Stackelberg Game for Service Deployment of IoT-Enabled Applications in 6G-aware Fog Networks

Abhishek Hazra, Mainak Adhikari, Tarachand Amgoth, and Satish Narayana Srirama

***Abstract*—Fog computing has emerged as a promising paradigm that borrows the user-oriented cloud services to the proximity of the Internet-of-Things (IoT) users in sixth-generation (6G) networks. Currently, service providers establish a proprietary fog architecture to prolong a specific group of IoT users by offering resources and services to the edge level. However, this sort of activity creates a service barrier and limits the development of fog services to the IoT-users. Keeping this in mind, we develop a 6G-aware fog federation model for utilizing maximum fog resources and providing demand specific services across the network while maximizing the revenue of fog service providers and guaranteeing the minimum service delay and price for IoT-users. To achieve this goal, we formulate our objective function into a mixed-integer nonlinear problem. By jointly optimizing the dynamic services cost and user demands, a non-cooperative Stackelberg game interaction algorithm is formulated to schedule the fog and cloud resources distributively. Further maximizing the profit for the service providers and the seamless resource provisioning, a resource controller is initiated to manage the available fog resources. Extensive simulation analysis over 6G-aware Quality-of-Service parameters demonstrates the superiority of the proposed fog federation model and it reduces up to 15%-20% service delay and 20%-25% of service cost over the standalone fog and cloud frameworks.**

***Index Terms*—Internet-of-Things, Fog federation, cost optimization, service deployment, Stackelberg game, 6G networks.**

I. Introduction

THE emergence of revolutionary Internet-of-Things (IoT) is expected to offer new service paradigms related to smart city, smart grid [1], healthcare, intelligent transportation, rural area coverage, etc. and provides faster and secure data processing to the IoT-users [2], [3]. All such IoT applications demand ultra-fast communication (such as beyond 5G or 6G technology) and collaborative fog computation capabilities to breathe appropriately in the network. The main idea of the distributed fog computing is to bring the cloud services to the edge of the network using fog storage and 6G communication, which can exhibit cloud processing power and reduce the long latency. With successful development, fog computing and 6G communication technology can potentially bring new opportunities to network operators, cloud service providers, and heterogeneous IoT-users, where multiple service providers and network operators collaboratively handle the users' requests [4]. On the other hand, IoT-users can experience excellent Quality-of-Services (QoS) parameters such as high data rates, seamless network coverage, interoperable connectivity, and new services, hence, increase the user's satisfaction ratio. Nevertheless, compared with existing cloud infrastructure, fog computing still suffers from limited storage and processing with higher infrastructure and maintenance costs [5].

To keep the above-mentioned challenges in mind, a new collaborative service deployment strategy is developed, where the overloaded fog devices in the network can share the excessive workload with the nearby underloaded fog devices, is also called as fog federation model. Note that fog federation is a trusted consortium, which is authorized to control the resources of various fog service providers. Fog federation aims to realize the targets of scalability, low-latency, and cost-efficient platform by seamlessly integrating fog-cloud resources and ultra-reliable 6G communication technology. From the IoT-users' point of view, the key advantage of the fog federation model is to handle the user's dynamic service requests efficiently while optimizing the service cost and delay, as depicted in Fig. 1. Further, from the service providers point of view, the main advantage of the fog federation model over the traditional fog model is that the fog federation model unions the infrastructures of different fog service providers to improve the service performance by optimizing the service deployment strategy.

A. Related Study and Scope

Over the last few years, several efforts are made to address scheduling, offloading, and resource provisioning related issues in the fog-cloud frameworks [6]. In this regard, an obvious optimization solution is to distribute the user's workload to the nearby fog devices or resource-rich cloud server [7]. In [8], Mohamed *et al.* have designed a marine predators-based binary task offloading framework in a fog-cloud environment for minimizing energy. Similarly, Li *et*

Manuscript received October XX, 2020; revised January XX, 2020; January XX, 2020; accepted February XX, 2020. Date of publication February XX, 2020; date of current version July XX, 2020. *(Corresponding author: Tarachand Amgoth.)*

A. Hazra and T. Amgoth are with the Indian Institute of Technology(Indian School of Mines) Dhanbad, Jharkhand, India, 826004. (e-mail: abhishek-hazra.18DR0018@cse.iitism.ac.in, tarachand@iitism.ac.in).

M. Adhikari is with the Mobile & Cloud Lab, Institute of Computer Science, University of Tartu, Estonia (e-mail: mainak.ism@gmail.com).

S. N. Srirama is with the School of Computer and Information Sciences, University of Hyderabad, India (e-mail: satish.srirama@uohyd.ac.in).

Digital Object Identifier 10.1109/JTOT.2020.3021XXX

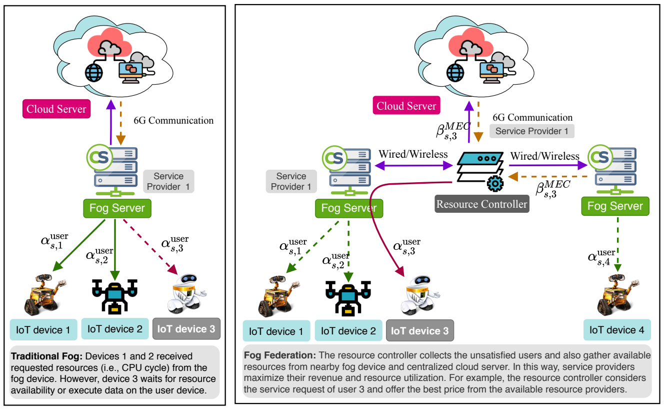


Fig. 1. Comparison between traditional fog model and fog federation model.

al. [9] have designed a workload distribution strategy in small cell networks. However, binary data offloading strategy might not be suitable for latency-critical applications (e.g., healthcare and industrial applications). Another solution could be vertical resource sharing, where computing devices distribute data partially to the nearby fog devices [10]. The fog devices also share some amount of data to the centralized cloud server for further processing and storage, also known as partial offloading. In [11], Mukherjee *et al.* have designed a partial computation offloading strategy for delay-sensitive fog networks. In [12]–[14], authors mostly considered several resource management techniques for maximizing the service revenue of the fog-cloud frameworks.

Existing efforts mostly concentrate on standalone fog/cloud service provisioning mechanisms for IoT-users and also not suitable for large-scale data processing. However, future generation applications will be mostly on multimedia data, and a large amount of this data can be 4K or 8K video streaming data. Moreover, those applications demand for deadline and energy-aware resource provisioning mechanisms, which makes the situation more complicated [15]. Therefore, a decent fog federated network with an efficient service deployment strategy should be adopted in order to satisfy the QoS parameters of the IoT-users. That helps to share the computing resources among the multiple service providers collaboratively for maximizing service revenue [16]. However, designing a fog federated model is not an easy process due to the users' and network dynamicity. There are three critical challenges for 6G enabled fog federation network: Firstly, *how to design an efficient, scalable, and low-latency fog federation model for future 6G communication?* Secondly, *how to model an efficient service provisioning strategy across the fog, cloud, and IoT-users?* Finally, *how can an optimization technique reduce the cost-performance trade-off and simultaneously satisfy both user and service provider's requirements?*

Among the discussed literature, a handful of research activity is to enhance the traditional fog model and utilize the game-theoretic approach for minimizing service costs of IoT-users and maximizing the revenue of service providers [17], [18]. Fog federation model is a distributed umbrella, where multiple IoT-users request their desired services, and numerous service providers can collaboratively work under the same umbrella for maximizing their profits. Considering the above challenges in mind, we design a 6G-aware fog federation model using a game strategy to establish a cost-efficient platform for IoT-users with a transparent resource management policy by seamlessly integrating the distributed fog devices and centralized cloud servers. Further, to provide services to more number of IoT-users and maximize the revenue, a resource controlled mechanism is incorporated for collecting the current state information of the active set of devices in the 6G-enabled network.

B. Contribution

In this paper, we develop a fog federation model with the Stackelberg game strategy for optimizing cost-performance trade-off by integrating service deployment strategy in the 6G-enabled fog networks. The significant contributions of this paper are summarized as follows.

- Design a novel fog federation model for utilizing computing resources efficiently and obtaining maximum service revenue in a 6G coupled fog networks. In particular, a service deployment strategy is formulated as a mixed-integer nonlinear programming problem under opportunistic service costs and available resources.
- To achieve the cost-performance trade-off, we transform our objective function into a Stackelberg game problem and optimize the network resources and service cost of the IoT-users. Further, a resource controller is integrated to monitor user's dynamic requests and simultaneously schedule the fog resources in such a way that both user and service providers can obtain maximum benefit.
- Extensive simulation with emerging 6G-aware parameters demonstrates that the proposed fog federation model encompasses near-optimal solutions and outperforms than the traditional fog and cloud frameworks.

The rest of the paper is structured as follows. In section II, the system model and problem formulation are presented. The proposed strategy on the fog federation model is presented in section III. The performance analysis of the proposed strategy over baseline algorithms is performed in section IV. Finally, the conclusion and future direction of the work are discussed in section V.

II. System Model and Problem Formulation

In this section, we discuss the system model of the proposed fog federation model followed by the problem formulation.

A. System Model

The proposed fog federation model is depicted in Fig. 2, where a set of service providers $\mathcal{P} = \{1, 2, \dots, P\}$ offer a set of services $\mathcal{S} = \{1, 2, \dots, S\}$ and a set of IoT-users utilize those services simultaneously. Let us consider that the IoT-user, fog device, and cloud server are the three key components in this network. Further, we consider that a set of IoT-users $\mathcal{U} = \{1, 2, \dots, U\}$ and fog devices $\mathcal{F} = \{1, 2, \dots, F\}$ are randomly and uniformly distributed over the small-cell network [19]. To handle the excessive workload from the IoT-users, a set of cloud servers $\mathcal{C} = \{1, 2, \ldots, C\}$ are deployed over the geo-distributed locations. Additionally, we consider that an IoT-user u can generate $\mathcal{X}_i^{in}$ amount of data, which can partially offload to the local fog device f for utilizing the resources efficiently. Further, we assume that each IoT-user can request for $\mathcal{Y}_i$ amount of CPU cycle to execute $\mathcal{X}_i^{in}$-bit task. Each IoT-user $u \in U$ can request for multiple services $\mathcal{S}$ from the set of computing devices in the network based on their availability. Further, each computing device $\mathcal{V}_j$ can receive multiple service requests, however, each $\mathcal{V}_j$ can accept at most one request at a time, i.e., $\sum_{s=1}^{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{V}_j|}\lambda_{s,j} \leq |\mathcal{V}_j|$, where $|\cdot|$ denotes cardinality of a set.

Here, we assume a distributed service deployment scenario, where an IoT-user deploys a task to the available computing device $\mathcal{V}_j, \forall j \in (\mathcal{F} \cup \mathcal{C})$ in the network. In the fog federation model, each request generates from uth IoT-user device with input/output data size, represented as $\mathcal{X}_i^{in}/\mathcal{X}_i^{out}$ (in bits). Each service request s is either execute locally on the generated IoT-user device u or requests for service from the available fog device f or the centralized CDC server c.

1) Local Task Processing: In the initial phase, the IoT-users intended to process its requests locally. Let $\mathcal{O}_u^{CPU}$ be the computational capacity [cycle/sec] of the IoT-user. The execution time for processing the tasks on the requesting IoT-user device is defined as follows.

$$T_u^{MD} = \frac{(\mathcal{X}_i^{in} - \widetilde{\mathcal{X}}_i^{in})\mathcal{Y}_i}{\mathcal{O}_u^{CPU}} \quad [s] \tag{1}$$

2) Remote Task Processing: Owing to limited storage and processing capability, IoT-users often transfer the excessive workloads to the service providers (e.g., fog or cloud server) for further processing. Let us assume that $\mathscr{C}_{u,j}$ and P_u^{power} represent the maximum channel gain and the transmission power of the IoT-user device, respectively, where $j \in \{\mathcal{F} \cup \mathcal{C}\}$. Thus, the uploading rate from uth IoT-user device to jth computing device can be expressed as $R_{u,j}^{trans} = \mathcal{B}_{u,j} log_2\left(1 + \frac{P_u^{power}\mathscr{C}_{u,j}}{\mathcal{E}_0}\right)$, where $\mathcal{B}_{u,j}$ be the transmission bandwidth and $\mathcal{E}_0$ be the additive Gaussian noise of IoT-user device u. Therefore, the transmission time from an IoT-user to the selected computing device j is expressed as follows.

$$T_j^{UL} = \frac{\widetilde{\mathcal{X}}_i^{in}}{R_{u,j}^{trans}} \quad [s] : j \in \{\mathcal{F} \cup \mathcal{C}\} \tag{2}$$

Hence, the service time in a computing device is defined as follows.

$$T_j^{MEC} = \frac{\widetilde{\mathcal{X}}_i^{in}\mathcal{Y}_i}{\mathcal{O}_j^{CPU}} \quad [s] : j \in \{\mathcal{F} \cup \mathcal{C}\} \tag{3}$$

Where $\mathcal{O}_j^{CPU}$ represents the computational capacity of the jth computing device. Similarly, $\mathscr{C}_{j,u}$ be the channel gain and P_j^{power} be the transmission power of a computing device, where $j \in \{\mathcal{F} \cup \mathcal{C}\}$. Thus, the downloading data transmission rate from jth fog device to uth IoT-user device can be expressed as $R_{j,u}^{trans} = \mathcal{B}_{j,u} log_2\left(1 + \frac{P_j^{power}\mathscr{C}_{j,u}}{\mathcal{E}_0}\right)$, where $\mathcal{B}_{j,u}$ be the transmission bandwidth and $\mathcal{E}_0$ be the additive white Gaussian noise of jth computing device. Therefore,

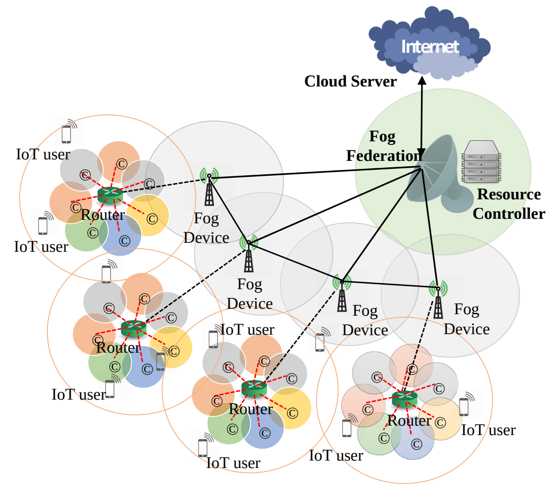


Fig. 2. Illustration of fog federation architecture.

the downloading time from the jth computing device to the requested IoT-user device is expressed as follows.

$$T_j^{DL} = \frac{\widetilde{\mathcal{X}}_i^{out}}{R_{j,u}^{trans}} \quad [s] : j \in \{\mathcal{F} \cup \mathcal{C}\} \tag{4}$$

As expected, 6G-enable fog devices can be capable of reducing end-to-end delay up to $< .001 - .01$ ms. Thus, we simply neglect the communication delay among fog devices inside fog federation network.

3) Overall Service Delay: The service delay is a critical QoS parameter of the proposed model and mostly depends on several network and device-dependent parameters. As per this model, we mainly consider delay related to transmitting (uploading and downloading) and processing of service requests. Thus, the total processing delay $(T_{s,u}^{service})$ to complete a user's service request on the selected computing device can be expressed as follows.

$$T_{s,u}^{service} = \left(\frac{\widetilde{\mathcal{X}}_i^{in}}{R_{u,j}^{trans}} + \frac{\widetilde{\mathcal{X}}_i^{in}\mathcal{Y}_i}{\mathcal{O}_j^{CPU}} + \frac{\widetilde{\mathcal{X}}_i^{out}}{R_{j,u}^{trans}}\right) \tag{5}$$

In the 6G based fog federation model, IoT-users QoS must be considered to offer a guaranteed Quality of Experience (QoE) parameter. Let us considered that $T_{s,u}^{max}$ be the maximum latency demand to accomplish service s. Now, service providers can easily monitor accomplished users demand using a satisfaction parameter, which is expressed as follows.

$$\Gamma_{s,u}^{MEC} = \begin{cases} 1, & \text{if } T_{s,u}^{service} \leq T_{s,u}^{max} \\ 0, & \text{Otherwise} \end{cases} \tag{6}$$

Moreover, the providers also need to keep track of the IoT-users satisfaction ratio to establish a high-level business model [20], which can be expresses as follows.

$$\mathcal{R}_{s,u}^{satis} = \frac{\sum_{j\in\mathcal{V}_j}\sum_{s\in S}\sum_{u\in U}\widetilde{\mathcal{X}}_i^{in}\Gamma_{s,u}^{MEC}}{\sum_{j\in\mathcal{V}_j}\sum_{s\in S}\sum_{u\in U}\widetilde{\mathcal{X}}_i^{in}} \tag{7}$$

Eq. (7) determines that the satisfaction ratio $l_1 \leq \mathcal{R}_{s,u}^{com} \leq l_2$.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2020.3041102

4) Cost for processing in Fog Federation Model: The fog federation model considers the depreciation of the total cost (i.e., revenue maximization) of service provider, which is an essential evaluation criterion to fulfill a group of IoT-users' requirements. Let $\alpha_{s,u}^{user}$ represents the cost co-efficient for the IoT-user u to process request s locally, where $0 \leq \alpha_{s,u}^{user} \leq 1$. Therefore, the computation cost to process $\mathcal{X}_i^{in}$ amount of task locally is defined as follows.

$$C_{s,u}^{MD}\left(\alpha_{s,u}^{user}, \mathcal{X}_i^{in}\right) \triangleq \sum_{u\in U, s\in S} \underbrace{\alpha_{s,u}^{user}\mathcal{X}_i^{in}\mathcal{Y}_i}_{\text{Processing cost}} \tag{8}$$

However, if an IoT-user device u requests a service s from the remote computing devices, then the overall cost $C_{s,u}^{MEC}$ at the service provider should include storage cost, processing cost, transmission cost and infrastructure maintenance cost for processing the requested task. Further, we define four cost coefficients $\beta_{s,j}^{MEC} = \langle \beta_{s,j}^{stor}, \beta_{s,j}^{main}, \beta_{s,j}^{proc}, \beta_{s,j}^{trans} \rangle$ for per storage unit, where $0 \leq \beta_{s,j}^{MEC} \leq 1, \forall j \in \mathcal{V}_j, s \in S$. Therefore, the service cost at the remote computing device can be expressed as follows.

$$\begin{aligned} C_{s,j}^{MEC}\left(\beta_{s,j}^{MEC}, \widetilde{\mathcal{X}}_i^{in}\right) \triangleq & \sum_{s\in S} \underbrace{\beta_{s,j}^{stor}\widetilde{\mathcal{X}}_i^{in}}_{\text{Storage cost}} + \sum_{s\in S} \underbrace{\beta_{s,j}^{main}\widetilde{\mathcal{X}}_i^{in}}_{\text{Maintenance cost}} \\ & + \sum_{s\in S} \underbrace{\beta_{s,j}^{proc}\widetilde{\mathcal{X}}_i^{in}\mathcal{Y}_i}_{\text{Processing cost}} + \sum_{s\in S} \underbrace{\beta_{s,j}^{tran}\widetilde{\mathcal{X}}_i^{in}}_{\text{Transmission cost}} \end{aligned} \tag{9}$$

Further, it is important to note that the processing and storage demand should not be exceed the maximum capacity of the selected computing device (i.e., $\beta_{s,j}^{p\_max}$, and $\beta_{s,j}^{s\_max}$), which can be determined as follows.

$$\sum_{s\in S}\sum_{j\in\mathcal{V}_j} \widetilde{\mathcal{X}}_i^{in}\beta_{s,j}^{proc}\mathcal{Y}_i \leq \beta_{s,j}^{p\_max}, \quad \forall j \in \{\mathcal{F}\cup\mathcal{C}\} \tag{10}$$

$$\sum_{s\in S}\sum_{j\in\mathcal{V}_j} \widetilde{\mathcal{X}}_i^{in}\beta_{s,j}^{stor} \leq \beta_{s,j}^{s\_max}, \quad \forall j \in \{\mathcal{F}\cup\mathcal{C}\} \tag{11}$$

Ideally, all the service providers and IoT-user devices strictly follow the Eq. (10) and Eq. (11) to reduce latency, cost and computation overhead.

B. Problem Formulation

This work aims to design a 6G-enabled fog federation model for leading network resources and provide demand-specific assistance across the network. Since, revenue maximization has been a critical business concern and the crucial bottleneck of IoT-users, we design the objective function for service deployment as the optimization of the service cost for both the IoT-user and service provider while guaranteeing minimum service delay in the proposed fog federation model. Therefore, we define two objectives for service deployment of the proposed fog federation model, which are defined as follows.

Objective 1: *Minimize service cost for IoT-user.*
Objective 2: *Maximize revenue for service provider.*

Mathematically, the users objective function is expressed as follows (users problem).

$$\mathbf{P1:} \quad \sum_{t\in T} \min_{\mathcal{X}_i^{in}, \alpha_{s,u}^{user}} \left(\sum_{s\in S}\sum_{u\in U} C_{s,u}^{MD}\left(\alpha_{s,u}^{user}, \mathcal{X}_i^{in}\right)(t) \right)$$

Similarly, the service provider objective function is derived as follows (service providers problem).

$$\mathbf{P2:} \quad \sum_{t\in T} \max_{\widetilde{\mathcal{X}}_i^{in}, \beta_{s,j}^{MEC}} \left(\sum_{s\in S}\sum_{j\in\mathcal{V}_j} C_{s,j}^{MEC}\left(\beta_{s,j}^{MEC}, \widetilde{\mathcal{X}}_i^{in}\right)(t) \right)$$

Now, we derive the dependencies and related constraints of two objective functions. The proposed objectives should be optimized concerning maximum delay constraints. Moreover, other storage and processing related constraints are defined in Eq. (7), Eq. (10), and Eq. (11) respectively, which need to be granted for reducing workload in the federated fog model.

III. Proposed Service Deployment strategy in Fog Federation Model

This section discusses the proposed fog federation model for efficient service deployment while minimizing the users service cost and maximizing revenue of service providers. The proposed service deployment strategy is divided into two subsections. Firstly, we reformulate **P1** and **P2** objective functions, namely **P3** and **P4** into the form of a two-player Stackelberg game. Next, we discuss the proposed service deployment strategy for optimizing **P3** and **P4** functions.

A. Game Formulation

In the proposed fog federation model, the IoT-user devices $\mathcal{U}$ intend to process their generated data $\mathcal{X}_i^{in}$ locally. However, in complex application scenarios (e.g., AR, VR, and 4K or 8K video streaming, etc.), each IoT-user device u requests additional assistance from the service provider $p \in P$ to reduce the computation overhead T_u^{MD}. In this process, IoT-user u seeks to minimize the service cost from the provider p while customarily maximizing the revenue of the service provider p within the maximum resource constraints (i.e., $\beta_{s,j}^{p\_max}$ and $\beta_{s,j}^{s\_max}$). A resource controller is usually devised to regulate the trade-off between dynamic request demand s and balances service deployment strategy between providers and consumers, which essentially contains the IoT-users' requests according to the availability of the resources and prices. Thus, a Stackelberg game can be designed to capture the interaction between the service provider $\mathcal{P}$ and consumer $\mathcal{U}$ [21]. Stackelberg game is a non-cooperative game theory, where providers initiate the game by setting the best price/resource, and IoT-users $\mathcal{U}$ negotiate for the best price by observing the provider's price rule. This process continues until the IoT-users and service providers reach an equilibrium state.

Initially, the service providers take initiative advantage and maximize their prices by offering a service as a form of computing resources to the requested IoT-user. Mathematically,

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JIOT.2020.3041102

the service provider optimization problem (i.e., **P2**) can be reformulated as follows.

$$\begin{aligned}\mathbf{P3}: &\sum_{s\in S}\left(\max_{\widetilde{\mathcal{X}}_i^{in},\beta_{s,j}^{MEC}}\sum_{j\in\mathcal{V}_j} C_{s,j}^{MEC}\left(\beta_{s,j}^{MEC},\widetilde{\mathcal{X}}_i^{in}\right)\right)\\ \text{s.t.} &\sum_{s\in S}\sum_{j\in\mathcal{V}_j}\widetilde{\mathcal{X}}_i^{in}\beta_{s,j}^{proc}\mathcal{Y}_i\le\beta_{s,j}^{p\_max},\quad\forall s\in S\\ &\sum_{s\in S}\sum_{j\in\mathcal{V}_j}\widetilde{\mathcal{X}}_i^{in}\beta_{s,j}^{stor}\le\beta_{s,j}^{s\_max},\quad\forall s\in S\\ &T_{s,u}^{service}\le T_{s,u}^{max},\quad\forall s\in S,\quad\forall u\in U\end{aligned}\tag{12}$$

It is noteworthy that service providers $\mathcal{P}$ usually devised their CPU cycle $\mathcal{O}_j^{CPU}$ to the requested IoT-users CPU cycle, i.e., $\mathcal{O}_j^{CPU}/\mathcal{U}$ to achieve maximum benefit $C_{s,j}^{MEC}$. Whenever an IoT-user u requests a new service s, the IoT-user u needs to bear the service cost plus the delay $\mathcal{T}_{s,u}^{user}$ incurred in this process [22]. Similarly, the user's minimization problem can be interpreted as follows.

$$\begin{aligned}C_{s,u}^{MD}\left(\alpha_{s,u}^{user},\mathcal{X}_i^{in}\right)=&\sum_{s\in S}\sum_{u\in U}\alpha_{s,u}^{user}\mathcal{X}_i^{in}\mathcal{Y}_i+\mathcal{T}_{s,u}^{user}\\ &-\sum_{s\in S}\sum_{u\in U}\alpha_{s,u}^{user}\widetilde{\mathcal{X}}_i^{in}\mathcal{Y}_i\end{aligned}\tag{13}$$

Further, the cost in the IoT-user device is equivalent to $C_{s,u}^{MD}(\alpha_{s,u}^{user},\mathcal{X}_i^{in}) = \widetilde{\mathcal{X}}_i^{in}(\beta_{s,j}^{stor}+\beta_{s,j}^{main}+\beta_{s,j}^{proc}\mathcal{Y}_i+\beta_{s,j}^{trans})+\frac{(\mathcal{X}_i^{in}-\widetilde{\mathcal{X}}_i^{in})\mathcal{Y}_i}{\mathcal{O}_u^{CPU}}$: if $0\le\widetilde{\mathcal{X}}_i^{in}\le m_{s,u}^{user}$ and $C_{s,u}^{MD}(\alpha_{s,u}^{user},\mathcal{X}_i^{in}) = \widetilde{\mathcal{X}}_i^{in}(\beta_{s,j}^{stor}+\beta_{s,j}^{main}+\beta_{s,j}^{proc}\mathcal{Y}_i+\beta_{s,j}^{trans})+\widetilde{\mathcal{X}}_i^{in}\mathscr{B}_{s,u}^{MEC}$: if $m_{s,u}^{user}\le\widetilde{\mathcal{X}}_i^{in}\le\mathcal{X}_i^{in}$, where $\mathscr{B}_{s,u}^{MEC}=\frac{1}{R_{u,j}^{trans}}+\frac{\mathcal{Y}_i}{\mathcal{O}_j^{CPU}}+\frac{1}{R_{j,u}^{trans}}$. Here, $m_{s,u}^{user}$ is defined as $m_{s,u}^{user}=\frac{\mathcal{X}_i^{in}\mathcal{Y}_i}{\mathscr{B}_{s,u}^{MEC}\mathcal{O}_u^{CPU}+\mathcal{Y}_i}$.

Each IoT-user u intends to overcome his/her own cost $C_{s,u}^{MD}$ by selecting the optimal data size $\widetilde{\mathcal{X}}_i^{in}$, which is set by the service provider for the given price $\alpha_{s,u}^{user}$. Mathematically, the optimization problem of the IoT-user (i.e., **P1**) can be reformulated as follows.

$$\mathbf{P4}:\quad\sum_{s\in S}\left(\min_{\alpha_{s,u}^{user},\mathcal{X}_i^{in}}\sum_{u\in U}C_{s,u}^{MD}\left(\alpha_{s,u}^{user},\mathcal{X}_i^{in}\right)\right)\tag{14}$$

Where $0\le\widetilde{\mathcal{X}}_i^{in}\le\mathcal{X}_i^{in}$. In the Stackelberg game theory, objective functions **P3** and **P4** are complicatedly combined, i.e., pricing policy in the fog federation model affects the users' unloaded data sizes, which also impacts the service revenues [23].

B. *Proposed Service Deployment Strategy*

In this section, we develop the proposed two-player Stackelberg game approach for efficient service deployment on the fog federation model, where both the players (i.e., IoT-users and service providers) try to maximize their benefits and reach in an equilibrium state [24]. The proposed strategy follows two steps for service deployment on the fog networks. Firstly, the distributed fog devices control the IoT-users requests and process the requested tasks as standalone resource supervisors. For this resolution, a delay-based service deployment strategy is formulated using a *multi-user single-provider* Stackelberg game theory, where IoT-user devices (multi-user) requests for resources in a distributed manner, and each fog device (single-provider) provides services and allocates the requested resources along with the price.

However, due to the limited resource capacity and battery power, a single fog device might not handle all IoT-user requests completely. In such situations, the fog device needs to interact with other fog devices or centralized cloud server for further processing. Thus, *single-user multi-provider* game strategy comes into action, where a resource controller takes leadership and gathers available resources of the active fog devices on the network. The resource controller offers the least-cost resources for processing the tasks on the suitable fog devices while maximizing the provider's revenue [25]. In this scheme, the resource controller (single-user) uses the same Stackelberg game theory for finding suitable computing devices (multi-provider) for further processing the failed/partially postponed tasks.

1) Multi-User Single-Provider Stackelberg Game: In this stage, the proposed fog federated model follows differential pricing rules (as depicted in [22]) to maximize the satisfaction ratio of the fog device $f\in F$ while considering maximum tolerable delay as a specifiable IoT-user parameter. The IoT-users take the initial decisions and request for services for processing the tasks. After receiving the service requests, the local fog devices play a second role in the model. Each fog device calculates the overall cost of each requested resource and announces the price. Using the declared price, IoT-users calculate their expenditure and decides the optimal decision of whether to obtain services with this cost or not. Let, $\lambda_{s,j}=\{0,1\}$ be a service indicator where $\lambda_{s,j}=1$ if sth service request is accepted by $\mathcal{F}_j$ or 0 otherwise, and the IoT-user takes one service only if $\alpha_{s,u}^{user}>1/\mathcal{O}_u^{CPU}$ [23]. Therefore, the revenue maximization problem of the fog service provider can be expressed as follows.

$$\begin{aligned}\mathbf{P5}: &\sum_{j\in\mathcal{V}_j}\Bigg(\max_{\lambda_{s,j}\in\{0,1\}}\sum_{s\in S}\frac{\lambda_{s,j}\mathcal{Y}_i m_{s,u}^{user}}{\mathcal{O}_u^{CPU}}+\sum_{s\in S}\beta_{s,j}^{stor}m_{s,u}^{user}+\\ &\qquad\sum_{s\in S}\beta_{s,j}^{main}m_{s,u}^{user}+\sum_{s\in S}\beta_{s,j}^{tran}m_{s,u}^{user}\Bigg)\\ \text{s.t.} &\sum_{s\in S}\sum_{j\in\mathcal{V}_j}\lambda_{s,j}m_{s,u}^{user}\le\beta_{s,u}^{s\_max},\ \forall s\in S\\ &\sum_{s\in S}\sum_{j\in\mathcal{V}_j}\lambda_{s,j}\mathcal{Y}_i m_{s,u}^{user}\le\beta_{s,u}^{p\_max},\ \forall s\in S\\ &\sum_{s\in S}T_{s,u}^{service}\le T_{s,u}^{max},\quad\forall s\in S,\quad\forall u\in U\end{aligned}\tag{15}$$

Where $m_{s,u}^{user}\mathcal{Y}_i$ denotes the maximum process requirement of service request s and $\alpha_{s,u}^{p\_max}$ represents the maximum resource availability on the IoT-user device u.

2) Single-User Multi-Provider Stackelberg Game: Owing to resource constraint $\beta_{s,u}^{p\_max}$, some IoT-users in the fog federated model can not receive services earlier from the fog devices. For this purpose, *single-user multi-provider* Stackelberg game is formulated to capture the cost-delay trade-off in the network [26]. In this stage, the resource controller (single-user) monitors the current state of the services on the active fog devices and maximizes the revenue by accumulating available resources from other sets of computing devices (multi-user) in the federated model [23]. The resource controller monitors the computing devices and finds the availability of the computing resources, which is defined as follows.

$$\mathcal{O}^{RM} = \sum_{s \in S} \sum_{j \in \mathcal{V}_j} (1 - \lambda_{s,j}) \frac{\mathcal{O}_j^{CPU}}{\mathcal{U}} \tag{16}$$

$$\mathcal{R}^{RM} = \sum_{s \in S} \left(\beta_{s,j}^{p\_max} - \sum_{u \in U} \lambda_{s,u} \mathcal{Y}_i m_{s,u}^{user} \right) \tag{17}$$

Let $\mathcal{U}^{RM} = \{1, 2, \ldots, U'\}$ be the set of waiting user devices that left the game due to high service cost or resource restrictions from the fog devices. Again, the IoT-user devices start participating with the new pricing system offered by the resource controller, where the number of fog devices is calculated as follows.

$$\mathcal{U}^{RM} = \sum_{s \in S} \sum_{j \in \mathcal{V}_j} (1 - \lambda_{s,j}) \tag{18}$$

Further, the controller calculates the service delay and announces it over the network. Simultaneously, the active set of computing devices in the network calculate their revenues. Once the price is finalized, the computing devices advertise their revenues to the resource controller. Finally, the resource controller calculates the revenue and decides whether to accept it or reject it. Once the resource controller confirms the service, the selected fog device or cloud server can provide services for further processing the stopped/postponed tasks. However, the IoT-users achieve a maximum processing capacity as $\mathcal{O}_u^{MAX} = \mathcal{O}^{RM}/\mathcal{U}^{RM}$ and data size $m_{s,u}^{RM} = \frac{\mathcal{X}_i^{in} \mathcal{Y}_i}{\mathscr{B}_{s,u}^{RM} \mathcal{O}_u^{RM} + \mathcal{Y}_i}$, where $\mathscr{B}_{s,u}^{RM} = \frac{1}{R_{u,j}^{trans}} + \frac{\mathcal{Y}_i}{\mathcal{O}_j^{RM}} + \frac{1}{R_{j,u}^{trans}}$. Let $\lambda_{s,j}^{RM} = \{0, 1\}$ be a service indicator for an IoT-user u while the resource controller controls the network, i.e., $\lambda_{s,j}^{RM}$ is 1 if sth service request is accepted to the jth computing device, otherwise 0. Thus, the revenue maximization problem is further reformulated as follows.

$$\begin{aligned} \textbf{P6}: \sum_{j \in \mathcal{V}_j} \Bigg(\max_{\lambda_{s,j}^{RM} \in \{0,1\}} \sum_{u \in U'} \frac{\lambda_{s,j}^{RM} \mathcal{Y}_i m_{s,u}^{RM}}{\mathcal{O}_u^{CPU}} + \sum_{s \in S} \beta_{s,j}^{stor} m_{s,u}^{RM} + \\ \sum_{s \in S} \beta_{s,j}^{main} m_{s,u}^{RM} + \sum_{s \in S} \beta_{s,j}^{tran} m_{s,u}^{RM} \Bigg) \\ \text{s.t.} \sum_{u \in U'} \sum_{j \in \mathcal{V}_j} m_{s,u}^{RM} \mathcal{Y}_i \beta_{s,j}^{proc} \leq \beta_{s,j}^{p\_max}, \ \forall s \in S \\ \sum_{u \in U'} \sum_{j \in \mathcal{V}_j} m_{s,u}^{RM} \beta_{s,j}^{stor} \leq \beta_{s,j}^{s\_max}, \ \forall s \in S \end{aligned} \tag{19}$$

Where, $\mathcal{O}_f^{CPU} << \mathcal{O}_c^{CPU}$ and $\beta_{s,f}^{stor} << \beta_{s,c}^{stor}$. According to the above methodology, the neighboring fog devices and centralized cloud service providers make maximum revenue in the proposed fog federation model. Moreover, with this strategy, the resource controller can efficiently distribute the workload while meeting the IoT-users expectations and service providers' revenues. The algorithm of the proposed services deployment strategy on the fog federation model is depicted in *Algorithm* 1.

Algorithm 1: `Service Deployment` Strategy

INPUT : $\mathcal{U}$: IoT-users, $\mathcal{F}$: Fog devices, $\mathcal{C}$: Cloud servers.
OUTPUT : Users satisfaction Ratio $\mathcal{R}_{s,u}^{satis}$.
1 **begin**
2 Initialize $\mathcal{O}_j^{CPU}$, $\mathcal{U}^{RM}$, $\mathcal{C}^{RM}$, $\mathcal{R}^{RM}$, $\mathcal{V}_j \leftarrow \{\mathcal{F} \cup \mathcal{C}\}$
3 **if** $|\mathcal{V}_j| = \phi$ **then**
4 Wait for computing device $\mathcal{V}_j$
5 **end**
6 **for** u= *1 to U* **do**
7 Initialize $\alpha_{s,u}^{user}$ and $\beta_{s,j}^{MEC}$
8 Calculate $m_{s,u}^{user} = \frac{\mathcal{X}_i^{in} \mathcal{Y}_i}{\mathscr{B}_{s,u}^{MEC} \mathcal{O}_u^{CPU} + \mathcal{Y}_i}$
9 **if** $\widetilde{\mathcal{X}}_i^{in} \beta_{s,j}^{proc} \mathcal{Y}_i \leq \beta_{s,j}^{p\_max}$ *and* $\widetilde{\mathcal{X}}_i^{in} \beta_{s,j}^{stor} \leq \beta_{s,j}^{s\_max}$ **then**
10 Calculate $\sum_{j \in \mathcal{V}_j} C_{s,j}^{MEC}(\beta_{s,j}^{MEC}, \widetilde{\mathcal{X}}_i^{in})$
11 Update $\alpha_{u,j}^{user}$ and $\beta_{s,j}^{MEC}$
12 Broadcast service cost
13 **if** $C_{s,u}^{MD}(t) \leq C_{s,u}^{MD}(t-1)$ **then**
14 Accept the service s
15 **end**
16 Update $\mathcal{U}^{RM}$ using rejected devices
17 **end**
18 **for** u'= *1 to U'* **do**
19 Control goes to resource controller
20 Calculate $\mathcal{O}^{RM} = \sum_{s \in S} \sum_{j \in \mathcal{V}_j} (1 - \lambda_{s,j}) \frac{\mathcal{O}_j^{CPU}}{\mathcal{U}}$
21 Calculate $\mathcal{R}^{RM} = \sum_{u \in U} (\beta_{s,j}^{p\_max} - \lambda_{s,u} \mathcal{Y}_i m_{s,u}^{user})$
22 Follow Step-9 to Step-17 for resource controller
23 **end**
24 Calculate Satisfaction Ratio $\mathcal{R}_{s,u}^{satis}$
25 **end**
26 **end**

Theorem 1. *The equilibrium strategy of the IoT-users is the best response to the strategic approach that the service providers follow.*

Proof : Let $\alpha_{s,u}^{user}$ be the cost co-efficient for the IoT-user $\mathcal{U}$ and $\beta_{s,u}^{MEC}$ be the cost strategy for resource controller. In the Stackelberg game theory, IoT-users try to minimize their service costs for the requested resources, and the resource controller maximizes their revenue by declaring an optimal price for the requested resources. Thus, the resource controller makes optimal decision with $\alpha_{s,u}^{user^*} \in \alpha_{s,u}^{user}$ if

$$C_{s,u}^{MD}\left(\alpha_{s,u}^{user^*}, \mathcal{X}_i^{in}\right)(t) \leq C_{s,u}^{MD}\left(\alpha_{s,u}^{user^*}, \mathcal{X}_i^{in}\right)(t-1) \tag{20}$$

Theorem 2. *There exist a unique Nash equilibrium between recurring IoT-users and service providers, which holds a unique Stackelberg equilibrium.*

Proof : The proof of Theorem 2 can be found in [13].

Theorem 3. *The computational complexity of the proposed service deployment strategy is $O(mn)$.*

Proof : The calculation of the computational complexity of the proposed service deployment strategy on the fog federation model is two-folded. Let us consider that $m = |\mathcal{U}|$, $n = |\mathcal{F}|$, and $k = |\mathcal{C}|$. In the first phase, the m IoT-user devices request services to n fog service providers. The fog devices calculate the service cost $C_{s,u}^{user}$ and accordingly update the service deployment strategy of m IoT-users with respect to the cost $C_{s,u}^{user}$. Based on the price structure defined by the provider p, the n IoT-users accept or reject the services. This process

TABLE I
PARAMETERS USED FOR SIMULATION

Parameters	Values
Number of end users ($\mathcal{U}$)	100-500
Total number of fog devices ($\mathcal{F}$)	10-50
Number of cloud servers ($\mathcal{C}$)	1-5
Maximum channel bandwidth ($\mathcal{B}$)	30 MHz
User cost coefficient ($\alpha_{s,u}^{user}$)	0.5-0.7
User cost coefficient ($\beta_{s,u}^{MEC}$)	0.3-0.7
CPU frequency in end devices ($\mathcal{O}_u^{CPU}$)	50×10^6 [cyc/sec]
CPU frequency in fog devices ($\mathcal{O}_f^{CPU}$)	50×10^9 [cyc/sec]
CPU frequency in cloud servers ($\mathcal{O}_c^{CPU}$)	100×10^9 [cyc/sec]
Maximum delay threshold ($T_{s,u}^{max}$)	150-250 time unit

takes at most $O(m) \times O(n)$ times, where $1 \leq m \leq |U|$ and $1 \leq n \leq |F|$, as the users are allowed to request multiple fog devices simultaneously.

However due to constraint $T_{s,u}^{max}$, $\beta_{s,u}^{p\_max}$ and $\beta_{s,u}^{s\_max}$, some users $l = |U'|$, where $U' = \mathcal{U} \setminus \{m\}$ might not receive the requests. In such situation, resources controller collects $\mathcal{O}^{RM}$, $\mathcal{R}^{RM}$ and $\mathcal{U}^{RM}$ from the computing devices. Based on the price offered by various service provider $\mathcal{P}$, resource controller offers best price $C_{s,u}^{MEC}$ to the remaining l IoT-users. This process takes $O(l) \times O(n+k) = O(ln + lk)$ time. Thus, the total time complexity of the proposed fog federation model is $O(mn) + O(ln) + O(lk)$. As $mn >> ln >> lk$, thus, the overall time complexity is $O(mn)$.

IV. EXPERIMENTAL ANALYSIS

In this section, we numerically quantify the proposed service deployment strategy of the fog federation model in terms of service delay, revenue generation, and user satisfaction ratio over two existing models such as standalone fog (SF) framework [27] and standalone cloud (SC) framework [13].

A. Simulation Setup

In the simulation setup, we consider $\mathcal{U} = [100, 500]$ number of IoT-user devices, and $\mathcal{P}$ number of service providers, where $\mathcal{F} = [10, 50]$ and $\mathcal{C} = [1, 5]$ in the 6G-enable fog federation model. According to 6G cellular characteristics, we take $\mathcal{O}_j^{CPU} = [1.5, 3]$ GHz, transmission power is [34,45] dBm, noise -174 dBm, and channel bandwidth is 30 MHz [28]. Further, we consider $T_{s,u}^{max} = [150, 250]$ time units, $\mathcal{X}_i^{in} = [60, 800]$ KB, required CPU cycle of each data [200,1400] cycles/bit and connection density 40/KM^2 in a 5×5 KM square area [29]. Besides that we consider T_j^{DL} and T_j^{UL} uniformly distributed such as [20,30] Mbps and [15,25] Mbps, respectively [23]. We set the maximum resource capacity of user devices, fog devices, and cloud servers as 512 MB, 4GB, and 64GB, respectively. Other simulation parameters are depicted in Table I.

B. Service Delay

This criterion addresses the issues related to the overall service delay of the proposed service deployment strategy on the federated fog model. The service delay in a network mostly depends on upstream data transmission, data processing, and downstream data transmission [30]. Further, from $T_j^{UL} = \widetilde{\mathcal{X}}_i^{in}/R_{u,j}^{trans}$ and $T_j^{DL} = \widetilde{\mathcal{X}}_i^{out} R_{j,u}^{trans}$, it can be easily observed that the transmission delay mostly depends on transmission rate. The transmission delay decreases while increasing the value of R^{trans}. Moreover, from $T_j^{MEC} = \widetilde{\mathcal{X}}_i^{in}\mathcal{Y}_i/\mathcal{O}_j^{CPU}$, we can analyze that processing delay of the computing devices inversely proportional to the computation frequency $T_j^{MEC} \propto 1/\mathcal{O}_j^{CPU}$, i.e., T_j^{MEC} increases with lower $\mathcal{O}_j^{CPU}$. For this experiment, we strictly consider $\mathcal{O}_f^{CPU} > \mathcal{O}_c^{CPU}$, where $\forall f \in \mathcal{F}$ and $\forall c \in \mathcal{C}$. Fig. 3 describes the comparison of overall service delay of the proposed fog federation model with the existing SF framework and SC framework. Fig. 3(a) and Fig. 3(b) reveals that the proposed fog federation model is stable and can handle more workload in 6G-enabled networks. However, as the number of IoT-users increases, $T_{u,j}^{service}$ also increases drastically. This also indicates the necessity to increase the number of fog devices near the IoT-users. Experimental observations reveal that the proposed fog federation model succeeded in reducing up to 15%-25% of service delay as compared with the existing SF framework and SC framework.

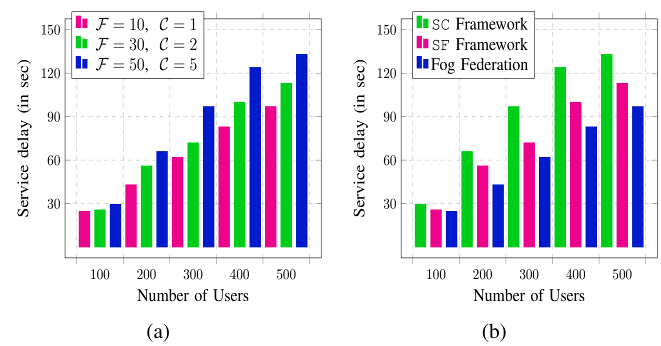
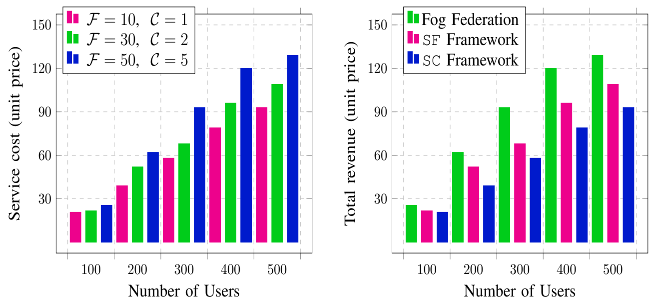


Fig. 3. Service delay. (a) Among the various computing devices. (b) Comparison with baseline frameworks.

C. Total Revenue

This metric represents the cost performance of the proposed service deployment strategy on the fog federation model in terms of user service cost $C_{s,u}^{MD}$, and providers service deployment cost $C_{s,j}^{MEC}$ including processing cost, maintaining cost, and storage cost. For this experiment, we define $\beta_{s,j}^{stor} = [0.3, 0.7]$, $\beta_{s,j}^{main} = [0.3, 0.7]$, $\beta_{s,j}^{proc} = [0.3, 0.7]$ and $\beta_{s,j}^{trans} = [0.3, 0.7]$. From **P1** and **P2**, it is worth noting that the service cost for an IoT-user mainly depends on the input data size $\mathcal{X}_i^{in}$ and the price per unit of resource usage $\alpha_{s,u}^{user}$. Similarly, on the provider side, resource cost mostly depends on requested data $\widetilde{\mathcal{X}}_i^{in}$ and price structure of the provider i.e., $\beta_{s,j}^{MEC}$. Besides that, these two objectives are directly connected with the computational frequency of the computing devices. With lower computation capacity $\mathcal{O}_f^{CPU}$ in fog devices, most of the IoT-users unable to obtain service directly from the fog device, hence, the IoT-users have to pay more amount of service cost. Simultaneously, the IoT-user satisfaction ratio also decreases. Fig. 4 presents the performance analysis of the proposed service deployment strategy in federated fog networks. From Fig. 4(a), it can be easily observed that the computation cost of the proposed strategy comparatively less with limited IoT-users, and increases with more number of IoT-users. Further, Fig. 4(a) shows that the overall service cost can be controlled by increasing the numbers of fog devices in the fog federated model. Finally, Fig. 4(b) illustrates the performance improvement of the proposed fog federation model up to 20%-25% as compared with the existing SF framework and SC framework.

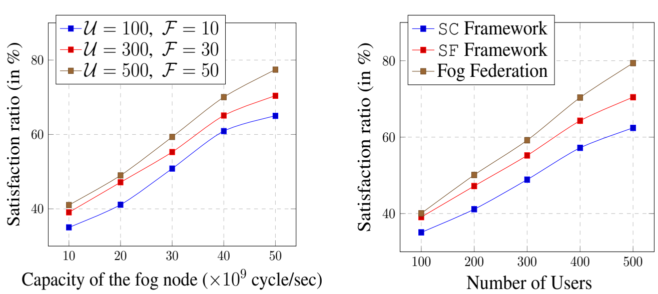


(a) (b)

Fig. 4. Total revenue. (a) Among the various computing devices. (b) Comparison with baseline frameworks.

D. User Satisfaction Ratio

The user satisfaction ratio $\mathcal{R}^{satis}_{s,u}$ represents the overall performance of the network in terms of user QoS parameter. Moreover, according to industrial standard, IoT-users satisfaction ratio should be lies in between $l_1 \leq \mathcal{R}^{satis}_{s,u} \leq l_2, \quad u \in U, s \in S$ [20]. Fig. 5 represents the performance of the proposed service deployment strategy concerning IoT-users $T^{max}_{s,u}$, service providers $\beta^{p\_max}_{s,J}$ and $\beta^{s\_max}_{s,J}$ constraints. From Fig. 5(a) and Fig. 5(b), it is obvious to say that the proposed fog federation model offers more satisfaction ratio compared to the SC framework and SF framework. The reason behind that the proposed model utilizes the advantages of resource controller over the fog federated network, in which the resource controller monitors the available resources $\mathcal{R}^{RM}$ and unsatisfied users $\mathcal{O}^{RM}$. In this way, resource provider maximizes their revenue $C^{MEC}_{s,j}$ and increase IoT-users satisfaction ratio $\mathcal{R}^{satis}_{s,u}$. Analysis shows that proposed fog federation model succeed to increase IoT-user satisfaction ratio up to 20%-30% as compared to the existing SF framework and SC framework.



(a) (b)

Fig. 5. Users satisfaction ratio: (a) On various computing devices, (b) Comparison with existing baseline frameworks.

The above experiment results show that the proposed service deployment strategy on the fog federation model indeed solves the difficulties and challenges of the existing standalone fog and cloud frameworks with 6G communication technology. It performs efficiently under the heavy load and reduces the user service cost while maximizing the revenue of the providers.

V. Conclusion

In this work, we have designed a service deployment strategy to facilitate massive IoT applications in the 6G-enabled fog federated network. The main objective of this model is to maximize the cost-benefit for both the IoT-user and the service provider while convening various constraints. For this purpose, a Stackelberg game approach is adopted to minimize the users' service costs using a *multi-user single-provider* strategy and maximize the revenue of the service providers using a *single-user multi-provider* strategy. Further, to control users' dynamic service demands and maximize users satisfaction ratio, a resource controlled mechanism is incorporated with the proposed fog federation model. This mechanism helps to redeploy the requested services on nearby fog devices. Extensive simulation results with 6G-enable parameters demonstrate that our proposed fog federation model reduces up to 15%-20% service delay and 20%-25% of service cost over the SF or SC frameworks. In the future, we will extend our proposed fog federation model by incorporating software-defined network to achieve maximum benefits of the 6G network.

References

[1] M. Alazab, S. Khan, S. S. R. Krishnan, Q. Pham, M. P. K. Reddy, and T. R. Gadekallu, "A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid," *IEEE Access*, vol. 8, pp. 85 454–85 463, 2020.

[2] A. K. Das, N. Kumar, M. Alazab *et al.*, "Designing Authenticated Key Management Scheme in 6G-enabled Network in a Box Deployed for Industrial Applications," *IEEE Transactions on Industrial Informatics*, 2020.

[3] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. Pham, S. K. Padannayil, and K. Simran, "A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4436–4456, 2020.

[4] V. G. Menon and J. Prathap, "Vehicular Fog Computing: Challenges Applications and Future Directions," *International Journal of Vehicular Telematics and Infotainment Systems (IJVTIS)*, vol. 1, no. 2, pp. 15–23, 2017.

[5] V. G. Menon, S. Jacob, S. Joseph, P. Sehdev, M. R. Khosravi, and F. Al-Turjman, "An IoT-Enabled Intelligent Automobile System for Smart Cities," *Internet of Things*, p. 100213, 2020.

[6] Y. Wang, X. Wang, and L. Wang, "Low-Complexity Stackelberg Game Approach for Energy-Efficient Resource— Allocation in Heterogeneous Networks," *IEEE Communications Letters*, vol. 18, no. 11, pp. 2011–2014, 2014.

[7] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog Computing for Sustainable Smart Cities: A Survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–43, 2017.

[8] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, "Energy-Aware Marine Predators Algorithm for Task Scheduling in IoT-based Fog Computing Applications," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[9] J. Li, A. Wu, S. Chu, T. Liu, and F. Shu, "Mobile Edge Computing for Task Offloading in Small-Cell Networks via Belief Propagation," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[10] R. Chaudhary, N. Kumar, and S. Zeadally, "Network Service Chaining in Fog and Cloud Computing for the 5G Environment: Data Management and Security Challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 114–122, 2017.
[11] M. Mukherjee, S. Kumar, M. Shojafar, Q. Zhang, and C. X. Mavromoustakis, "Joint Task Offloading and Resource Allocation for Delay-Sensitive Fog Networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.
[12] C. Yi, J. Cai, and Z. Su, "A Multi-User Mobile Computation Offloading and Transmission Scheduling Mechanism for Delay-Sensitive Applications," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 29–43, 2020.
[13] G. S. Aujla, M. Singh, N. Kumar, and A. Y. Zomaya, "Stackelberg Game for Energy-Aware Resource Allocation to Sustain Data Centers Using RES," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 1109–1123, 2019.
[14] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based Resource Allocation for Edge Computing: A Market Equilibrium Approach," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2018.
[15] F. Farivar, M. S. Haghighi, A. Jolfaei, and M. Alazab, "Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2716–2725, 2020.
[16] Y. Liu, R. Wang, and Z. Han, "Interference-Constrained Pricing for D2D Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 475–486, 2016.
[17] M. Adhikari, M. Mukherjee, and S. N. Srirama, "DPTO: A Deadline and Priority-Aware Task Offloading in Fog Computing Framework Leveraging Multilevel Feedback Queueing," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5773–5782, 2020.
[18] A. Irshad, M. Usman, S. A. Chaudhry, A. K. Bashir, A. Jolfaei, and G. Srivastava, "Fuzzy-in-the-Loop-Driven Low-Cost and Secure Biometric User Access to Server," *IEEE Transactions on Reliability*, 2020.
[19] C. Benzaid, K. Lounis, A. Al-Nemrat, N. Badache, and M. Alazab, "Fast authentication in wireless sensor networks," *Future Generation Computer Systems*, vol. 55, pp. 362–375, 2016.
[20] X. Cao, G. Tang, D. Guo, Y. Li, and W. Zhang, "Edge federation: Towards an integrated service provisioning model," *IEEE/ACM Transactions on Networking*, 2020.
[21] N. Kumar, N. Chilamkurti, and S. C. Misra, "Bayesian coalition game for the internet of things: an ambient intelligence-based evaluation," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 48–55, 2015.
[22] M. Liu and Y. Liu, "Price-Based Distributed Offloading for Mobile-Edge Computing With Computation Capacity Constraints," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 420–423, 2018.
[23] M. Mukherjee, V. Kumar, J. Lloret, and Q. Zhang, "Revenue Maximization in Delay-aware Computation Offloading among Service Providers with Fog Federation," *IEEE Communications Letters*, 2020.
[24] N. Kumar, J. J. P. C. Rodrigues, and N. Chilamkurti, "Bayesian Coalition Game as-a-Service for Content Distribution in Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 544–555, 2014.
[25] N. Kumar, S. Misra, J. J. P. C. Rodrigues, and M. S. Obaidat, "Coalition Games for Spatio-Temporal Big Data in Internet of Vehicles Environment: A Comparative Analysis," *IEEE Internet of Things Journal*, vol. 2, no. 4, pp. 310–320, 2015.
[26] R. Chaudhary, G. S. Aujla, N. Kumar, and J. J. P. C. Rodrigues, "Optimized Big Data Management across Multi-Cloud Data Centers: Software-Defined-Network-Based Analysis," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 118–126, 2018.
[27] A. Hazra, M. Adhikari, T. Amgoth, and S. Srirama, "Joint Computation Offloading and Scheduling Optimization of IoT Applications in Fog Networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.
[28] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency Federated Learning and Blockchain for Edge Association in Digital Twin empowered 6G Networks," *IEEE Transactions on Industrial Informatics*, 2020.
[29] W. Sun, R. Wang, Y. ZHANG *et al.*, "Reducing Offloading Latency for Digital Twin Edge Networks in 6G," *IEEE Transactions on Vehicular Technology*, 2020.
[30] M. Adhikari, S. N. Srirama, and T. Amgoth, "Application Offloading Strategy for Hierarchical Fog Environment Through Swarm Optimization," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4317–4328, 2020.

Abhishek Hazra is currently pursuing Ph.D. in IIT(ISM) Dhanbad, India. He completed his master's degree in Computer Science and Engineering from NIT Manipur, India in 2018 and Bachelor degree from NIT Agartala, India in 2014. He has authored and co-authored various national and international journal and conference articles. His research area of interest is in the field of Fog computing, 6G and Industrial Internet of Things.

Mainak Adhikari is currently working as a Post Doctorate Research Fellow at University of Tartu, Estonia. He has completed his Ph.D in Cloud Computing from IIT(ISM) Dhanbad, India in 2019. He has obtained his M.Tech. from Kalyani University in the year 2013. He earned his B.E. Degree from West Bengal University of Technology in the year of 2011. He is the Associate Editor of Cluster Computing and IEEE IoT Magazine.

Tarachand Amgoth received B.Tech in Computer Science and Engineering from JNTU, Hyderabad and M.Tech in Computer Science Engineering from NIT, Rourkela in 2002 and 2006 respectively and Ph.D. form IIT(ISM), Dhanbad in 2015. Presently, he is working as an Assistant professor in the Department of Computer Science and Engineering, IIT(ISM), Dhanbad. His current research interest includes Fog/Edge computing, and Internet of Things.

Satish Narayana Srirama is an Associate Professor at School of Computer and Information Sciences, University of Hyderabad, India. He was a Research Professor and the head of the Mobile & Cloud Lab at the Institute of Computer Science, University of Tartu, Estonia. His current research focuses on cloud computing, mobile cloud, IoT and Fog computing. He is an IEEE Senior Member and an Editor of Wiley Software: Practice and Experience journal.