Dynamic Network Slicing for Scalable Fog Computing Systems with Energy Harvesting

Yong Xiao, *Senior Member, IEEE* and Marwan Krunz, *Fellow, IEEE*

Abstract

This paper studies fog computing systems, in which cloud data centers can be supplemented by a large number of fog nodes deployed in a wide geographical area. Each node relies on harvested energy from the surrounding environment to provide computational services to local users. We propose the concept of *dynamic network slicing*, a software-Defined Networking (SDN)-based concept in which a regional orchestrator coordinates workload distribution among local fog nodes, providing partitions/slices of energy and computational resources to support a specific type of service with certain quality-of-service (QoS) guarantees. The resources allocated to each slice can be dynamically adjusted according to service demands and energy availability. A stochastic overlapping coalition-formation game is developed to investigate distributed cooperation and joint network slicing between fog nodes under randomly fluctuating energy harvesting and workload arrival processes. We observe that the overall processing capacity of the fog computing network can be increased by allowing fog nodes to maintain a belief function about the unknown state and private information of other nodes. An algorithm based on a belief-state partially observable Markov decision process (B-POMDP) is proposed to achieve the optimal resource slicing structure among all fog nodes. We describe how to implement our proposed dynamic network slicing within the 3GPP network sharing architecture, and then we simulate a possible implementation of fog computing using data from a real cellular network with 400 BSs deployed in the city of Dublin. Our numerical results show that our framework can significantly improve the workload processing capability of fog computing networks. In particular, even when each fog node can coordinate only with its closest neighbor, the total amount of workload processed by fog nodes can be almost doubled under certain scenarios.

Index Terms

Fog computing, software-defined networking, energy harvesting, network virtualization, network slicing, game theory, coalition formation.

I. INTRODUCTION

With the widespread proliferation of intelligent systems, Internet of Things (IoT) devices, and smart infrastructures, computation-intensive mobile applications that require low delay and fast processing time are becoming quite popular. Next-generation mobile networks (e.g., 5G and

Y. Xiao and M. Krunz are with the Department of Electrical and Computer Engineering at the University of Arizona, Tucson, AZ (e-mails: yongxiao and krunz@email.arizona.edu).

beyond) are expected to serve over 50 billion mobile devices, most of which are smart devices requiring as low as 1 millisecond latency and very little energy consumption. Major IT service providers, such as Google, Yahoo, Amazon, etc., are heavily investing in large-scale data centers to meet the demand for future data services. However, these data centers are expensive and often built in remote areas to save costs. This makes it difficult to provide the quality-of-service (QoS) requirements of end users, especially for users located at the edge of a coverage area. To provide low-latency services to end users, a new framework referred to as *fog computing* has emerged [1], in which a large number of wired/wireless, closely located, and often decentralised devices, commonly referred to as *fog nodes*, can communicate and potentially cooperate with each other to perform certain computational tasks. Fog computing complements existing cloud services by distributing computation, communication, and control tasks closer to end users. Fog nodes include a variety of devices between end users and data centers, such as routers, smart gateways, access points (APs), base stations (BSs), and set-top boxes. According to Next-Generation Mobile Network (NGMN) Alliance [2], fog computing will be an important component of 5G systems, providing support for computation-intensive applications that require low latency, high reliability, and secure services. Examples of these applications include intelligent transportation, smart infrastructure, e-healthcare, and augmented/virtual reality (AR/VR). The success of fog computing heavily relies on the ubiquity and intelligence of low-cost fog nodes to reduce the latency and relieve network congestion [3]–[5].

Over the last decade, there has been a significant interest in climate change and energy sustainability for information and communication technologies. The telecommunication network infrastructure is already one of the leading sources of global carbon dioxide emissions [6]. In addition, the unavailability of a reliable energy supply from electricity grids in some areas is forcing mobile network operators (MNOs) to use sources like diesel generators for power, which not only increase operating costs but also contribute to pollution. Energy harvesting is a technology that allows electronic devices to be powered by the energy converted from the environment, such as sunlight, wind power, and tides. It has recently attracted significant interest due to its potential to provide a sustainable energy source for electronic devices with zero carbon emission [7]–[10]. Allowing fog nodes to utilize the energy harvested from Nature can provide ubiquitous computational resources anywhere at any time. For example, fog nodes deployed inside an edge network can rely on renewable energy sources to support low-latency, real-time computation for applications such as environmental control, traffic monitoring and congestion avoidance, automated real-time vehicle guidance systems, and AR/VR assisted manufacturing.

Incorporating energy harvesting into the design of the fog computing infrastructure is still relatively unexplored. In contrast to data centers that can be supported by massive photovoltaic solar panels or wind turbines, fog nodes are often limited in size and location. Unlike a data center, it is generally difficult to have a global resource manager that coordinates resource distribution among fog nodes in a centralized fashion. Developing a simple and effective method for fog nodes to optimize their energy and computational resources, enabling autonomous resource management according to the time-varying energy availability and user demands, is still an open problem.

Enabled by software-defined networking (SDN) and network function virilization (NFV) technologies, the concept of *network slicing* has recently been introduced by 3GPP to further improve the flexibility and scalability of fog computing for 5G systems [11]–[14]. Network slicing allows a fog node to support multiple types of service (use cases) by partitioning its resources, such as spectrum, infrastructure, network functionality, and computing power among these types. Resource partitions, commonly referred to as *slices*, can be tailored and orchestrated according to different QoS requirements of different service types (e.g., real-time audio, image/video processing with various levels of delay tolerance, etc.) [15]. Multiple SDN-based network slicing architectures have been proposed by 3GPP [16], NGMN Alliance [2], [14], and Open Networking Foundation (ONF) [17], [18]. However, these architectures are all based on a centralized control plane and cannot be directly applied to large-scale network systems.

In this paper, we introduce a new *dynamic network slicing* architecture for large-scale energy-harvesting fog computing networks. This architecture embodies a new network entity, the *regional SDN-based orchestrator*, that coordinates the workload processed by multiple closely located fog nodes and creates slices of energy and computing resources for various types of service requested by end users To minimize the coordination cost, the workload of each user is first sent to the closest fog node. Fog nodes will then make autonomous decisions on how much energy resource is to be spent on activating computational resources and how to partition the activated computational resources according to time-varying energy availability, user demands, and QoS requirements. If a fog node decides that it needs help from its neighboring fog nodes to process a part of its received workload, or if it has surplus resource to help other fog nodes in proximity, it will coordinate with these fog nodes though the regional SDN-based orchestrator. Our main objective is to develop a simple distributed network slicing policy that can maximize the utilization efficiency of available resources and balance the workloads among fog nodes over a wide geographical area. The distributed and autonomous decision making process at each fog node makes game theory a suitable tool to analyze the interactions among fog nodes. In this paper, we develop a stochastic overlapping coalition-formation game-based framework, called *dynamic* *network slicing game*, to analyze such interactions. In contrast to the traditional partition-based coalition formation game, in our game, players are allowed to interact with each other across multiple coalitions, which has the potential to further improve the resource utilization efficiency and increase the outcome for players. Unfortunately, finding a stable coalitional structure in this game is known to be notoriously difficult. Because each player can allocate a fraction of its resources to each coalition, there can be infinitely many possible coalitions among players. It has already been proved that an overlapping coalition game may not always have a stable coalitional structure. Even it does, there is no general method that can converge to such a structure. We propose a distributed algorithm based on a belief-state partially observable Markov decision process (B-POMDP) for each fog node to sequentially learn from its past experience and update its belief function about the state and offloading capabilities of other nodes. We prove that our proposed algorithm can achieve the optimal resource slicing policy without requiring back-and-forth communication between fog nodes. Finally, we evaluate the performance of our proposed framework by simulations, using actual base station topological deployment of a large-scale cellular network in the city of Dublin. Results show that our proposed framework can significantly improve the workload offloading capability of fog nodes. In particular, even when each fog node can only cooperate with its closest neighbor, the total amount of workload offloaded by the fog layer can almost be doubled especially for densely deployed fog nodes in urban areas.

The remainder of this paper is organized as follows. Related works on resource allocation for fog computing networks are reviewed in Section II. In Section III, we briefly review existing network slicing architectures proposed by 3GPP and ONF and then propose our dynamic network slicing architecture. We introduce the system model and formulate the optimization problem in Section IV. Dynamic network slicing game is introduced in Section V. We discuss how to implement the proposed dynamic network slicing within the existing 3GPP network sharing architecture and present numerical results in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work

A key challenge for fog computing is to provide QoS-guaranteed computational services to end users while optimizing the utilization of local resources owned by fog nodes. In [19], the joint optimization of allocated resources while minimizing the carbon footprint was studied for video streaming services over fog nodes. A proximal algorithm was proposed to decompose the large-scale global optimization problem into subproblems that are easier to solve. A service-oriented resource estimation and management model was proposed in [20] for fog computing systems consisting of multiple fog nodes.

Fog computing has recently been extended to IoT systems, whose low-cost and resource-limited devices can be assisted by fog nodes in making decisions and analyzing data (e.g., traffic congestion detection and environmental monitoring.) [21]–[23]. In [21], the authors evaluated the suitability of fog computing within the framework of IoT and analyzed it for latency-sensitive IoT applications running at the network-edge. Several workload offloading problems were studied for cloud computing networks with centralized infrastructure support [24]–[27]. More specifically, in [24], the authors proposed a framework called ThinkAir, which exploits the concept of smartphone virtualization in the cloud to provide method-level computation offloading at fog nodes. The authors in [26] developed a system called MAUI, which enables fine-grained energy-aware offload of mobile code to the edge infrastructure.

Game theory has been shown to be a promising tool to analyze the performance and optimize fog computing networks. Specifically, in [28], a hierarchical game-based model was applied to analyze the interactions between cloud data centers (CDCs) and fog nodes. An optimal pricing mechanism was proposed for CDCs to control resource utilization at fog nodes. The authors in [29] proposed an auction mechanism to study optimal resource allocation for mobile cloud computing (MCC) systems. In [30], a cooperative game theoretic model was established to study MCC systems in which multiple service providers share their resources by combining their radio and computational resources into a common resource pool.

To the best of our knowledge, our paper is the first work to study the distributed workload offloading problem for renewable-energy-supported fog computing networks from the game theoretic perspective.

III. SDN-based Dynamic Network Slicing Architecture

In this section, we first briefly describe existing network slicing architectures introduced by ONF and 3GPP. We then propose our SDN-based dynamic network slicing architecture for supporting a more flexible and scalable deployment of fog computing systems.

A. Existing Network Slicing Architectures

A generic fog computing architecture consists of the following elements [31]–[34]:

1) *Cloud Computing Service Provider (CSP)* The CSP owns and manages large-scale CDCs that can provide sufficient hardware and software resources with low processing latency. CDCs are often built in low-cost remote areas and therefore services processed at the cloud are expected to experience high transmission latency.

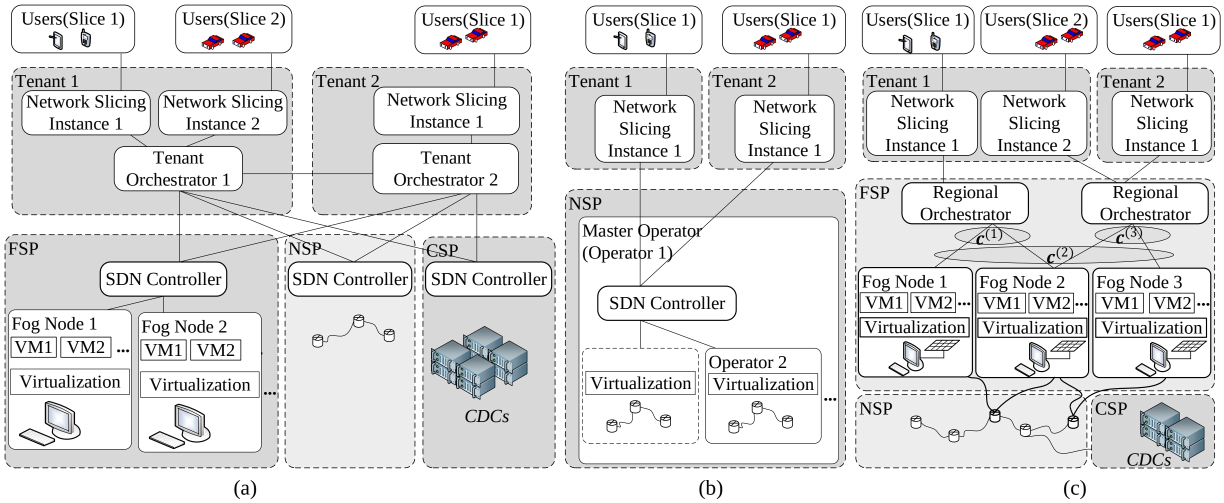


Fig. 1. Comparison between our proposed architecture and existing ones: (a) ONF SDN network slicing architecture, (b) 5G network slice broker, and (c) our proposed network slicing architecture based on regional SDN-based orchestrator.

2) *Fog Computing Service Provider (FSP)* The FSP controls a large number of low-cost fog nodes (e.g., mini-servers), deployed in a wide geographical area. Typically, fog nodes do not have high-performance processing units. However, they are much cheaper to deploy and require much less energy to operate. In this paper, we focus on energy-harvesting fog computing networks in which the computing resources that can be activated and offered to serve users are time-varying and depend on variations in harvested energy.
3) *Networking Service Provider (NSP)* The NSP deploys a large wired or wireless network infrastructure that connects users to fog nodes or remote CDCs.
4) *Tenants*: can request resources (e.g., slices) from one or more service providers to serve users' computing needs.
5) *Users*: consume the services offered by tenants. Users can be located in a wide geographical area and can request different types of services with different QoS requirements.

Note that the above elements may not always be physically separated from one another. For example, a cellular network operator may deploy mini-servers at some sites/locations to offer fog computing services to the tenants within its service coverage area [35]. In this case, the FSP and NSP correspond to the same entity (i.e., cellular network operator).

A comprehensive SDN architecture was introduced by ONF in [18]. In this architecture, an intermediate control plane is used to deliver tailored services to users in the application plane by configuring and abstracting the physical resources. The proposed SDN architecture can naturally support network slicing [17]. In particular, the SDN controller is supposed to collect all the information needed to communicate with each user and create a complete abstract set of resources (as resource groups) to support control logic that constitutes a slice, including the complete collection of related service attributes of users. Although ONF's SDN architecture provides a comprehensive view of the control plane functionalities that enable network slicing, its centralized nature cannot support scalable deployment. It also lacks the capability for efficient management over the life cycle of network slices, which is critical to enable dynamic slicing for networks with time-varying resource availability and user demands [36]. To address these issues, the authors in [36] suggested integrating the ETSI's NFV architecture [37] to further improve the flexibility of network slicing. The integrated architecture is illustrated in Figure 1(a). In this architecture, each service provider converts its physical resources into a set of virtualized resources (i.e., virtual machines) that can be accessed by all tenants. Each tenant can then manage a particular set of slices, consisting of virtualized network functions (VNFs) that are appropriately composed by a tenant SDN controller to deliver the requested services to end users. This architecture improves the flexibility and scalability of ONF's network slicing architecture by introducing the SDN controller at each tenant and service provider.

In [16], [38]–[40], the authors introduced the concept of 5G network slicing broker in 3GPP service working group SA 1. The proposed concept is based on the 3GPP's network sharing management architecture [11], [12], as illustrated in Figure 1(b). In this concept, each tenant can acquire slices from service providers to run network functions. In contrast to the ONF's network slicing architecture in which the slicing is created by exchanging resource usage information between tenants and service providers, the 5G network slicing broker allows the service providers to directly create network slices for tenants. It can therefore support on-demand resource allocation and admission control. However, the network slicing broker only supports slicing of networking resources that are centrally controlled by a master operator-network manager (MO-NM)[1].

B. Regional SDN-based Orchestrator

The above mentioned SDN architectures cannot be directly applied to large-scale networks with time-varying demands and resource availability, for the following reasons:

1) *Difficulty to Synchronize Over Large-scale Network Systems* – In the ONF architecture, the SDN controllers at tenants and service providers must synchronize with each other when managing and distributing resources. This can be quite challenging when performed over a large-scale network, covering a wide geographical area.
2) *High Coordination and Information Exchange Overhead* – One of the main motivations of introducing fog computing is to reduce latency by performing computations closer to users.

[1]According to 3GPP's network sharing management architecture [11], [12], in order to ensure optimized and secure resource allocation, a single master operator must be assigned as the only entity to centrally monitor and control network resources.

However, deploying a centralized SDN controller to collect and keep track of global user demands and resource usage information, and making a centralized decision about resource distribution will result in intolerably high coordination overhead and latency.

3) *Temporal and Spatial Variations of Resource Availability* – The energy and computational resources available at fog nodes may exhibit temporal and spatial variations. It is difficult to enable dynamic resource allocation over different nodes at different times using a monolithic architecture.

In this paper, we introduce a dynamic network slicing architecture that supports large-scale fog computing on a new network entity, the regional SDN-based orchestrator. In our architecture, each fog node i coordinates with a subset of its neighboring fog nodes $\mathcal{C}_i$ via a regional SDN-based orchestrator to create network slices for a common set of services requested by local users. More specifically, each tenant sends the resource request together with the location information of each user. The workload request of each user is first assigned to the closet fog node, which then partitions its own resources according to received requests. If a fog node receives requests that exceed its available resources, it will coordinate with the regional SDN-based orchestrator to outsource a part of its load to neighboring fog nodes. Similarly, if a fog node has surplus resources, it will report this surplus to the regional orchestrator, who will then coordinate with other fog nodes and forward appropriate amounts of their workload to the nodes with surplus. Our proposed architecture is illustrated in Figure 1(c).

The proposed architecture is general and can be applied to various network systems. For example, if the fog computing network is a cellular network (e.g., the cellular operator deploys mini-servers in some base stations), the regional SDN-based orchestrator can be implemented in each serving gateway (S-GW) and can then coordinate workload distribution between locally connected fog nodes via the internal X2 interface.

Our architecture balances coordination complexity and scalability. However, it is known that distributed optimization without a centralized control generally results in sub-optimal solutions. In addition, the time-varying energy harvesting process further increase the complexity of the optimization problem. In the rest of this paper, we introduce a stochastic game-theory-based distributed framework that can achieve the long-term optimal network slicing solution without requiring centralized coordination between fog nodes.

IV. Problem Formulation

We consider a fog computing network deployed by the FSP. The network consists of a set of N fog nodes, labeled as $\mathcal{F} = \{1, 2, \ldots, N\}$. Each node i serves a set $\mathcal{B}_i$ of tenants located in its coverage area. Each tenant can request multiple types of service with different QoS requirements, in here measured by the service response time. Examples of services that require ultra-low response time (<10 ms) and high computational resources include virtual reality (VR), augmented reality (AR), as well as guidance and planning services for high-speed (self-driving) vehicles [41], [42]. Other services that can tolerate higher latency (e.g., ≈100 ms) include speech recognition and language translation. Let $\mathcal{V}$ be the set of K types of service that can be supported by the FSP. Each tenant can request a subset of the service types in $\mathcal{V}$. We use superscript k to denote the parameters related to the kth service type. Let $\theta^{(k)}$ be the maximum tolerable response time for type k service. In this paper, we consider computational resource-limited scenarios and focus on the slicing of the computing resources of the FSP. We assume the tenants have sufficient communication resources to transfer their workload to the fog nodes. Our methods can be directly extended to support slicing of both communication and computing resources, e.g., if network resources obtained by each tenant can support a portion of the users' workload, then the tenant will only request slices of computational resources to process this portion of workload. We list the main notations adopted in this paper in Figure 3.

A. Resource Constraints

We consider an energy-harvesting fog computing network with limited computational and energy resources. In particular, the workload processing service for each fog node is powered by harvested energy. We assume time-varying (slotted) energy harvesting and workload arrival processes as in [8], [43]. Each fog node can harvest different amounts of energy and receive different amounts of workload from its users during different time slots. However, the workload arrival rate as well as the computational resource activated by each fog node are assumed to be constant within each time slot. We follow a commonly adopted setting [44], [45] and assume that the workload arrival process for each service type k in each time slot t at each tenant $b \in \mathcal{B}_i$ follows a Poisson process $\mathcal{P}\left(\tilde{\lambda}^{(k)}_{i[b],t}\right)$ where $\lambda^{(k)}_{i[b],t}$ is average number of requests received in time slot t. Accordingly, the aggregated workload arrival rate for each service type k at each fog node i also follows a Poisson distribution $\mathcal{P}\left(\lambda^{(k)}_{i,t}\right)$, where $\lambda^{(k)}_{i,t} = \sum_{b\in\mathcal{B}_i} \tilde{\lambda}^{(k)}_{i[b],t}$. We focus on the computational resource partition/slicing and energy scheduling for fog nodes and assume that the NSP has an unlimited energy supply (e.g., powered by the electric grid). Several previous works considered renewable-energy-supported communication services for large-scale network infrastructures [46]–[48]. How to simultaneously optimize the usage of renewable energy for both communications and computational services is left for future work.

1) Computational Resource Constraint: We assume each fog node has limited computational resources that can be dynamically activated and deactivated according to energy availability.

Various energy-control approaches have been proposed to allow electronic devices to dynamically adjust their power consumption according to the available energy supply [49]. For example, a fog node can scale up or down the frequencies of its processing units depending on the computational workload. Another simpler and more widely adopted approach is to allow each node to dynamically switch on and off some of its processing units according to energy availability [50]. In this paper, we adopt the latter approach and assume that each fog node i has a minimum amount of computational resource, measured by the amount of workload that can be processed by each of its processing units per time slot. Let w_i be the service rate that can be provided by each unit of computational resource at fog node i. Let $p_{i,t}^{(k)}$ be the number of computational resource units activated by fog node i to serve the kth service type during time slot t. The maximum service rate that can be supported by node i for service type k can then be written as $w_i p_{i,t}^{(k)}$. The workload that can be offloaded by each node i during time slot t cannot exceed the maximum service rate. In particular, let $\alpha_{i,t}^{(k)}$ be the portion of received type k workload that is offloaded to fog node i, $0 \leq \alpha_{i,t}^{(k)} \leq 1$, we have the following constraint:

$$\alpha_{i,t}^{(k)} \lambda_{i,t}^{(k)} \leq w_i p_{i,t}^{(k)}, \ \forall i \in \mathcal{F}. \tag{1}$$

Note that $\alpha_{i,t}^{(k)} = 0$ means that node i cannot offload any type k workload during time slot t. In this case, this workload has to be forwarded to other neighboring fog nodes or directly to the CDCs. If $\alpha_{i,t}^{(k)} = 1$, then fog node i can offload all received type k workload.

In energy-harvesting fog computing networks, different fog nodes can receive different amounts of workload and harvest different amounts of energy in each time slot. The workload processing capability can be further improved if nodes that harvest more renewable energy than needed can also process part of the workload received by fog nodes that cannot harvest sufficient energy. We therefore consider a cooperative setting in which two or more fog nodes can help each other process their workloads. Note that allowing every fog node to always forward part of its workload to other fog nodes is uneconomic and difficult to manage. Each fog node should only coordinate with a limited set of closely located fog nodesvia the regional SDN-based orchestrator. Similarly, each fog node can help only its neighboring fog nodes. Let $\mathcal{C}_i$ be the set of neighboring fog nodes of fog node i, $\mathcal{C}_i \subseteq \mathcal{F} \backslash \{i\}$.

Suppose fog node i decides to offload $\alpha_{i,t}^{(k)} \lambda_{i,t}^{(k)}$ workload of type k service in time slot t with the help of its neighboring fog nodes. Node i will need to carefully divide this total workload into $|\mathcal{C}_i|$ partitions, to be forwarded to its neighboring fog nodes. Let $\alpha_{im,t}^{(k)}$ be the portion of type k load of fog node i to be forwarded to fog node m, $m \in \mathcal{C}_i$, at time slot t. We also use

$\alpha_{ii,t}^{(k)}$ to denote the portion of type k load that will be processed by node i itself. Clearly,

$$0 < \alpha_{i,t}^{(k)} = \sum_{j \in \mathcal{C}_i \cup \{i\}} \alpha_{ij,t}^{(k)} \leq 1, \ \forall i \in \mathcal{F}. \tag{2}$$

2) Energy Constraint: Let $e_{i,\text{unit}}$ be the amount of energy consumed by fog node i to activate each unit of computational resource. We can write the total energy consumed by fog node i to offload type k service in time slot t as $e_{i,t}^{(k)} = e_{i,\text{unit}} p_{i,t}^{(k)}$. The total energy consumed by fog node i during slot t is given by $e_{i,t} = \sum_{k \in \mathcal{V}} e_{i,t}^{(k)}$.

Each node i has equipped with a battery that can store up to $e_{i,\max}$ energy. We consider an energy-harvesting system with causality constraints. In particular, a node cannot consume the energy that will be harvested in the future. We further assume that a node cannot use the energy harvested in the current time slot. This is because the energy harvested by each fog node can be highly unstable and fluctuated. Most energy harvesting-based electronic devices have an energy-stabilizing circuit to stabilize the energy input into the battery, and each device is typically supplied by the energy output from its battery. We can write the battery level of node i at the beginning of time slot t $\tilde{e}_{i,t}$ as

$$\tilde{e}_{i,t} = \min\{e_{i,\max}, \tilde{e}_{i,t-1} + \hat{e}_{i,t-1} - e_{i,t-1}\}, \tag{3}$$

where $\hat{e}_{i,t-1}$ is the amount of energy that can be harvested by fog node i during time slot $t-1$. We have the following constraint for the energy that can be consumed by fog node i in each time slot t:

$$\sum_{k \in \mathcal{V}} e_{i,t}^{(k)} \leq \tilde{e}_{i,t}. \tag{4}$$

B. Problem Formulation

At the beginning of each time slot t, each node i needs to carefully decide the values of the following two vectors:

1) Energy distribution vector $\boldsymbol{d}_{i,t} = \langle e_{i,t}^{(k)} \rangle_{k \in \mathcal{V}}$, which specifies the amount of energy spent on activating and partitioning the limited computational resources to serve different types of services.
2) Workload offloading vector $\boldsymbol{\alpha}_{i,t}$ given by

$$\boldsymbol{\alpha}_{i,t} = \begin{cases} \langle \alpha_{i,t}^{(k)} \rangle_{k \in \mathcal{V}}, & \text{no cooperation,} \\ \langle \alpha_{im,t}^{(k)} \rangle_{k \in \mathcal{V}, m \in \mathcal{C}_i}, & \text{when cooperating with neighboring fog nodes} \end{cases} \tag{5}$$

which specifies the portions of the workload to be offloaded by fog node i with/without cooperation with its neighboring fog nodes.

We assume that the CDC can offer a certain reward to incentivize the workload offloading behaviors of fog nodes. Each fog node receives reward only for workload that can be processed within the required QoS measured by response time, i.e., the response time $\pi_{i,t}^{(k)}$ for the offloaded type k service of fog node i needs to satisfy $\pi_{i,t}^{(k)} \leq \theta^{(k)}\ \forall k \in \mathcal{V}$. The reward received by each node is closely related to the total amount of offloaded workload and types of service. This reward can be a monetary value paid to each fog node owner or a virtual currency that can be used by fog nodes to exchange a certain service from CDCs. If fog nodes are deployed and managed by the MNOs, the reward can be regarded as the control mechanism imposed by the MNOs to regulate the workload offloading behaviors of fog nodes. $\pi_{i,t}^{(k)}$ depends on the amount of offloaded workload and energy distributed for type k service. For example, suppose that fog node i decides to offload $\alpha_{i,t}^{(k)}\lambda_{i,t}^{(k)}$ workload of type k service by itself with $p_{i,t}^{(k)} = e_{i,t}^{(k)}/e_{i,\text{unit}}$ activated computational resource units. Consider, for simplicity, an M/M/1 queuing delay for each type of service at fog nodes. The response-time for type k service offloaded by fog node i in time slot t without the help from other nodes is given by [44], [51]:

$$\pi_{i,t}^{(k)} = \frac{1}{w_i p_{i,t}^{(k)} - \alpha_{i,t}^{(k)}\lambda_{i,t}^{(k)}}. \tag{6}$$

If node i forwards part of its offloaded workload to and/or help processing the workload from its neighboring fog nodes. We can then follow the same line as [44], [51] and write $\pi_{i,t}^{(k)}$ as

$$\pi_{i,t}^{(k)} = \sum_{m\in\mathcal{C}_i\cup\{i\}} \alpha_{im,t}^{(k)}\left(\tau_{im} + \frac{1}{w_m p_{m,t}^{(k)} - \sum_{j\in\mathcal{C}_i\cup\{i\}} \alpha_{jm,t}^{(k)}\lambda_{j,t}^{(k)}}\right) \tag{7}$$

where τ_{im} is the round-trip time between fog nodes i and m, with $\tau_{im} > 0$ for $m \neq i$ and $\tau_{ii} = 0$. Note that $\alpha_{jm,t}^{(k)} \neq 0$ ($\alpha_{jm,t}^{(k)} = 0$) for $m \in \mathcal{C}_j$ means that fog node m has (does not have) extra computational resource to process the workload of fog node j.

Let $\rho_i^{(k)}$ be the reward received by node i to successfully process each workload unit of type k. Generally speaking, the higher the required QoS, the higher the reward received by the service supporting fog nodes. However, with more stringent QoS, the total amount of workload that can be offloaded by each fog node will also decrease. The main objective for each fog node is to maximize the amount of offloaded workload that can be supported within the allowed response time. We follow a commonly adopted setting and assume the reward of fog node i is linearly proportional to the amount of offloaded services. More specifically, the reward obtained by fog node i from offloading type k service is given by

$$\varpi_{i,t}^{(k)}\left(\boldsymbol{\alpha}_{i,t}^{(k)}\right) = \rho_i^{(k)} \sum_{m\in\mathcal{C}_i\cup\{i\}} \alpha_{im,t}^{(k)}\lambda_{i,t}^{(k)}. \tag{8}$$

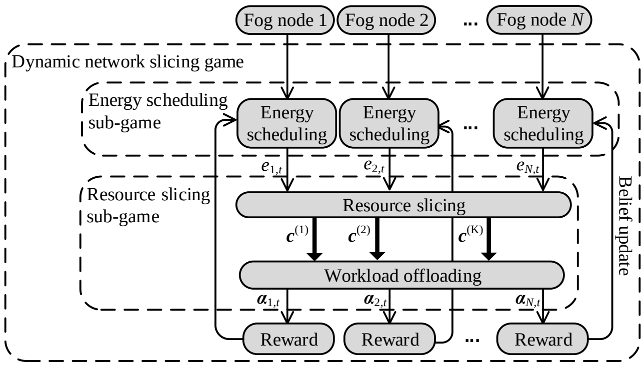


Fig. 2. Relationship between different sub-games in dynamic network slicing game.

Symbol	Definition
$\mathcal{F}$	Set of fog nodes
$\mathcal{V}$	Set of service types supported by fog nodes
$\lambda_{i,t}^{(k)}$	Aggregated workload arrival rate for service type k at fog node i during time slot t
$\alpha_{i,t}^{(k)}$	Portion of received type k workload offloaded to fog node i
$p_{i,t}^{(k)}$	Number of computational resource units activated by fog node i to service type k service in time slot t
$e_{i,t}$	Energy consumed by fog node i in time slot t
$\boldsymbol{d}_{i,t}$	Energy distribution vector of fog node i in time slot t
$\boldsymbol{\alpha}_{i,t}$	Workload offloading vector of fog node i in time slot t
$\pi_{i,t}^{(k)}$	Response-time provided by fog node i for processing type k service in time slot t
$\varpi_{i,t}^{(k)}$	Reward obtained by fog node i from processing type k service during time slot t
$\boldsymbol{c}^{(k)}$	Resource slicing vector for type k service

Fig. 3. List of notations.

Motivated by the fact that practical network systems can often tolerate small periods of performance degradation as long as the long-term average performance is good, we consider a system in which the main objective for each fog node is to maximize its long-term discounted reward. We can then write the sequential resource slicing and workload offloading problem for each fog node i as follows:

$$\max_{\langle \boldsymbol{d}_{i,t}, \boldsymbol{\alpha}_{i,t} \rangle_{t=0,1,\ldots}} \mathbb{E}\left(\lim_{N \to \infty} \sum_{t=0}^{N} \gamma^t \sum_{k \in \mathcal{V}} \varpi_{i,t}^{(k)} \left(\boldsymbol{\alpha}_{i,t}^{(k)} \right) \right) \tag{9}$$

$$\text{s.t.} \quad \pi_{i,t}^{(k)} \leq \theta^{(k)}, \text{ and constraints (1), (2), and (4) } \forall k \in \mathcal{V}, t = 0, 1, \ldots. \tag{10}$$

V. Dynamic Network Slicing Game

In this section, we formulate a so-called *dynamic network slicing game* and use it to investigate energy and computational resource partitioning/slicing among fog nodes under time-varying energy harvesting and workload arrival processes. Our proposed game consists of two sub-games: resource slicing sub-game and energy scheduling sub-game, as illustrated in Figure 2. The first sub-game characterizes the energy distribution and workload offloading decision made by each fog node based on the scheduled energy usage and received workload in each time slot. The second sub-game investigates the scheduling of energy usage for fog nodes based on the statistics of the long-term energy harvesting and workload arrival processes. Next, we give a more detailed discussion about these two sub-games.

A. Resource Slicing Sub-game

We first consider resource distribution among fog nodes in a single time slot t. Suppose the workload arrival rates $\boldsymbol{\lambda}_{i,t} = \langle \lambda_{i,t}^{(k)} \rangle$ for all supported same types at every fog node i is fixed. Each fog node i has already decided the total amount of energy $e_{i,t}$ it will spend to serve supported services, $0 < e_{i,t} \leq \tilde{e}_{i,t}$. Note that $e_{i,t}$ specifies the total amount of resources that is available at node i to distribute among different types of service in time slot t. We use $\boldsymbol{e}_t = \langle e_{i,t} \rangle_{i \in \mathcal{F}}$ to denote the energy usage vector for all fog nodes in time slot t. We discuss how to schedule energy usage over different time slots in the next subsection.

The main objective for each fog node i is to carefully decide the energy distribution vector $\boldsymbol{d}_{i,t}$ and workload offloading vector $\boldsymbol{\alpha}_{i,t}$ for the rest of the time slot t. Since each fog node can only receive reward from the offloaded workload, it should always utilize all the available energy resource, i.e., $\boldsymbol{d}_{i,t}$ needs to satisfy $e_{i,t} = \sum_{k \in \mathcal{V}} e_{i,t}^{(k)}$. We use the framework of the overlapping-coalition-formation game to model the energy and computational resource slicing problem among fog nodes. The overlapping coalition formation game has been widely applied to study the resource allocation problem among multiple players. In this game, multiple players can form different coalitions and distribute their resources to serve different types of services. Coalitions are said to be overlapped when the same player joins different coalitions to serve different service types [52].

Let us formally introduce the resource slicing sub-game as follows.

Definition 1: A *resource slicing sub-game* is defined by a tuple $\mathcal{G} = \langle \mathcal{F}, \boldsymbol{e}_t, \mathcal{V}, \boldsymbol{\varpi} \rangle$ where $\mathcal{F}$ is a set of fog nodes that correspond to the players of the game, $\boldsymbol{e}_t = \langle e_{i,t} \rangle_{i \in \mathcal{F}}$ is the energy resources that can be distributed by fog nodes, $\mathcal{V}$ is the set of service types for each fog node to distribute energy, $\boldsymbol{\varpi}$ is the vector of rewards received by fog nodes.

We give a more detailed discussion for each element in the above game as follows. Each fog node can divide its energy $e_{i,t}$ into different partitions (slices) each of which will be allocated to activate the computational resource to support a specific type of service. Each fog node needs to carefully decide the amount of workload that can be offloaded by itself and/or with the help of its neighboring fog nodes. The main objective for each fog node is to maximize its reward received in the currently time slot. We define a (resource) slice $\boldsymbol{c}^{(k)}$ serving type k service as a vector of energy distributed by fog nodes to serve type k service, i.e., $\boldsymbol{c}^{(k)} = \langle e_{i,t}^{(k)} \rangle_{i \in \mathcal{F}}$. Note that it is not necessary for every fog node to distribute energy to support all types of services, e.g., fog node i may have $e_{i,t}^{(k)} = 0$ for some $k \in \mathcal{V}$. We denote the support of $\boldsymbol{c}^{(k)}$ as $\mathrm{supp}\left(\boldsymbol{c}^{(k)}\right) = \{i \in \mathcal{F} : e_{i,t}^{(k)} \neq 0\}$. We define a (resource) slicing structure $\boldsymbol{c} = \langle \boldsymbol{c}^{(k)} \rangle_{k \in \mathcal{V}}$ as a vector specifying the energy allocations for fog nodes among all supported services.

We consider a transferrable-utility game setting in which the total reward obtained by a slice can be freely transferred among contributing fog nodes. In this case, the main objective for each fog node is to maximize the total workload offloaded for each type of service. We define the worth of slice $\boldsymbol{c}^{(k)}$ as the total reward that can be obtained by all the member fog nodes distributing energy to type k service. In particular, we can write the worth of a slice $\boldsymbol{c}^{(k)}$ as

$$v\left(\boldsymbol{c}^{(k)}\right) = \sum_{i \in \mathrm{supp}(\boldsymbol{c}^{(k)})} \varpi_{i,t}^{(k)}\left(\alpha_{i,t}^{(k)}\right). \tag{11}$$

Let us consider the workload offloading vectors for each given slicing $\boldsymbol{c}$. We can show that the optimal workload offloading vector $\boldsymbol{\alpha}^{(k)*}$ for a given slicing structure $\boldsymbol{c}$ that maximizes the reward of slice for type k service is unique. More specifically, let us first consider the case that each fog node cannot forward its workload to other fog nodes but can only process the received workload by itself. Since the reward received by each fog node i is proportional to the amount of offloaded workload, for a given energy distribution vector $\boldsymbol{d}_{i,t}$ of fog node i, the optimal portion of offloaded workload needs to satisfy

$$\alpha_{i,t}^{(k)*}\left(e_{i,t}^{(k)}\right) = \min\left\{1, \frac{w_i e_{i,t}^{(k)}}{\lambda_{i,t}^{(k)} e_i} - \frac{1}{\theta^{(k)}\lambda_{i,t}^{(k)}}\right\} \; \forall k \in \mathcal{V}. \tag{12}$$

From (12), we can observe that since, for a given energy distribution vector $\boldsymbol{d}_{i,t} = \langle e_{i,t}^{(k)} \rangle_{k \in \mathcal{V}}$, the optimal workload offloading vector $\boldsymbol{\alpha}_{i,t}^{*}\left(\boldsymbol{d}_{i,t}\right) = \langle \alpha_{i,t}^{(k)*}(e_{i,t}^{(k)}) \rangle_{k \in \mathcal{V}}$ can be determined, the optimization problem for both computational and energy resource distribution is equivalent to only optimizing the energy distribution $\boldsymbol{d}_{i,t}$ among all supported types of service.

If each fog node can cooperate with its neighboring fog nodes as described in Section IV, we can write the optimal workload offloading vector $\boldsymbol{\alpha}^{(k)*}\left(\boldsymbol{c}^{(k)}\right)$ under a given slice $\boldsymbol{c}^{(k)}$ as

$$\begin{aligned} \boldsymbol{\alpha}^{(k)*}\left(\boldsymbol{c}^{(k)}\right) = \arg \max_{\boldsymbol{\alpha}^{(k)} = \langle \alpha_{i,t}^{(k)} \rangle_{k \in \mathcal{V}, i \in \mathcal{F}}} & \varpi_{i,t}^{(k)}\left(\boldsymbol{\alpha}^{(k)}\left(\boldsymbol{c}^{(k)}\right)\right) \\ \text{s.t. } \; \pi_{i,t}^{(k)} \leq \theta^{k}, \forall k \in \mathcal{V}, & \text{ and constraints (1), (2), and (4).} \end{aligned} \tag{13}$$

Following the same line as [51], we can prove that (13) is a convex optimization problem and the optimal value of $\boldsymbol{\alpha}^{(k)*}\left(\boldsymbol{c}^{(k)}\right)$ is unique if all fog nodes have decided their energy allocated to service type k. We omit the detailed derivation due to limit of space. In other words, instead of optimizing both $\boldsymbol{\alpha}^{(k)*}\left(\boldsymbol{c}^{(k)}\right)$ and $\boldsymbol{c}^{(k)}$, fog nodes only needs to decide the optimal slicing $\boldsymbol{c}^{(k)*}$ for each service type k.

We can observe that the worth function is monotone, i.e., $v\left(\boldsymbol{c}^{(k)}\right) \geq v\left(\boldsymbol{c}^{(k)\prime}\right)$ for any $\boldsymbol{c}^{(k)}$, $\boldsymbol{c}^{(k)\prime}$ such that $e_{i,t}^{(k)} \geq e_{i,t}^{(k)\prime}$ for all $i \in \mathcal{F}$. In other words, the more energy has been distributed to a slice, the higher reward can be received by the fog nodes.

We define an reward allocation among member fog nodes for each slice $\boldsymbol{c}^{(k)}$ as $\boldsymbol{\varpi}^{(k)} = \langle \varpi_{i,t}^{(k)} \rangle_{i \in \mathrm{supp}(\boldsymbol{c}^{(k)})}$ which describes the worth distributed among fog nodes for serving type k service. $\boldsymbol{\varpi}^{(k)}$ is said to be efficient if $\sum_{i \in \mathrm{supp}(\boldsymbol{c}^{(k)})} \varpi_{i,t}^{(k)} = v\left(\boldsymbol{c}^{(k)}\right)$. $\boldsymbol{\varpi}^{(k)}$ is also called imputation if it is efficient and satisfies the individual rationality, i.e., $\varpi_{i,t}^{(k)} \geq v\left(\underline{\varpi}_{i,t}^{(k)}\right)$ where $\underline{\varpi}_{i,t}^{(k)}$ is the reward obtained by fog node i for offloading type k service if fog node i cannot cooperate with other fog nodes. We refer to a (resource) slicing agreement as a tuple $\langle \boldsymbol{c}, \boldsymbol{\varpi} \rangle$ where $\boldsymbol{c} = \langle \boldsymbol{c}^{(k)} \rangle_{k \in \mathcal{V}}$ and $\boldsymbol{\varpi} = \langle \boldsymbol{\varpi}^{(k)} \rangle_{k \in \mathcal{V}}$.

The resource distribution and negotiation among fog nodes across different types of services can be very complex. For example, when two or more fog nodes decide to cooperate to offload a specific type of service, they can also impose a certain term that may affect their cooperation with other fog nodes when serving some other types of service. When a fog node deviates from a slicing agreement for a specific type of service, it will also affect its cooperation with other fog nodes in other service types. The main solution concept in the resource slicing sub-game is the *core*. We extend the concept of the conservative core in the overlapping coalition formation game into our resource slicing sub-game as follows:

Definition 2: Given a resource slicing sub-game $\mathcal{G} = \langle \mathcal{F}, \boldsymbol{c}_t, \mathcal{V}, \boldsymbol{\varpi} \rangle$ and a subset of fog nodes $\mathcal{N} \subseteq \mathcal{F}$. Suppose $\langle \boldsymbol{c}, \boldsymbol{\varpi} \rangle$ and $\langle \boldsymbol{c}', \boldsymbol{\varpi}' \rangle$ are two slicing agreements such that for any slice $\boldsymbol{c}^{(k)} \in \boldsymbol{c}$ either $\mathrm{supp}(\boldsymbol{c}^{(k)}) \subseteq \mathcal{N}$ or $\mathrm{supp}(\boldsymbol{c}^{(k)}) \subseteq \mathcal{F} \setminus \mathcal{N}$. We say that slicing agreement $\langle \boldsymbol{c}', \boldsymbol{\varpi}' \rangle$ is a profitable deviation of $\mathcal{N}$ from $\langle \boldsymbol{c}, \boldsymbol{\varpi} \rangle$ if for all $j \in \mathcal{N}$, we have $\varpi_j(\boldsymbol{c}') > \varpi_j(\boldsymbol{c})$. We say that a slicing agreement $\langle \boldsymbol{c}, \boldsymbol{\varpi} \rangle$ is in the core of $\mathcal{G}$ if no subset of $\mathcal{F}$ has a profitable deviation from it. In other words, for any subsets of fog nodes $\mathcal{N} \subseteq \mathcal{F}$, any slicing structure $\boldsymbol{c}'$, and any imputation $\boldsymbol{\varpi}'$, we have $\varpi_j'(\boldsymbol{c}') \leq \varpi_j(\boldsymbol{c})$.
We can prove the following result.

Theorem 1: The core of the resource slicing sub-game is non-empty and consists of an unique outcome that maximizes the social welfare.

Proof: See Appendix A. ∎

From Theorem 1, we can observe that if all the fog nodes have decided their total amount of energy consumed in each time slot, the optimal slicing structure that maximizes the total reward of the fog layer is unique and stable, that is, no fog node can benefit by unilaterally deviate from this structure. In the next subsection, we investigate the energy scheduling for fog nodes in which each fog node i seeks an optimal energy scheduling policy that can maximize its long-term discounted reward.

B. Energy Scheduling Sub-game

From the previous discussion, we can observe that resource slicing among fog nodes in each time slot is closely related to the total amount of energy that can be used by fog nodes to activate their computational resources. Each fog node can further improve its long-term reward by scheduling its energy usage over different time slots according to the evolution of the energy harvesting and workload arrival processes. In this section, we assume for each given energy usage vector $\boldsymbol{e}_t$, fog nodes can always slice their energy following the same interaction described in Section V-A. We model the scheduling of energy usage among fog nodes in a time-varying environment as a stochastic game, we refer to as the energy scheduling sub-game. We have the following definition.

Definition 3: A *energy scheduling sub-game* is a tuple $\mathcal{G}' = \langle \mathcal{F}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{A}}, \mathcal{S}, \Omega, \boldsymbol{\Theta}, T, \boldsymbol{\varpi}, \rangle$ where $\mathcal{F}$ is the set of fog nodes, $\boldsymbol{\mathcal{Y}} = \times_{i \in \mathcal{F}} \mathcal{Y}_i$ is the set of type profile where $\mathcal{Y}_i$ is the type space for each fog node i, $\boldsymbol{\mathcal{A}}$ is the set of action profile, $\mathcal{S}$ is the set of possible outcome states, Ω is the set of observations that can be obtained by each fog node, $\boldsymbol{\Theta}$ is the observation function of fog nodes, $T(s', \boldsymbol{a}, s)$ is the state transition function, and $\boldsymbol{\varpi}$ is the vector of reward functions for fog nodes.

We give a detailed discussion on each of the above elements as follows: each fog node (player) is assumed to be rational and always try to maximize its long-term discounted reward. The action of each fog node specifies the energy it scheduled on activating the computational resources as well as the resource distribution in each time slot, i.e., we have $a_{i,t} = \langle \boldsymbol{d}_{i,t}, \boldsymbol{\alpha}_{i,t}, e_{i,t} \rangle$. The type of each fog node reflects its "workload offloading capability" related to how much this fog node can help or rely on other fog nodes to offload service workload. It depends on the battery level, computational power, and the number of neighboring fog nodes, QoS requirements of associated users, as well as the long-term energy harvesting and workload arrival processes. Each fog node can perfectly know its own type but not those of others. The state, denoted as $s_t \in \mathcal{S}$ in time slot t, is a composite variable of the battery level $\tilde{\boldsymbol{e}}_t$ and workload arrival rates $\boldsymbol{\lambda}_t$ of fog nodes. It is known that if the duration of each time slot is short enough, the energy harvesting and workload arrival processes for fog nodes can possess the Markov property, that is the conditional probability distribution of future states only depends on the present state [43], [53]. Let $\Pr(\hat{\boldsymbol{e}}_t | \hat{\boldsymbol{e}}_{t-1})$ and $\Pr(\boldsymbol{\lambda}_t | \boldsymbol{\lambda}_{t-1})$ be the transition probabilities for energy harvesting and workload arrival process between two consecutive time slots where $\hat{\boldsymbol{e}}_t = \langle \hat{e}_{i,t} \rangle_{i \in \mathcal{F}}$ and $\boldsymbol{\lambda}_t = \langle \lambda_{i,t} \rangle_{i \in \mathcal{F}}$. Since the workload arrival and energy harvesting are two independent processes, we can then apply similar approaches as [53], [54] and calculate the state transition function $T(s_t, s_{t-1}, \boldsymbol{a}_{t-1})$ as follows:

$$\begin{aligned} T(s_t, s_{t-1}, \boldsymbol{a}_{t-1}) &= \Pr(\tilde{\boldsymbol{e}}_t, \boldsymbol{\lambda}_t | \boldsymbol{e}_{t-1}, \boldsymbol{\alpha}_{t-1}, \tilde{e}_{t-1}, \boldsymbol{\lambda}_{t-1}) \\ &= \frac{\Pr(\boldsymbol{\lambda}_t | \boldsymbol{\lambda}_{t-1}) \Pr(\hat{e}_t | \hat{e}_{t-1}) \Pr(\tilde{e}_t, \boldsymbol{\lambda}_t | \boldsymbol{e}_{t-1}, \boldsymbol{\alpha}_{i,t-1}, \tilde{e}_{t-1}, \boldsymbol{\lambda}_{t-1})}{\sum_{\boldsymbol{\lambda}_t, \hat{e}_t} \Pr(\boldsymbol{\lambda}_t | \boldsymbol{\lambda}_{t-1}) \Pr(\hat{e}_t | \hat{e}_{t-1}) \Pr(\tilde{e}_t, \boldsymbol{\lambda}_t | \boldsymbol{e}_{t-1}, \boldsymbol{\alpha}_{t-1}, \tilde{e}_{t-1}, \boldsymbol{\lambda}_{t-1})}, \end{aligned} \tag{14}$$

where $\tilde{\boldsymbol{e}}_t = \{\tilde{e}_{i,t}\}_{i\in\mathcal{F}}$ and $\tilde{e}_{i,t}$ is given in (3). Each fog node cannot observe the complete state (e.g., the battery levels and the workload arrival rate for other nodes) at the beginning of each time slot but can obtain a partial observation labeled as $o_{i,t}$. The observation of each fog node at the beginning of each time slot can be its observed initial environmental condition and workload received by itself as well as that from the neighboring fog nodes. For example, if all fog nodes are co-located in the same area, they may observe the similar energy harvesting process (e.g., solar or wind powers collected by co-located fog nodes can have a strong correlation.) and traffic arrival rates (e.g., users located in the same coverage area can have similar traffic patterns.). Each fog node i can then infer the unknown information of other fog nodes from its own observed battery level $\tilde{e}_{i,t}$ and workload arrival rate $\lambda_{i,t-1}$ during the previous time slots. More specifically, fog node i can infer an observation function specifying the probability distribution of its possible observations given the action profile and resulting state as follows:

$$\Theta_{i,t}\left(o_{i,t}, \boldsymbol{a}_{i,t-1}, s_t\right) \quad = \quad \Pr\left(o_{i,t}|\boldsymbol{a}_{i,t-1}, s_t\right) = \frac{\Pr\left(\tilde{\boldsymbol{e}}_t, \boldsymbol{\lambda}_{t-1}|\boldsymbol{e}_{i,t-1}, \boldsymbol{\alpha}_{i,t-1}\right)}{\sum_{\tilde{\boldsymbol{e}}_{-i,t},\boldsymbol{\lambda}_{-i,t}} \Pr\left(\tilde{\boldsymbol{e}}_t, \boldsymbol{\lambda}_{t-1}|\boldsymbol{e}_{i,t-1}, \boldsymbol{\alpha}_{i,t-1}\right)}, \quad (15)$$

where $-i$ denotes all the fog nodes except fog node i. Each fog node i can establish a belief function $\hat{b}_i\left(\boldsymbol{y}_{-i}|s\right)$ about the unknown types of others under each possible state $s \in \mathcal{S}$. This belief function will help the fog node choose the most "capable" fog nodes from its neighbors to forward its workload without having the complete information about their types. Each fog node can decide its action at the beginning of each time slot by utilizing its observation and belief function. The action profile of all the fog nodes will jointly determine a stochastic outcome state of the game. Each outcome state and action profile determine the reward for each fog node. The main objective for each fog node is to find the optimal policy such that no fog node can further improve its long-term reward by unilaterally deviating from this policy.

Note that since each fog node cannot know the state and types of other fog nodes, it cannot know the exactly value of the final reward that can be obtained by joining each possible slice. However, the repeated interaction among fog nodes provides each fog node with the opportunities to learn from the past experience and estimate a potential value of the reward when it forms different slicing structures with its neighboring fog nodes. More specifically, each fog node i can establish and maintain a belief function reflecting its private belief about types of neighboring fog nodes. Each fog node i's belief function b_i is a probability distribution function over types of other fog nodes under each given state. We write $b_i\left(\boldsymbol{y}_{-i}, s\right)$ as the probability that fog node i assigns to other fog nodes in state s where $-i$ denotes all fog nodes except fog node i. We follow the commonly adopted setting and assume the decision of each fog node cannot directly affect the decision making process of other fog nodes [54], [55]. In addition, each fog node can also utilize its observation to establish a belief function, a probability distribution function, about the state and combine this belief function about the state with the belief function about the workload offloading capabilities of other fog nodes under each given state. In particular, if a fog node i can have a belief function $b_{i,t}\left(\boldsymbol{y}_{-i,t}, s_t\right)$ about the state and types of other fog nodes at the beginning of time slot t, it can calculate the expected reward $\bar{\varpi}_{i,t}$ as

$$\bar{\varpi}_{i,t} = d_{i,t} \sum_{\substack{\boldsymbol{y}_{-i} \in \mathcal{Y}^{|\mathcal{C}_i|-1} \\ s_t \in \mathcal{S}}} b_{i,t}\left(\boldsymbol{y}_{-i,t}, s_t\right) \sum_{k \in \mathcal{V}} \varpi_i^{(k)} \left(\boldsymbol{\alpha}_t^{(k)*}\left(\boldsymbol{e}_t^{(k)}\right)\right).$$

It can be observed that if the belief function of each fog node can refect the true state and offloading capability of other fog nodes, each fog node can then decide optimal actions that can maximize its long-term expected rewards.

Let us now introduce a distributed algorithm based on B-POMDP for each fog node to establish and update its belief function about the state and the types of others from its past experience. Each fog node can then use the belief functions to sequence of actions that maximize its long-term discounted reward. We model the decision making process of each fog node as a B-POMDP. B-POMDP extends the traditional single-agent POMDP into multi-agent cases by allowing each agent to include its interactions with other agents as a part of the state space. We present the formal definition as follows:

Definition 4: A belief-state POMDP (B-POMDP) for the decision making of fog node i is defined as *B-POMDP*$_i = \langle \mathcal{S}_i, \boldsymbol{\mathcal{A}}, T_i', b_i, \Omega, \Theta, \varpi_i \rangle$ where $\mathcal{S}_i$ is the state space of fog node i, $\boldsymbol{\mathcal{A}}$ is the set of action profiles for the fog nodes, b_i is fog node i's belief function which specifies its belief about the state and types of other fog nodes, $T_i'(s', \boldsymbol{a}, s)$ is the transition dynamics specifying the probability of possible outcome state for fog node i given that action $\boldsymbol{a}$ has been taken in state s, Ω is the set of observation for each fog node, Θ is the observation function and ϖ_i is the reward function.

As mentioned earlier, the main difference between the B-POMDP and the single-agent POMDP is that in the former model, each fog node includes the decision making process of other fog nodes as a part of the environmental state. More specifically, in B-POMDP, the state space $\mathcal{S}_i = \mathcal{U} \times_{j \in \mathcal{F}\setminus\{i\}} \mathcal{M}_j$ for each fog node i consists of two parts: the set of states about the physical environment $\mathcal{U}$ which includes the harvested energy, battery level and received workload, and the set of possible model states about other fog nodes $\times_{j \in \mathcal{F}\setminus\{i\}} \mathcal{M}_j$ such as their types that specify the decision making of these fog nodes under each state of physical environment. Each fog node believes that all the other fog nodes decide their actions according to an unknown distribution and fog node i's prior belief about this distribution follows the Dirichlet distribution [8], [54], [55].

At the beginning of each time slot, each fog node can obtain an observation $o_{i,t}$ about the current physical environmental state $u_{i,t} \in \mathcal{U}$ of the system and can therefore follow the similar approach as POMDP to updates its belief about the environmental state as follows:

$$\begin{aligned} b'_{i,t}\left(u_{i,t}\right) &= \Pr\left(u_{i,t}|o_{i,t}, a_{i,t-1}, \hat{b}_{i,t-1}\right) \\ &= \frac{\Pr\left(o_{i,t+1}|u_{i,t+1}, a_{i,t}, \hat{b}_{i,t}\right) \sum_{u_{i,t}\in\mathcal{U}} \Pr\left(u_{i,t+1}|a_{i,t}, \hat{b}_{i,t}, u_{i,t}\right) \Pr\left(u_{i,t}|a_{i,t}, \hat{b}_{i,t}\right)}{\Pr\left(o_{i,t+1}|a_{i,t}, \hat{b}_{i,t}\right)} \\ &= \beta\Theta\left(u_{i,t+1}, a_{i,t}, o_{i,t+1}\right) \sum_{u_{i,t}\in\mathcal{U}} T\left(u_{i,t}, \boldsymbol{a}_{i,t-1}, u_{i,t-1}\right) \hat{b}_{i,t-1}\left(u_{i,t-1}\right), \end{aligned} \tag{16}$$

where $\beta = 1/\Pr(o_{i,t+1}|a_{i,t}, \hat{b}_{i,t})$ is a normalizing factor that is independent of u_t.

Each fog node cannot observe the types of other fog nodes but can derive the model information about the types of others after it finishes its interaction with other fog nodes and receives the reward at the end of each time slot. In other words, there is a mapping function g_i that maps the types of other fog nodes and the physical environmental state of the system to the final slicing structure and reward of fog node i, i.e., $\langle \varpi_{i,t}, \boldsymbol{c}_{i,t} \rangle = g_i(y_{i,t}, a_{i,t}, \boldsymbol{y}_{-i,t}, u_t)$ where $\boldsymbol{c}_{i,t}$ is the slices that consists of fog node i in time slot t. Note that fog node i cannot know g_i, but can develop a belief function about other fog nodes' type information by estimating the probability distribution about $\boldsymbol{y}_{-i,t}$ from its previous observations of $\varpi_{i,t}, \boldsymbol{c}_{i,t}, a_i, \boldsymbol{y}_{-i,t}$ and $u_{i,t}$. In this paper, we adapt Bayesian reinforcement learning for each fog node to learning the unknown type information of other fog nodes [10], [54]. Each fog node can update its belief information about the type of other fog nodes using the observed reward and slicing structure at the end of each time slot via Bayes' rule. More specifically, each fog node i in slices $\boldsymbol{c}_{i,t}$ can update its belief function about types of other fog nodes in $\boldsymbol{c}_{i,t}$ by

$$\begin{aligned} b''_{i,\boldsymbol{c}_{i,t}}\left(\boldsymbol{y}_{\boldsymbol{c}_{i,t}}\right) &= \Pr\left(\boldsymbol{y}_{\boldsymbol{c}_{i,t}}|a_{i,t}, u_{t-1}, b_{i,t-1}, o_{i,t}, u_t\right) \\ &= \frac{\Pr\left(u_t|a_{i,t}, \boldsymbol{y}_{\boldsymbol{c}_{i,t-1}}, o_{i,t}b_{i,t}\right) b_{i,t-1}\left(\boldsymbol{y}_{\boldsymbol{c}_{i,t-1}}\right)}{\Pr\left(u_t|a_{i,t}, o_{i,t}\right)} \\ &= \beta' T'_i\left(u_t, a_{i,t}, u_{t-1}, \boldsymbol{y}_{\boldsymbol{c}_{i,t-1}}\right) b_{i,t-1}\left(\boldsymbol{y}_{\boldsymbol{c}_{i,t-1}}, u_t\right) \end{aligned} \tag{17}$$

where $\beta' = 1/b'_{i,t}(u_t)$is a normalizing factor that is independent of $\boldsymbol{y}_{\boldsymbol{c}_{i,t}}$.

Each fog node can then use the obtained belief functions about the physical environmental state and types of other fog nodes to calculate the expected reward in each time slot. Each fog node will also evaluate the future expected reward by calculating a future updated belief state, i.e., we use $B_i\left(s_i, \boldsymbol{a}\right)$ to denote the estimated future belief state when the current state and action of fog node i are given by s_i and $\boldsymbol{a}$, respectively. At the beginning of each time slot t, fog node i can calculate $\bar{\varpi}'_i\left(\boldsymbol{c}_{i,t}, s_t, \boldsymbol{a}_t, b_{i,t}\right)$ by

$$\bar{\varpi}'_i\left(\boldsymbol{c}_{i,t}, s_t, a_{\mathcal{C}_t}, b_{i,t}\right) = \sum_{\substack{\boldsymbol{y}_{-i,t}\in\mathcal{Y}^{|\boldsymbol{c}_{i,t}|-1} \\ s_t\in\mathcal{S}}} b'_i\left(u_t\right) b''_i\left(\boldsymbol{y}_{-i,t}\right) \bar{\varpi}_{i,t}$$
$$+\gamma \sum_{o_{i,t}\in\Omega} \Theta\left(o_{i,t}, u_t, a_{i,t-1}\right) \upsilon_i\left(B_i\left(s_{i,t}, a_{i,t}\right)\right), \tag{18}$$

where the first term on the right-hand-side of above equation is the expected reward fog node i obtained in current time slot t, and the second term is the expected reward that fog node i can obtain in the future time slots, and $\upsilon_i\left(B_{i,t}\left(s_{i,t}, a_{i,t}\right)\right)$ is given by

$$\upsilon_i\left(B_{i,t}\left(s_t, a_{i,t}\right)\right) = \sum_{\boldsymbol{c}_{i,t}} \Pr\left(\boldsymbol{c}_{i,t}, \boldsymbol{a}_{t+1}, s_{t+1}\right| \tag{19}$$
$$B_i\left(s_t, a_{i,t}\right)\left.\right) \bar{\varpi}'_i\left(\boldsymbol{c}_{i,t}, s_{t+1}, \boldsymbol{a}_{t+1}, B_i\left(s_{i,t+1}, a_{i,t+1}\right)\right).$$

Following from the above analysis, we can write the optimal policy for each fog node to decide its action as:

$$a^*_{i,t} = \max_{a_{i,t}\in\mathcal{A}} \bar{\varpi}'_i\left(\boldsymbol{c}_{i,t}, s_t, \boldsymbol{a}_t, b_{i,t}\right). \tag{20}$$

We can then prove the following result.

Theorem 2: The belief function in (17) can always converge to a stationary distribution. The policy given in (20) is optimal for every initial states of fog nodes.

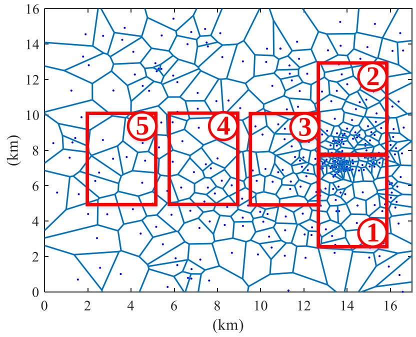
Proof: See Appendix B. ■

VI. Implementation and Numerical Results

Fog computing provides a promising solution to improve computational capability for the fast growing mobile applications in 5G networks [3], [56]. In this section, we consider cellular network system as an example to describe how to implement our proposed dynamic network slicing into 3GPP network sharing architecture. We then consider a fog computing setup deployed in a real cellular network infrastructure in the city of Dublin to evaluate the potential performance improvement that can be achieved by our proposed algorithms.

A. Dynamic Network Slicing for Fog Computing Supported 5G Networks

The dynamic network slicing among multiple fog nodes can be supported by the network sharing management architecture recently introduced by 3GPP [11], [12]. In particular, a *network slice* consists of a set of isolated computational and networking resources (e.g., processing units,



(a)

Area number	Total number of BSs	Average number of BSs within forwarding distance
1	53	6.40
2	44	3.82
3	31	2.39
4	24	1.50
5	12	1.00

(b)

Fig. 4. Distribution of BSs and considered areas.

network infrastructure, and bandwidth) orchestrated according to a specific type of service. Popular wireless services that can be supported by fog computing include voice process (e.g., voice recognition applications such Apple's Siri and Amazon's Alexa services) and image process (e.g., image recognition applications such as the traffic sign recognition in automotive devices). To facilitate on-demand resource allocation, admission control, workload distribution and monitoring, the regional SDN-based orchestrator can be deployed at the S-GW with accessability to the core network elements such as Mobility Management Entity (MME) and Packet Data Gateways (P-GW) via the S1 interface. Fog nodes can be deployed inside of the BSs. Regional SDN-based orchestrator can coordinate the workload distribution among connected fog nodes via the X2 interface. Regional SDN-based orchestrator can connect with the network element manager (NEM) of each BS (i.e., eNB) to evaluate the received workload for every type of service and decide the amount of workload to be offloaded by each connected fog node. It also performs workload monitoring, information exchange and control of the computational resources belonging to different fog nodes through the NEMs of their associated BSs.

B. Numerical Results

We simulate the possible implementation of fog computing infrastructure in over 400 BSs (including GSM and UMTS BSs) deployed by a primary MNO in the city of Dublin [57], [58]. The locations and coverage areas of the BSs are presented in Figure 4(a). To compare the workload offloading performance with different deployment densities of BSs, we consider 5 areas from the city center to the rural areas as shown in Figure 4(b). We assume a mini-server consisting of 100 processing units that can be activated or deactivated according to the energy availability has been built inside of each of these BS. We assume each fog node can support two types of services: image and voice recognition with maximum tolerable response-time of 50 ms and 100 ms, respectively. Each processing unit can process 10 image or 40

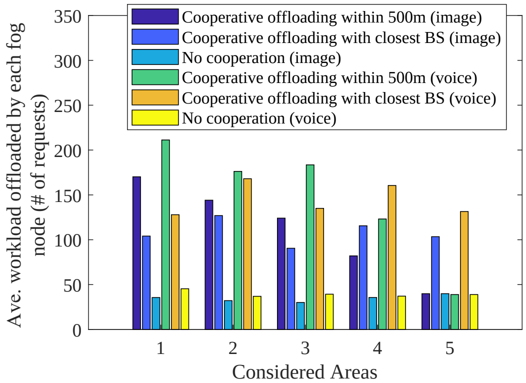


Fig. 5. Offloaded workload for both types of service in different areas shown in Figure 4.

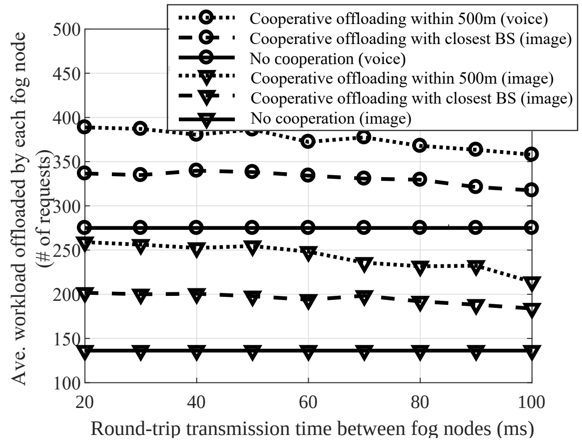


Fig. 6. Offloaded workload under different round trip transmission times between fog nodes

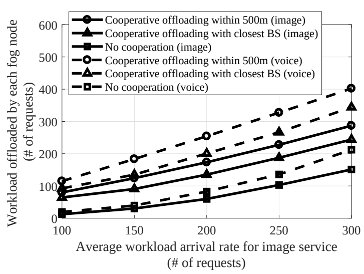


Fig. 7. Offloaded workload under different workload arrival rate for image service.

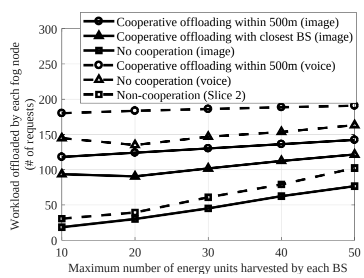


Fig. 8. Offloaded workload under different amount of harvested energy.

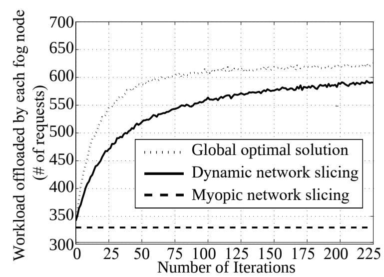


Fig. 9. Offloaded workload under different numbers of iterations.

voice recognition requests per second. We consider two cooperative offloading settings. In the first setting, each fog node can only cooperate with its closest fog node. In the second setting, each fog node can forward its workload to any neighboring fog nodes within a limited distance called (workload) forwarding distance. We assume the round-trip workload transmission latency between two fog nodes within the workload forwarding distance can be regarded as a constant which is to $\tau_{ij} = 20ms$.

To evaluate the performance improvement that can be achieved by allowing each BS to forward part of its received workload to its neighboring BSs, we compare the number of requests that can be processed by each BS in the five considered areas in Figure 5. We can observe that by allowing each BS to cooperate with all the neighboring BSs within the forwarding distance can significantly improve the numbers of offloaded requests for both supported services. We can also observe that even when each fog node can only cooperate with its closest fog node, the workload offloading capability in terms of the number of offloaded requests can almost doubled compared to the case without cooperative offloading. The workload offloading capability can improve more if each fog node can cooperate with more neighboring nodes. Note that in Figure

5, we observe that in areas 4 and 5, allowing each fog node to cooperate with other fog nodes within 500 meters cannot achieve the highest workload offloading performance. This is because in these two rural areas, some fog nodes cannot have any other fog nodes located in the 500 meters.

It can be observed that the round-trip workload transmission latency between neighboring fog nodes directly affect the performance of the cooperative offloading. In Figure 6, we compare the average number of offloaded service requests over all the considered areas when fog nodes within the workload forwarding distance have different round-trip transmission latencies between each other. We can again observe that allowing each fog node to cooperate with all the neighboring nodes within the forwarding distance achieves the best offloading performance among other strategies even when the transmission latency between fog nodes becomes large. This is because in our proposed dynamic network slicing, each fog node can carefully schedule the energy consumed for activating the computing resources at each time slot. More specifically, when the transmission latency between fog nodes is small, each fog node may spend more energy in processing the workload for others. On the other hand, when forwarding workload to other neighboring fog nodes will introduce high transmission latency, fog nodes will reserve more energy for its own use instead of helping others.

In Figure 7, we fixed the workload arrival rate of the voice service for each fog node and compare the workload that can be processed by the fog layer when fog nodes have different arrival rates for the image services. We can observe that both cooperative offloading settings can significantly improve the offloading performance for the video service. This is because when each fog node receives more workload for image service, it will be more likely to seek help from other member fog nodes and jointly offload the image service with others. We can also observe that each fog node is always willing to offload more workload for the voice service than the image service. This is because the voice service require less computational resources for each activated processing unit.

We consider the effect of energy harvesting process on the workload offloading performance of the fog layer in Figure 8 where we investigate the workload offloading performance of the fog layer when the maximum energy that can be harvested by each fog node is changed. Note that we assume the number of energy units that can be harvested by each fog node in each time slot is upper bounded by a maximum value. Therefore, the maximum amount of energy that can be harvested by each fog node also reflects the average energy that can be obtained by fog nodes as well as the energy available to activate the local computing resources. We observe that when the harvested energy for each fog node is limited, allowing two or more fog nodes to help each other can significantly improve the workload offloading performance of the fog layer. As the amount of harvested energy increases, more computational resources can be activated and the queuing delay of fog nodes will be further reduced.

Finally, in Figure 9, we present the convergence performance of our proposed dynamic network slicing with the learning algorithm discussed in Section V. We observe that our proposed approach can quickly converge to the optimal network slicing structure. In this paper, we focus on the distributed optimization of the fog layer consisting of multiple rational fog nodes that can make autonomous decision on resource slicing. Each fog node is selfish and will only offload workload for others if this can improve its reward. However, it is known that the total amount of workload that can be offloaded by the fog layer can be further increased if some fog nodes can sacrifice their performance and allow other fog nodes to utilize their computing resources. We refer to the solution that can maximize the total amount of offloaded workload of the entire fog layer without the constraint of rationality for each fog node as the global optimal solution which is also presented in Figure 9. As can be observed from Figure 9, although the global optimal solution can offer better offloading performance than our proposed framework, it however cannot provide sufficient incentive for fog nodes to form a stable slicing structure in which no fog node can benefit by deviating unilaterally. We also compare the performance of myopic network slicing in which each fog node only tries to maximize the current workload offloading performance without considering the future energy harvesting and workload arrival processes. We can observed that allowing each fog node to carefully schedule its energy usage can significantly improve the workload offloading capability for the fog computing networks.

VII. Conclusion

We have proposed the concept of dynamic network slicing for energy-harvesting fog computing networks consisting of a set of fog nodes that can offload and/or forward its received workload to CDCs using the energy harvested from natural environment. In this concept, each fog node can offload multiple types of service with QoS guarantee using its locally installed computing resources. The limited resources and uncertainty of the energy harvesting process restrict the total amount of workload that can be offloaded by each individual fog node. We introduce a dynamic network slicing architecture supporting large-scale fog computing networks on a new entity, the regional orchestrator, to improve the workload offloading performance of the fog layer. In this architecture, two or more neighboring fog nodes can coordinate through the region orchestrator to jointly offload their received workload from CDCs. To capture the fact that each fog node cannot know the global information but can make autonomous decisions using local information, we developed a stochastic overlapping coalition-formation game-based model to investigate the workload offloading problems. We observe that the workload offloading performance can be significantly improved if each fog node can learn from its previous interactions with others and sequentially refine the knowledge about the environmental state and private information of others. We have introduced a distributed optimization algorithm and proved that this algorithm can achieve the optimal network slicing policy among fog nodes. Finally, we have considered a wireless network that supports fog computing with renewable-energy-supply as a case study to evaluate the performance improvement that can be brought by our proposed framework.

ACKNOWLEDGMENT

The authors would like to thank Professor Luiz A. DaSilva and Dr. Jacek Kibilda at CONNECT, Trinity College Dublin to provide the BS location data of Dublin city.

APPENDIX A
PROOF OF THEOREM 1

To prove the core of the resource slicing sub-game is always non-empty, we need to first show that the resource slicing sub-game is a special overlapping-coalition-formation game that satisfies the property of convexity, that is, a coalition of players can obtain more reward when it joins a larger coalition. Let $\mathcal{E}(\mathcal{F})$ be the set of all feasible resource slicing agreements agreed by set $\mathcal{F}$ of fog nodes. We use $\langle \boldsymbol{c}^{\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{C}} \rangle$ to denote a slicing agreement mutually agreed by a subset $\mathcal{C} \subseteq \mathcal{F}$ of fog nodes. We give a formal definition as follows.

Definition 5: [52, Definition 13] A resource slicing game is convex if for each $\mathcal{C} \subseteq \mathcal{F}$ and $\mathcal{N} \subset \mathcal{O} \subseteq \mathcal{F} \backslash \mathcal{C}$, the following condition holds: for any $\boldsymbol{c}^{\mathcal{N}}, \boldsymbol{\alpha}^{\mathcal{N}} \in \mathcal{E}(\mathcal{N})$, any $\langle \boldsymbol{c}^{\mathcal{O}}, \boldsymbol{\alpha}^{\mathcal{O}} \rangle \in \mathcal{E}(\mathcal{O})$, and any $\langle \boldsymbol{c}^{\mathcal{N}\cup\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{N}\cup\mathcal{C}} \rangle \in \mathcal{E}(\mathcal{N}\cup\mathcal{C})$ that satisfies $\boldsymbol{\varpi}_i\left(\boldsymbol{c}^{\mathcal{N}\cup\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{N}\cup\mathcal{C}}\right) \geq \boldsymbol{\varpi}_i\left(\boldsymbol{c}^{\mathcal{N}}, \boldsymbol{\alpha}^{\mathcal{N}}\right)$, $\forall i \in \mathcal{N}$, there exists an outcome $\langle \boldsymbol{c}^{\mathcal{O}\cup\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{O}\cup\mathcal{C}} \rangle \in \mathcal{E}(\mathcal{O}\cup\mathcal{C})$ such that $\boldsymbol{\varpi}_i\left(\boldsymbol{c}^{\mathcal{O}\cup\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{O}\cup\mathcal{C}}\right) \geq \boldsymbol{\varpi}_i\left(\boldsymbol{c}^{\mathcal{O}}, \boldsymbol{\alpha}^{\mathcal{O}}\right)$, $\forall i \in \mathcal{O}$ and $\boldsymbol{\varpi}_i\left(\boldsymbol{c}^{\mathcal{O}\cup\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{O}\cup\mathcal{C}}\right) \geq \boldsymbol{\varpi}_i\left(\boldsymbol{c}^{\mathcal{N}\cup\mathcal{C}}, \boldsymbol{\alpha}^{\mathcal{N}\cup\mathcal{C}}\right)$, $\forall i \in \mathcal{C}$.

We have the following lemma.

Lemma 1: A resource slicing sub-game is convex.

Proof: We can observe that the reward function of the resource slicing sub-game is a linear combination of the reward obtained by workload offloaded by the fog nodes when they can cooperate with their neighboring fog nodes. It can be observed that different fog nodes have different sets of neighboring fog nodes. The more fog nodes can contribute to the workload offloading, the more resources can be distributed by all the member fog nodes. We can also observe that problem (13) is a linear function of $\boldsymbol{\alpha}_{i,t}$. In addition, as mentioned in Section V, each fog node will only cooperate with a subset of its neighboring fog nodes if it cannot obtain a higher reward by forming a coalition with other subsets of fog nodes. Let us write the solution of problem (13) as $\langle \boldsymbol{c}_i^{\mathcal{C}*}, \boldsymbol{\alpha}_i^{\mathcal{C}*} \rangle$ when the maximum set of fog nodes that can slice their resource to support all types of services is given by $\mathcal{C}*$. We can apply the standard convex optimization method to prove that the solution $\varpi_{i,t}\left(\langle \boldsymbol{c}_i^{\mathcal{C}*}, \boldsymbol{\alpha}_i^{\mathcal{C}*} \rangle\right)$ satisfies the following properties:

$$\begin{aligned} \varpi_{i,t}\left(\langle \boldsymbol{c}_i^{\mathcal{O}\cup\mathcal{C}*}, \boldsymbol{\alpha}_i^{\mathcal{O}\cup\mathcal{C}*} \rangle\right) &\geq \varpi_i\left(\langle \boldsymbol{c}_i^{\mathcal{O}*}, \boldsymbol{\alpha}_i^{\mathcal{O}*} \rangle\right), \\ \varpi_{i,t}\left(\langle \boldsymbol{c}_i^{\mathcal{O}\cup\mathcal{C}*}, \boldsymbol{\alpha}_i^{\mathcal{O}\cup\mathcal{C}*} \rangle\right) &\geq \varpi_i\left(\langle \boldsymbol{c}_i^{\mathcal{N}\cup\mathcal{C}*}, \boldsymbol{\alpha}_i^{\mathcal{N}\cup\mathcal{C}*} \rangle\right), \\ & \forall \mathcal{N} \subset \mathcal{O} \subseteq \mathcal{F} \backslash \mathcal{C}. \end{aligned} \tag{21}$$

We can therefore claim that the network slicing game is convex. This concludes the proof. ■

We can then use the following theorem given in [52] to prove the non-emptiness of the core for any resource slicing sub-game.

Theorem 3: [52, Theorem 3] If an overlapping coalition formation game is convex, and the worth v is continuous, bounded, monotone and the maximum number of partial coalitions that each MNO can be involved in is finite, then the core of the game is not empty.

From Section V, we can directly observe that the worth of the resource slicing sub-game satisfies all the above conditions. Therefore, we can claim that the core of the resource slicing sub-game is always non-empty. From the definition of the core and following the same line as [52], we can also prove that a network slicing agreement $\langle \boldsymbol{c}, \boldsymbol{\alpha} \rangle$ is in the core if and only if

$$\sum_{i \in \mathcal{F}} \varpi_i\left(\boldsymbol{c}, \boldsymbol{\alpha}\right) \geq v^*\left(\boldsymbol{c}^{\mathcal{F}}\right), \tag{22}$$

where $v^*\left(\boldsymbol{c}^{\mathcal{F}}\right)$ is the supremum of $v\left(\boldsymbol{c}^{\mathcal{F}}\right)$. In other words, any outcome in the core maximizes the social welfare. This concludes the proof.

APPENDIX B
PROOF OF THEOREM 2

We briefly describe the proof of Theorem 2 as follows. We can observe that the belief updating scheme for each fog node in (16) has the Markov property. That is, the updated belief function of fog node i is only related to its belief, outcome state and action in the previous time slot. We can also observe that (18) is equivalent to the Bellman equation for single-agent POMDP. In other words, if each fog node regards the environment as well as the decision making processes of the other fog nodes as part of the system state, the workload offloading problem for each fog node can be regarded as a single-agent POMDP. We can then follow the same line as [59] to prove that (20) is the optimal policy for each fog node to maximize its long-term workload offloading performance. We omit the details of the proof due to the limit of space. This concludes the proof.

REFERENCES

[1] L. Vaquero and L. Rodero-Merino, “Finding your way in the fog: Towards a comprehensive definition of fog computing,” *Proc. of ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[2] NGMN Alliance, “5G white paper,” Feb. 2015. [Online]. Available: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf

[3] A. Vahid Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, “Fog Computing: Principals, Architectures, and Applications,” *ArXiv e-prints*, Jan. 2016.

[4] S. Yi, C. Li, and Q. Li, “A survey of fog computing: Concepts, applications and issues,” in *Proc. of ACM Workshop on Mobile Big Data*, Hangzhou, China, Jun. 2015, pp. 37–42.

[5] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, “Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing,” in *Proc. of the IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, Athens, Dec. 2014, pp. 325–329.

[6] V. Chamola and B. Sikdar, “Solar powered cellular base stations: current scenario, issues and proposed solutions,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 108–114, May 2016.

[7] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, “Energy harvesting wireless communications: A review of recent advances,” *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[8] Y. Xiao, D. Niyato, Z. Han, and L. DaSilva, “Dynamic energy trading for energy harvesting communication networks: A stochastic energy trading game,” *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 12, pp. 2718–2734, Dec. 2015.

[9] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, “Wireless networks with RF energy harvesting: A contemporary survey,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 757–789, 2015.

[10] Y. Xiao, Z. Han, D. Niyato, and C. Yuen, “Bayesian reinforcement learning for energy harvesting communication systems with uncertainty,” in *Proc. of the IEEE ICC Conference*, London, UK, Jun. 2015.

[11] 3GPP, “Telecommunication management; network sharing; concepts and requirements,” 3GPP TS 32.130, Jun. 2016.

[12] ——, “Network sharing; artechecture and functional description,” 3GPP TR 23.251, Jun. 2016.

[13] M. Vaezi and Y. Zhang, *Virtualization and Cloud Computing*. Springer, 2017, pp. 11–31. [Online]. Available: https://doi.org/10.1007/978-3-319-54496-0_2

[14] NGMN Alliance, “Description of network slicing concept,” Sep. 2016. [Online]. Available: https://www.ngmn.org/uploads/media/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf

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

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

[17] ONF, “Applying SDN architecture to 5G slicing issue 1,” ONF TR-526, Apr. 2016.

[18] ——, “SDN architecture issue 1.1,” ONF TR-521, 2016.

[19] C. Do, N. Tran, C. Pham, M. Alam, J. H. Son, and C. S. Hong, “A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing,” in *Proc. of the IEEE ICOIN Conference*, Jan Siem Reap, Cambodia, Jan. 2015, pp. 324–329.

[20] M. Aazam and E.-N. Huh, “Dynamic resource provisioning through fog micro datacenter,” in *Proc. of the IEEE PerCom Workshops*, St. Louis, MO, Mar. 2015, pp. 105–110.

[21] S. Sarkar, S. Chatterjee, and S. Misra, “Assessment of the suitability of fog computing in the context of internet of things,” *to appear at IEEE Transactions on Cloud Computing*. [Online]. Available: http://ieeexplore.ieee.org/document/7286781/

[22] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, ch. Fog Computing: A Platform for Internet of Things and Analytics, pp. 169–186.

[23] S. Datta, C. Bonnet, and J. Haerri, “Fog computing architecture to enable consumer centric internet of things services,” in *Proc. of IEEE ISCE Conference*, Madrid, Spain, Jun. 2015, pp. 1–2.

[24] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *Proc. of the IEEE INFOCOM Conference*, Orlando, FL, Mar. 2012, pp. 945–953.

[25] T. Truong-Huu, C.-K. Tham, and D. Niyato, “To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud,” in *Proc. of IEEE CloudCom*, Singapore, Dec. 2014, pp. 182–189.

[26] E. Cuervo, A. Balasubramanian, D. ki Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: Making smartphones last longer with code offload,” in *Proc. of the ACM MobiSys*, San Francisco, CA, Jun. 2010.

[27] D. Niyato, P. Wang, P. Joo, Z. Han, and D. I. Kim, “Optimal energy management policy of a mobile cloudlet with wireless energy charging,” in *Proc. of the IEEE SmartGridComm*, Venice, Italy, Nov. 2014.

[28] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, “Fog computing in multi-tier data center networks: A hierarchical game approach,” in *Proc. of the IEEE ICC Conference*, Kuala Lumpur, Malaysia, May 2016.

[29] Y. Zhang, D. Niyato, and P. Wang, “An auction mechanism for resource allocation in mobile cloud computing systems,” in *Proc. of the WASA Conference*, Zhangjiajie, China, Aug. 2013.

[30] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, “A framework for cooperative resource management in mobile cloud computing,” *IEEE J. Sel. Areas in Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[31] M. Chiang, B. Balasubramanian, and F. Bonomi, *Fog for 5G and IoT*. John Wiley & Sons, 2017.

[32] 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, 2017.

[33] H. Zhang, Y. Xiao, S. Bu, R. Yu, D. Niyato, and Z. Han, “Distributed resource allocation for data center networks: A hierarchical game approach,” *to appear at IEEE Transactions on Cloud Computing*.

[34] L. Tong, Y. Li, and W. Gao, “A hierarchical edge cloud architecture for mobile computing,” in *Proc. of IEEE INFOCOM Conf.*, San Francisco, CA, Apr. 2016.

[35] I. Hadžić, Y. Abe, and H. C. Woithe, “Edge computing in the epc: A reality check,” in *Proc. of the ACM/IEEE Symposium on Edge Computing*. New York, NY, USA: ACM, 2017, pp. 13:1–13:10. [Online]. Available: http://doi.acm.org/10.1145/3132211.3134449

[36] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, “Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, May 2017.

[37] E. G. N. 005, “Network functions virtualisation (NFV); ecosystem; report on sdn usage in NFV architectural framework,” v. 1.1.1, Dec. 2015.

[38] V. Sciancalepore, K. Samdanis, and X. Costa-Perez, “Mobile traffic forecasting for maximizing 5g network slicing resource utilization,” in *Proc. of the IEEE INFOCOM Conference*, Atlanta, GA, May 2017.

[39] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez, “Network slicing games: Enabling customization in multi-tenant networks,” in *Proc. of the IEEE INFOCOM Conference*, Atlanta, GA, May 2017.

[40] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, “Optimising 5g infrastructure markets: The business of network slicing,” in *Proc. of the IEEE INFOCOM Conference*, Atlanta, GA, May 2017.

[41] “The cloud comes to you: AT&T to power self-driving cars, AR/VR and other future 5G applications through edge computing,” AT&T Newsroom, Jul. 2017. [Online]. Available: http://about.att.com/story/reinventing_the_cloud_through_edge_computing.html

[42] “Problem statement: Transport support for augmented and virtual reality applications,” Internet-Draft, IETF, Mar. 2017. [Online]. Available: https://tools.ietf.org/id/draft-han-iccrg-arvr-transport-problem-00.xml

[43] A. Aprem, C. Murthy, and N. Mehta, “Transmit power control policies for energy harvesting sensors with retransmissions,” *IEEE J. Sel. Topics in Signal Process.*, vol. 7, no. 5, pp. 895–906, Oct. 2013.

[44] M. Keller and H. Karl, "Response time-optimized distributed cloud resource allocation," *arXiv preprint arXiv:1601.06262*, 2016. [Online]. Available: http://arxiv.org/abs/1601.06262

[45] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, Dec 2016.

[46] Y. Xiao, Z. Han, and L. A. DaSilva, "Opportunistic relay selection for cooperative energy harvesting communication networks," in *Proc. of the IEEE GLOBECOM*, Austin, TX, Dec. 2014.

[47] Y. K. Chia, S. Sun, and R. Zhang, "Energy cooperation in cellular networks with renewable powered base stations," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 6996–7010, Dec 2014.

[48] Q. Li, Y. Wei, M. Song, and F. R. Yu, "Traffic aware energy management in cellular networks with renewable energy powered base stations," in *Proc. of the IEEE VTC Spring Conference*, May 2016, pp. 1–5.

[49] L. Zhang, S. Ren, C. Wu, and Z. Li, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *Proc. of the IEEE INFOCOM Conference*, Hong Kong, China, Apr. 2015, pp. 2632–2640.

[50] N. Tran, C. Do, S. Ren, Z. Han, and C. S. Hong, "Incentive mechanisms for economic and emergency demand responses of colocation datacenters," *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 12, pp. 2892–2905, Dec. 2015.

[51] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proc. of the IEEE INFOCOM Conference*, Atlanta, GA, May 2017.

[52] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. R. Jennings, "Cooperative games with overlapping coalitions," *Journal of Artificial Intelligence Research*, vol. 39, no. 1, pp. 179–216, Sep. 2010.

[53] Y. Xiao, D. Niyato, Z. Han, and L. A. DaSilva, "Joint optimization for power scheduling and transfer in energy harvesting communication systems," in *Proc. of the IEEE GLOBECOM Conference*, San Diego, CA, Dec. 2015.

[54] G. Chalkiadakis and C. Boutilier, "Sequentially optimal repeated coalition formation under uncertainty," *Autonomous Agents and Multi-Agent Systems*, vol. 24, no. 3, pp. 441–484, May 2012.

[55] P. Gmytrasiewicz and P. Doshi, "A framework for sequential planning in multiagent settings," *Journal of Artificial Intelligence Research*, vol. 24, no. 1, pp. 49–79, Jul. 2005.

[56] M. Chiang, "Fog networking: An overview on research opportunities," *arXiv preprint arXiv:1601.00835*. [Online]. Available: http://arxiv.org/pdf/1601.00835

[57] P. D. Francesco, F. Malandrino, and L. A. DaSilva, "Assembling and using a cellular dataset for mobile network analysis and planning," *to appear at IEEE Transactions on Big Data*, 2018.

[58] J. Kibilda, B. Galkin, and L. A. DaSilva, "Modelling multi-operator base station deployment patterns in cellular networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 3087–3099, Dec 2016.

[59] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics, 2005.