal{S}, n \in \mathcal{N}. \tag{7f}$$

where $\boldsymbol{b} = \{\ldots, b_{sn}, \ldots\}$, and $\boldsymbol{\mu} = \{\ldots, \mu_{sn}, \ldots\}$.

V. Distributed Optimization for Joint Network Slicing

As mentioned before, in order to minimize the overall latency experienced by end UEs, we need to carefully decide the resources allocated to each network slice. However, solving problem (7) involves jointly deciding the proper amount of bandwidth and processing power for every type of services with global information such as the expected number of arrived task units and the computational capacity of every fog node which may result in intolerably high latency and communication overhead.

To address the above problems, we need to develop a distributed optimization algorithm for solving the joint network slicing problem in (7) with the following design objectives:

1) *Distributed Optimization with Coordination:* The proposed optimization algorithm should be able to separate the global optimization problem into a set of sub-problems, each of which can be solved by a BS according to its local information. The RO can then be used to coordinate the solution of these subproblems to achieve the globally optimal resource allocation solution.
2) *Privacy Preservation:* BSs and fog nodes may not want to share their private information such as bandwidth, expected number of arrived task units, and computational capacities with each other.
3) *Fast Convergence:* BSs and fog nodes connected to each RO can change over the time. Thus, the algorithm that needs to quickly converge to the global optimal solution.

We propose a distributed optimization algorithm based on Alternating Direction Method of Multipliers (ADMM) [18]. Compared to traditional convex optimization algorithms, ADMM is more suitable for solving inequality constrained optimization problems in a decentralized manner. Furthermore, the decomposition-coordination procedure of the ADMM makes it possible to protect the aforementioned private information of BSs and fog nodes. Unfortunately, normal ADMM approaches can only handle problems with two blocks of variables [19]. In order to solve problem (7) with the above objectives 1)-3), we propose a distributed ADMM algorithm with Partial Variable Splitting referred to as DistADMM-PVS. In this algorithm, the Lagrangian dual problem of (7) will be divided to S sub-problems, each of which can be solved by an individual BS using its local information. The RO will collect the intermediate results from BSs and send the coordination feedbacks.

Let us first follow the same line as [19] and combine constraints in (7) with the objective function by introducing a set of $S+1$ indicator functions. Specifically, for constraints (7b)-(7e) that can be separated across different BSs, we define $\mathcal{G}_s = \{\boldsymbol{b}_s, \boldsymbol{\mu}_s : b_{sn} > b_0, \mu_{sn} > \lambda_{sn}, t_{sn} \leq \overline{T_n}, \sum_{n\in\mathcal{N}} \theta_{sn} \cdot b_{sn} \leq \beta_s, \forall s \in \mathcal{S}, n \in \mathcal{N}\}$ as the feasible set corresponding to BS s where $\boldsymbol{b}_s = \langle b_{sn}\rangle_{n\in\mathcal{N}}$ is the vector of bandwidth allocated by BS s for each type of services and $\boldsymbol{\mu}_s = \langle \mu_{sn}\rangle_{n\in\mathcal{N}}$ is the vector of processing power allocated for each type of services connected to BS s. Let $\boldsymbol{x}_s = \langle \boldsymbol{b}_s, \boldsymbol{\mu}_s\rangle, \forall s \in \mathcal{S}$, we define S indicator functions as

$$\mathbf{I}_{\mathcal{G}_s}(\boldsymbol{x}_s) = \begin{cases} 0, & \boldsymbol{x}_s \in \mathcal{G}_s, \\ +\infty, & \boldsymbol{x}_s \notin \mathcal{G}_s. \end{cases} \quad \forall s \in \mathcal{S} \tag{8}$$

For constraint (7f) that cannot be separated, we can also define an indicator function $\mathbf{I}_{\mathcal{G}}(\boldsymbol{\mu})$ as

$$\mathbf{I}_{\mathcal{G}}(\boldsymbol{\mu}) = \begin{cases} 0, & \boldsymbol{\mu} \in \mathcal{G}, \\ +\infty, & \boldsymbol{\mu} \notin \mathcal{G}, \end{cases} \tag{9}$$

where $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_S]$, and $\mathcal{G}$ is the half space defined by $\mathcal{G} = \{\boldsymbol{\mu} : \sum_{s\in\mathcal{S}}\sum_{n\in\mathcal{N}} \mu_{sn} \leq \gamma\}$.

By including the above indicator functions $\mathbf{I}_{\mathcal{G}_s}$ and $\mathbf{I}_{\mathcal{G}}$, the original joint network slicing problem (7) with a set of inequality constraints can be converted to the following form without inequality constraints

$$\min_{\{\boldsymbol{x}_s, \boldsymbol{z}_s\}} \quad \sum_{s\in\mathcal{S}} \{f(\boldsymbol{x}_s) + \mathbf{I}_{\mathcal{G}_s}(\boldsymbol{x}_s)\} + \mathbf{I}_{\mathcal{G}}(\boldsymbol{z}) \tag{10a}$$

$$\text{s.t.} \quad \boldsymbol{x}_s = \boldsymbol{z}_s, \forall s \in \mathcal{S}, \tag{10b}$$

where $f(\boldsymbol{x}_s) = \sum_{n\in\mathcal{N}} t_{sn}$, and $\boldsymbol{z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_S]$ is the introduced auxiliary variable.

The augmented Lagrangian of problem (10) is given by

$$\begin{aligned} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Lambda}) &= \sum_{s\in\mathcal{S}} \{f(\boldsymbol{x}_s) + \mathbf{I}_{\mathcal{G}_s}(\boldsymbol{x}_s)\} + \mathbf{I}_{\mathcal{G}}(\boldsymbol{z}) \\ &+ \boldsymbol{\Lambda}^T(\boldsymbol{x} - \boldsymbol{z}) + \frac{\rho}{2}||\boldsymbol{x} - \boldsymbol{z}||_2^2, \end{aligned} \tag{11}$$

where $\boldsymbol{\Lambda}$ is the dual variable and ρ is the augmented Lagrangian parameter.

We can then prove the following result.

Theorem 1: The augmented Lagrangian of problem (10) specified in (11) is convex and partially separable among $\boldsymbol{x}_s$.

Proof: For the convexity, it can be directly shown that set $\mathcal{G}_s, \forall s \in \mathcal{S}$ and halfspace $\mathcal{G}$ are all convex sets, and their intersection which is the feasible set of problem (10) is also a convex set. We can also show that within the feasible set of problem (10), the second derivative of $f(\boldsymbol{x}_s)$ is always positive which means that it is convex. Due to the fact that summation preserves convexity, we can then prove that $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Lambda})$ is convex.

To prove (11) is partially separable, we rewrite the augmented Lagrangian in (11) as follows

$$\begin{aligned} \mathcal{L}_\rho(\boldsymbol{x}_1, ..., \boldsymbol{x}_S, \boldsymbol{z}, \boldsymbol{\Lambda}) = \sum_{s\in\mathcal{S}} \{&f(\boldsymbol{x}_s) + \mathbf{I}_{\mathcal{G}_s}(\boldsymbol{x}_s) \\ &+ \boldsymbol{\Lambda}_s^T(\boldsymbol{x}_s - \boldsymbol{z}_s) + \frac{\rho}{2}||\boldsymbol{x}_s - \boldsymbol{z}_s||_2^2 \} + \mathbf{I}_{\mathcal{G}}(\boldsymbol{z}), \end{aligned} \tag{12}$$

From (12), we can observe that $\mathcal{L}_\rho(\boldsymbol{x}_1, ..., \boldsymbol{x}_S, \boldsymbol{z}, \boldsymbol{\Lambda})$ can be partially separated across $\boldsymbol{x}_s$ for $s \in \mathcal{S}$. This concludes the proof. ■

We can then convert problem (10) into two-block ADMM form as follows

$$\boldsymbol{x}^{k+1} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{s\in\mathcal{S}} \{f(\boldsymbol{x}_s) + \mathbf{I}_{\mathcal{G}_s}(\boldsymbol{x}_s)\} + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}^k + \boldsymbol{\Lambda}^k\|_2^2, \quad (13a)$$

$$\boldsymbol{z}^{k+1} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \mathbf{I}_{\mathcal{G}}(\boldsymbol{z}) + \frac{\rho}{\mathbf{2}}\|\boldsymbol{x}^{\mathbf{k+1}} - \boldsymbol{z} + \boldsymbol{\Lambda}^{\mathbf{k}}\|_{\mathbf{2}}^{\mathbf{2}}, \quad (13b)$$

$$\boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k + \boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}, \quad (13c)$$

where k denotes the number of iterations. According to the partially separability of $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Lambda})$, we can divide (13a) into a set of sub-problems, each of which can be solved by a BS using its local information. In particular, each BS s solves the following sub-problem

$$\boldsymbol{x}_s^{k+1} = f(\boldsymbol{x}_s) + \mathbf{I}_{\mathcal{G}_s}(\boldsymbol{x}_s) + \frac{\rho}{2}\|\boldsymbol{x}_s - \boldsymbol{z}_s^k + \boldsymbol{\Lambda}_s^k\|_2^2, \quad \forall s \in \mathcal{S}. \quad (14)$$

Meanwhile, (13b) is equivalent to projecting the point $\boldsymbol{x}^{k+1} + \boldsymbol{\Lambda}^k$ onto the halfspace $\mathcal{G}$, i.e.

$$\boldsymbol{z}^{k+1} = \boldsymbol{\Pi}_{\mathcal{G}}(\boldsymbol{x}^{k+1} + \boldsymbol{\Lambda}^k), \quad (15)$$

where $\boldsymbol{\Pi}_{\mathcal{G}}(\cdot)$ denotes the projection onto halfspace $\mathcal{G}$.

Detailed description of our proposed algorithm is presented in Algorithm 1.

Theorem 2: The proposed DistADMM-PVS algorithm converges to the global optimal solution of the joint network slicing problem in (7) at a rate of $O(1/t)$.

Proof: The convergence property of our proposed DistADMM-PVS algorithm directly follows that of the standard ADMM approach [18], [19]. Here we omit the details due to the limit of space. ■

Algorithm 1 Distributed ADMM with Partial Variable Splitting (DistADMM-PVS)

Initialization: Each BS s chooses an initial variable $\boldsymbol{x}_s^0$ and the RO chooses an initial dual variable $\boldsymbol{\Lambda}^0$; $k = 1$
Set the maximum number of iterations as $K > 0$
while $k \leq K$ **do**
1. Each BS s simultaneously do:
   1) Update $\boldsymbol{x}_s^{k+1}$ according to (14) and report it to the RO;
   2) Allocate its bandwidth for each type of arrived task according to $\boldsymbol{b}_s^{k+1}$;
2. After all the $\boldsymbol{x}_s^{k+1}$ are received, the RO do:
   1) Update auxiliary variable $\boldsymbol{z}^{k+1}$ according to (15);
   2) Update dual variable $\boldsymbol{\Lambda}^{k+1}$ according to (13c);
   3) **if** Stopping criteria met
      break;
   end if
   4) Sends the sub-vectors $\boldsymbol{z}_s^{k+1}$ and $\boldsymbol{\Lambda}_s^{k+1}$ to the corresponding BS s;
3. $k = k + 1$
end while

VI. Simulation Results And Comparative Analysis

To evaluate the performance of our proposed network slicing architecture, we simulate a network system consisting of 285 BSs and 285 fog nodes deployed in Dublin which can support 3 types of services: text, audio, and video process service, as shown in Figure 2. To prove the applicability of our proposed distributed network slicing architecture, we simulate 3 areas from city centers to suburbs as shown in Figure 3, we assume each BS has reserved 30MHz bandwidth, and each fog node can process at most 180 task units per second. We assume BSs and fog nodes in the same area have reserved same bandwidth and computational resources. We compare the joint slicing involving both bandwidth of BSs and processing powers of fog nodes with two other network slicing scenarios, each of which only utilizes one type of resource, referred to as bandwidth slicing and computational resource slicing, respectively.

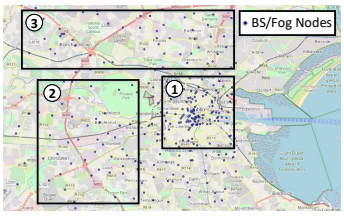

Fig. 2. Distribution of BSs

Area number	Total number of BSs	Average coverage area (km^2)
1	40	9.41
2	40	45.63
3	40	161.82

Fig. 3. Average Coverage Area of BSs

We first evaluate the convergence performance of Algorithm 1. In Figure 4, we compare the interior-point algorithm with our proposed algorithm 1 (DistADMM-PVS) under different number of iterations. The interior-point algorithm has been widely applied in communication network systems [20]. We can observe that our proposed Algorithm 1 can converge to a close neighborhood of the minimum latency within the first few iterations. It can offer much faster convergence performance compared to the interior-point algorithm.

We compare three architectures in different kinds of areas, as shown in Figure 5. We can observe that our proposed joint slicing architecture performs better than other network slicing architecture in all three kinds of areas, which proves that our architecture has significant regional applicability, and can apply to different areas. We have the same characteristics for three architectures in all areas, so we only cover simulation results of area 1 in detail, as shown in Figure 6, Figure 7 and Figure 8.

In Figure 6, we fix the processing power reserved for each fog node and the value of θ to evaluate the service response time under different bandwidth reserved for each BS. We can observe that the service response time decreases with the total reserved bandwidth. We can also observe that when the bandwidth of BSs is limited, the bandwidth slicing offers a better performance than computational resource slicing. However, as the bandwidth of each BS increases, the service response time of computational resource slicing starts to decrease much faster than that of the bandwidth slicing. This is because in our simulation, we fix the computational resource. In this case, when each BS has limited bandwidth, the communication latency dominates the overall latency. Therefore, applying bandwidth slicing to reduce the communication latency can have a higher impact than optimizing the computational resources in fog nodes for reducing the overall service response time. When the bandwidth of each BS becomes sufficient, the queuing delay will dominate overall latency. In this case, the computational resource slicing will become more useful to

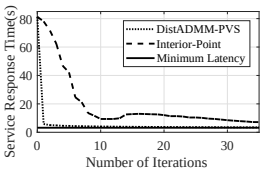

Fig. 4. Comparison of Interior-Point and Algorithm 1

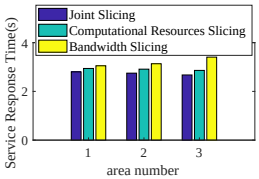

Fig. 5. Comparison of Three Architectures in Different Areas

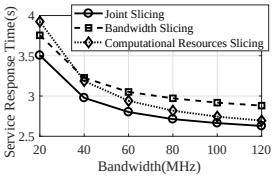

Fig. 6. Comparison of Three Architectures under Different Bandwidth of Each BS of Area1

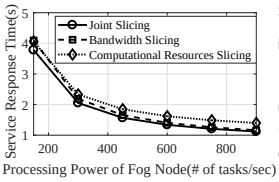

Fig. 7. Comparison of Three Architectures under Different Processing Rate of Each Fog Node of Area1

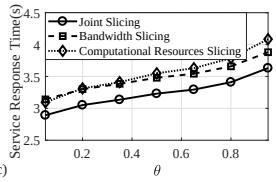

Fig. 8. Comparison of Three Architectures under Different θ of Area1

reduce the service response time.

In Figure 7, we fix the bandwidth reserved for each BS and value of θ to compare the service response time of three network slicing scenarios under different processing power reserved for each fog node. We can observe that the service response time decreases with the processing power reserved for each fog node. Similarly, we observe that when the computational resource of fog nodes is limited, the computational resource slicing offers a better performance than bandwidth slicing. However, as the computational resource of each fog node becomes sufficient, bandwidth slicing starts to decrease faster than the computational resource slicing. This is because in Figure 7, the bandwidth has been fixed. When each fog node has limited processing power, the queuing delay dominates the overall latency. When the processing power of each fog node increases, the communication latency starts to dominate the overall latency.

Similarly, in Figure 8, we fix the processing power reserved for each fog node and the bandwidth reserved for each BS to compare the service response time under different values of θ. We observe that the service response time increases with θ. This is because when θ increases, the total number of task units that need to be transported by BSs becomes larger. This will cause a higher communication delay, resulting in a higher service response time. We also observe that as θ increases, the service response time of bandwidth slicing starts to increase much slower than that of the computational resource slicing. This is because we fix both processing power reserved for each fog node and the bandwidth reserved for each BS. In this case, the number of task units for each service from each BS increases with θ. When θ is small, bandwidth allocated to each task unit is sufficient and the queuing delay dominates the overall latency. When θ becomes larger, bandwidth allocated to each task unit is limited and the communication latency will start to dominate the overall latency.

VII. Conclusion

In this paper, we have investigated the distributed network slicing for a 5G system consisting of a set of BSs offering wireless communication services coexisting with a fog computing network performing computationally-intensive tasks. We propose a novel distributed framework based on a new control plane entity, regional orchestrator (RO), which can be deployed between BSs and fog nodes to coordinate and control their bandwidth and computational resources. We have also proposed a distributed resource allocation algorithm to distributedly coordinate the resource allocation of bandwidth of BSs and computational resources of fog nodes. Our simulation results show that the proposed algorithm can approach the global optimal solution with fast convergence rate.

Acknowledgment

The authors would like to acknowledge the support from National Key R&D Program of China (2017YFE0121600).

References

[1] NGMN Alliance, "5G white paper," Feb. 2015.
[2] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, Feb. 2016.
[3] ETSI, "Network functions virtualisation (NFV); ecosystem; report on sdn usage in NFV architectural framework," v. 1.1.1, Dec. 2015.
[4] NGMN Alliance, "Description of network slicing concept," Sep. 2016.
[5] M. Chiang, B. Balasubramanian, and F. Bonomi, *Fog for 5G and IoT*. John Wiley & Sons, New Jersey, 2017.
[6] X. Ge, Y. Sun, H. Gharavi, and J. Thompson, "Joint optimization of computation and communication power in multi-user massive mimo systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4051–4063, Jun. 2018.
[7] X. Ge, B. Yang, J. Ye, G. Mao, C.-X. Wang, and T. Han, "Spatial spectrum and energy efficiency of random cellular networks," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 1019–1030, Mar. 2015.
[8] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5g mobile systems," in *European Wireless Conference*, Oulu, Finland, May 2016, pp. 1–6.
[9] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. Leung, "Network slicing based 5g and future mobile networks: mobility, resource management, and challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, Aug. 2017.
[10] Y. Xiao, M. Hirzallah, and M. Krunz, "Distributed resource allocation for network slicing over licensed and unlicensed bands," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2260–2274, Oct. 2018.
[11] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.
[12] B. Jia, H. Hu, Y. Zeng, T. Xu, and Y. Yang, "Double-matching resource allocation strategy in fog computing networks based on cost efficiency," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 237–246, Jun. 2018.
[13] N. M. Edelson, D. K. Hildebrand *et al.*, "Congestion tolls for poisson queuing processes," *Econometrica*, vol. 43, no. 1, pp. 81–92, Jan. 1975.
[14] ONF, "SDN architecture issue 1.1," ONF TR-521, 2016.
[15] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for OpenFlow," in *Internet Network Management Workshop*, San Jose, CA, Apr. 2010, pp. 1–6.
[16] 3GPP, "Telecommunication management;study on management and orchestration of network slicing for next generation network," 3GPP TR 28.801.
[17] ——, "Management and orchestration; provisioning," 3GPP TS 28.531, Dec. 2017.
[18] Y. Li, G. Shi, W. Yin, L. Liu, and Z. Han, "A distributed admm approach with decomposition-coordination for mobile data offloading," *IEEE Trans Veh. Tech.*, vol. 67, no. 3, pp. 2514–2530, Oct. 2017.
[19] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, Jul. 2011.
[20] H. Huang and A. V. Savkin, "An algorithm of efficient proactive placement of autonomous drones for maximum coverage in cellular networks," *IEEE Wireless Commun. Letters*, vol. 7, no. 6, Dec. 2018.