Bandwidth Allocation for Multiple Federated Learning Services in Wireless Edge Networks

Jie Xu, *Senior Member, IEEE*, Heqiang Wang, and Lixing Chen

***Abstract*—This paper studies a federated learning (FL) system, where *multiple* FL services co-exist in a wireless network and share common wireless resources. It fills the void of wireless resource allocation for multiple simultaneous FL services in the existing literature. Our method designs a two-level resource allocation framework comprising *intra-service* resource allocation and *inter-service* resource allocation. The intra-service resource allocation problem aims to minimize the length of FL rounds by optimizing the bandwidth allocation among the clients of each FL service. Based on this, an inter-service resource allocation problem is further considered, which distributes bandwidth resources among multiple simultaneous FL services. We consider both cooperative and selfish providers of the FL services. For cooperative FL service providers, we design a distributed bandwidth allocation algorithm to optimize the overall performance of multiple FL services, meanwhile catering it to the fairness among FL services and the privacy of clients. For selfish FL service providers, a new auction scheme is designed with the FL service providers as the bidders and the network operator as the auctioneer. The designed auction scheme strikes a balance between the overall FL performance and fairness. Our simulation results show that the proposed algorithms outperform other benchmarks under various network conditions.**

***Index Terms*—Federated learning (FL), bandwidth allocation, edge computing.**

I. Introduction

TODAY'S mobile devices are generating an unprecedented amount of data every day. Leveraging the recent success of machine learning (ML) and artificial intelligence (AI), this rich data has the potential to power a wide range of new functionalities and services, such as learning the activities of smart phone users, predicting health events from wearable devices or adapting to pedestrian behavior in autonomous vehicles. With the help of multi-access edge computing (MEC) servers, ML models can be quickly trained/updated using this data to adapt to the changing environment without moving the data to the remote cloud data center, which is envisioned in intelligent next-generation communication systems [1], [2]. Furthermore, due to the growing storage and computational power of mobile devices as well as privacy concerns associated with uploading personal data, it is increasingly attractive to store and process data directly on mobile devices. Federate learning (FL) [3] is thus proposed as a new distributed ML framework, where mobile devices collaboratively train a shared ML model with the coordination of an edge server while keeping all the training data on device, thereby decoupling the ability to do ML from the need to upload/store the data to/in a public entity.

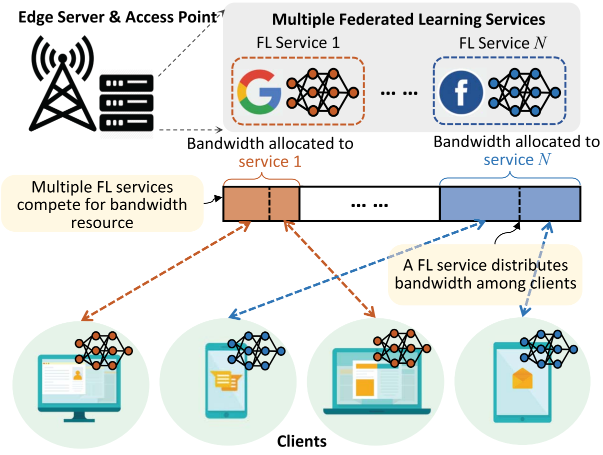
Edge Server & Access Point
Multiple Federated Learning Services
FL Service 1
FL Service N
Bandwidth allocated to service 1
Bandwidth allocated to service N
Multiple FL services compete for bandwidth resource
A FL service distributes bandwidth among clients
Clients

Fig. 1. System Overview.

A typical FL service involves a number of mobile devices (a.k.a., participating clients) and an edge server (a.k.a., a parameter server) to train a ML model, which lasts for a number of learning rounds. In each round, the clients download the current ML model from the server, improve it by learning from their local data, and then upload the individual model updates to the server; the server then aggregates the local updates to improve the shared model. For example, the seminal work [4] proposed the FedAvg algorithm in which the global model is obtained by averaging the parameters of local models. Although other FL algorithms differ in the specifics, the majority of them follow the same workflow. Because the clients work in the same wireless network to download and upload models, how to allocate the limited wireless bandwidth among the participating clients has a crucial impact on the resulting FL speed and efficiency.

Although existing works have made meaningful progress towards efficient resource allocation for wireless FL, they share the common limitation that only a *single* FL service was considered. As ML-powered applications grow and become more diverse, it is anticipated that the wireless network will host *multiple* co-existing FL services, the set of which may also dynamically change over time. See Figure 1 for an illustration of the multi-FL service scenario. The presence of

Manuscript received January 10, 2021; revised May 10, 2021 and July 30, 2021; accepted September 13, 2021. Date of publication September 24, 2021; date of current version April 11, 2022. This work was supported in part by NSF under Grant 2006630, Grant 2033681, Grant 2029858, and Grant 2044991. The associate editor coordinating the review of this article and approving it for publication was M. Sheng. *(Corresponding author: Jie Xu.)*

Jie Xu and Heqiang Wang are with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 22146 USA (e-mail: jiexu@miami.edu; hxw563@miami.edu).

Lixing Chen is with the Institute of Cyber Science and Technology, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lxchen@sjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3113346.

Digital Object Identifier 10.1109/TWC.2021.3113346

multiple FL services makes resource allocation for wireless FL much more challenging. **First**, the achievable FL performance depends on not only *intra-service* resource allocation among the participating clients within each FL service but also *inter-service* resource allocation among different FL services, and these two levels of allocation decisions are also strongly coupled. **Second**, the FL service providers (FLSP) may adopt different FL algorithms and choose different configurations (e.g., number of participating clients, number of epochs of local training, etc.), yet this information is not always available to the wireless network operator due to privacy concerns when making resource allocation decisions. **Third**, because FLSPs have their individual goals, they may have incentives to untruthfully reveal their valuation of the wireless bandwidth if by doing so they gain advantages in the inter-service bandwidth allocation. Without the correct information, there is no guarantee on the overall system performance. **Finally**, as in any multi-user system, resource allocation should strike a good balance between efficiency and fairness – every FLSP should obtain a reasonable share of the wireless resource to train their ML models.

In this paper, we make an initial effort to study wireless FL with multiple co-existing FL services, which share the same bandwidth to train their respective ML models. Our focus is on the efficient bandwidth allocation among different FL services as well as among the participating clients within each FL service, thereby understanding the interplay between these two levels of allocation decisions. Our main contributions are summarized as follows:

- We formalize a two-level bandwidth allocation problem for multiple FL services co-existing in the wireless network, which may start and complete at different time depending on their own demand and FL requirements. The model is general enough for any FL algorithm that involves downloading, local learning, uploading and global aggregation in each learning round, and hence has wide applicability in real-world systems. In addition, we explicitly take fairness into consideration when optimizing bandwidth allocation to ensure that no FL service is starved of bandwidth.
- We consider two use cases depending on the nature/goals of the FLSPs. In the first case, FLSPs are fully cooperative to maximize the overall system performance. For this, we design a distributed optimization algorithm based on dual decomposition to solve the two-level bandwidth allocation problem. The algorithm keeps all FL-related information at the individual FLSP side without sharing it with the network operator, thereby reducing the communication overhead and enhancing privacy protection.
- We further consider a second case where FLSPs are selfishly maximizing their own performance. To address the selfishness issue, we design a multi-bid auction mechanism based on [5] to elicit the FLSPs' truthful valuation of bandwidth according to their submitted bids, with the following new contributions. Firstly, we prove that the FL frequency function (defined later) is differentiable, increasing and concave, which is needed for applying the multi-bid auction framework to address our problem. Secondly, we introduce a novel fairness-adjusted *ex post* charge to make a tunable trade-off between efficiency and fairness. Thirdly, we design a uniform multi-bidding mechanism as a deployment example.

The rest of this paper is organized as follows. Section II discusses related works. Section III builds the system model. Section IV formulates the problem for the cooperative case and develops a distributed bandwidth allocation algorithm. Section V studies the selfish FLSPs case and develops a multi-auction mechanism. Section VI performs simulations. Section VII Concludes the paper.

II. Related Work

A lot of research has been devoted to tackling various challenges of FL, including but not limited to developing new optimization and model aggregation methods [6]–[8], handling non-i.i.d. and unbalanced datasets [9]–[11], dealing with the straggler problem [12], preserving model and data privacy [13], [14], and ensuring fairness [15], [16]. A comprehensive review of these challenges can be found in [17]–[19]. In particular, the communication aspect of FL has been recognized as a primary bottleneck due to the tension between uploading a large amount of model data for aggregation and the limited network resource to support this transmission, especially in a wireless environment [20]. In this regard, early research on communication-efficient FL largely focuses on reducing the amount of transmitted data while assuming that the underlying communication channel has been established, e.g., updating clients with significant training improvement [21], compressing the gradient vectors via quantization [22], or accelerating training using sparse or structured updates [23]. More recent research starts to address this problem from a more communication system perspective, e.g., using a hierarchical FL network architecture [24] that allows partial model aggregation, and leveraging the wireless transmission property to perform analog model aggregation over the air [25], [26].

As wireless networks are envisioned as a main deployment scenario of FL, wireless resource allocation for FL is another active research topic. Many existing works [27]–[29] study the trade-off between local model update and global model aggregation. Client selection is essential to enable FL at scale and address the straggler problem. Different types of joint bandwidth allocation and client scheduling policies [30]–[34] have been proposed to either minimize the training loss or the training time. In all these works, resource allocation is carried out among clients of a *single* FL service, while assuming that the FL service itself has already received dedicated resource. In stark contrast, our paper studies a network consisting of multiple co-existing FL services and performs resource allocation at both the FL service level and the client level. We notice that a related problem where multiple FL services are being trained at the same time is also considered in a recent work [35]. In that paper, different FL services run on the same set of clients and a joint computation and communication resource scheduling problem is studied. In our paper, different FL services have their separate client sets which may experience very different channel qualities. Moreover, while [35]

TABLE I
LIST OF NOTATIONS

Notation	Description	Notation	Description
$b_{n,k}$	Bandwidth of client k in service n	$w^{\text{LC}}_{n,k}$	Local computation workload of client k in service n
b_n	Bandwidth of service n	w^{GC}_n	Global model update workload of parameter server n
T	Length of a single period	$t^{\text{DT}}_{n,k}$	Download transmission latency
$\mathcal{N}_i$	Set of active FL services at period i	$t^{\text{LC}}_{n,k}$	Local computation latency
$\mathcal{K}_n$	Set of participation clients of FL service n	$t^{\text{UT}}_{n,k}$	Upload transmission latency
ϕ_k	Computing speed of client k	t^{GC}_n	Global computation latency
ϕ_n	Computing speed of parameter server n	λ	Lagrange multipier
g^{ul}_k	Uplink wireless channel gains	γ	Step size of DISBA
g^{dl}_k	Downlink wireless channel gains	ϵ	Convergence gap of DISBA
P_n	Transmission power of parameter server n	s_n	Set of bids submitted by service n
P_k	Transmission power of client k	b^m_n	Requested bandwidth of service n with bid m
r^{DT}_k	Download transmission rate of client k	p^m_n	Unit price of service n with bid m
r^{UT}_k	Upload transmission rate of client k	α	Fairness-adjusted parameter
s^{DT}_n	Size of download model in service n	c_n	Payment of service n
s^{UT}_n	Size of upload model in service n	M	Number of bids

assumed that all clients are obedient, we study the possible selfish nature of FLSPs and highlight the bandwidth allocation fairness.

Considering each FL service as a "user", our problem is a type of multi-user wireless network resource allocation problems. While many concepts and techniques adopted in this paper, including proportional fairness [36], dual decomposition [37] and multi-bid auction [5], have seen applications in other domains, applying them in multi-service FL requires special treatment as two levels of resource allocation are involved in our problem. Particularly, there is no closed-form expression of how the performance (i.e., learning speed) of a FL service depends on the resource allocation among its clients. Therefore, understanding the inter-dependency of intra- and inter-service bandwidth allocation is essential. Furthermore, we emphasize the resource fairness among different FL services by designing a new fairness-adjusted multi-bid auction mechanism in the selfish FLSP case, thereby achieving a tunable tradeoff between efficiency and fairness. We point out that there are some existing works [38]–[41] on designing incentive mechanisms for client participation of a single FL service. These works are very different from ours in terms of both the problem and the approach, and do not consider fairness when designing the mechanism.

III. SYSTEM MODEL

We consider a wireless network where ML models are trained using FL. The wireless network has a total bandwidth B, and the network operator (NO) has to allocate this bandwidth among concurrent FL services when needed to enable their individual training. Because new FL services may start and old FL services may finish over time, bandwidth allocation has to be periodically performed to adapt to the current active FL services. Therefore, we divide time into periods and let the length of a period be T. At the beginning of each period i, a set $\mathcal{N}_i$ of FL services are active and require wireless bandwidth to carry out their training. These services are either newly initiated services in period i or continuing services from the previous period. A FL service finishes and hence exits the wireless network when a certain termination criteria is satisfied (e.g., the training loss is below a threshold, the testing accuracy is above a threshold, or other convergence criterion), which usually varies across FL services and are pre-specified by the corresponding FLSP. Therefore, a FL service may span multiple periods. The wall clock time (i.e. the number of periods) that a FL service takes to finish depends on the difficulty and other inherent characteristics of the service itself as well as how much wireless resource is allocated to this service in each period for which it stays and how this bandwidth is further allocated among its participating clients. In what follows, we first formulate the client-level (i.e., intra-service) bandwidth allocation problem and then describe the service-level (i.e., inter-service) bandwidth allocation problem. The notations that we will encounter are summarized in Table I.

A. Intra-Service Bandwidth Allocation

To understand how bandwidth allocation affects the FL speed, let us consider a single representative FL service n in one period (period index i is dropped for conciseness). Suppose that this service is allocated with a bandwidth b_n in this period, which is further allocated among its participating clients, the set of which is denoted by $\mathcal{K}_n$. For each client $k \in \mathcal{K}_n$, let ϕ_k be its computing speed, and g^{ul}_k and g^{dl}_k be the uplink and downlink wireless channel gains to the parameter server of service n, respectively, which are assumed to be invariant within a period. We consider a synchronized FL model for each FL service, where a number of FL rounds take place in a period. Nonetheless, different FL services do not have to be synchronized – they learn at their own pace. See Figure 2 for an illustration.

A FL round consists of four stages: download transmission, local computation, upload transmission and global computation:

- *Download Transmission (DT).* Each FL round starts with a DT stage in which each client k downloads the current global model from its parameter server residing on the base station. Suppose client k is allocated with

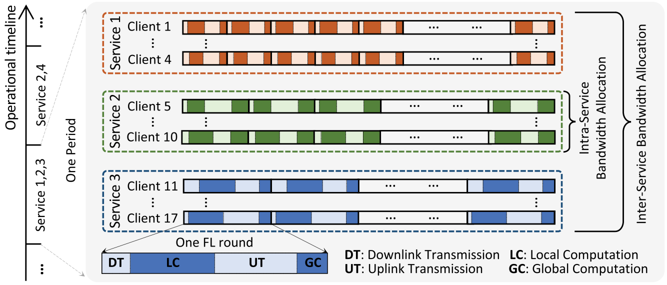

Fig. 2. Bandwidth Allocation among Multiple FL Services.

bandwidth $b_{n,k}$. Then the DT rate is $b_{n,k}\log_2(1+P_n g_k^{\mathrm{dl}}/N_0)$ following Shannon's equation, where P_n is the transmission power of parameter server n and N_0 is the noise power. For notational convenience, we denote $\log_2(1+P_n g_k^{\mathrm{dl}}/N_0) \triangleq r_k^{\mathrm{DT}}$ as the DT base rate of client k. Let s_n^{DT} be the download data size (e.g., the size of the global model), then the DT latency is $t_{n,k}^{\mathrm{DT}} = s_n^{\mathrm{DT}}/(b_{n,k} r_k^{\mathrm{DT}})$.

- *Local Computation (LC)*. With the current global model, each client k then updates its local model using its local dataset. Depending on the ML model complexity, the local dataset size and the number of epoches in local training, the per-round local computation workload is denoted by $w_{n,k}^{\mathrm{LC}}$. Therefore, the LC latency of client k is $t_{n,k}^{\mathrm{LC}} = w_{n,k}^{\mathrm{LC}}/\phi_k$.
- *Upload Transmission (UT)*. Once local update is finished, client k transmits the result to the parameter server n. Given the bandwidth $b_{n,k}$, its UT rate is $b_{n,k}\log_2(1+P_k g_k^{ul}/N_0)$, where P_k is the transmission power of client k and N_0 is the noise power. Again, for notational convenience, we denote $\log_2(1+P_k g_k^{ul}/N_0) \triangleq r_k^{\mathrm{UT}}$ as the UT base rate of client k. Let s_n^{UT} be the data size that has to be transmitted to the parameter server, then the UT latency of client k is $t_{n,k}^{\mathrm{UT}} = s_n^{\mathrm{UT}}/(b_{n,k} r_k^{\mathrm{UT}})$.
- *Global Computation (GC)*. Finally, once the local updates of all clients are received by parameter server n, the global model is updated. Let w_n^{GC} be the global model update workload and ϕ_n be the computing speed of parameter server n, then the GC latency is $t_n^{\mathrm{GC}} = w_n^{\mathrm{GC}}/\phi_n$.

Remark 1: Treating the bandwidth allocation as a fractional solution is mostly for mathematical convenience and also has been widely adopted in the existing works. Deriving an integer-valued solution for realistic wireless systems such as OFDMA can be done via rounding.

Remark 2: Uplink and downlink transmissions of a user do not occur at the same time, and hence the whole allocated bandwidth $b_{n,k}$ can be used for uplink or downlink when needed. For this reason, we do not use different notations for uplink and downlink.

Remark 3: Our framework is applicable to a vast set of FL algorithms (e.g., FedAvg, FedSGD) that can be chosen for service n. For instance, the downloaded/uploaded data may be the model itself, the compressed version of the model, or the model gradient information. For the purpose of bandwidth allocation, it is sufficient to describe the FL service as a tuple $\langle s_n^{\mathrm{DT}}, \{w_{n,k}^{\mathrm{LC}}\}_{k\in\mathcal{K}_n}, s_n^{\mathrm{UT}}, w_n^{\mathrm{GC}}\rangle$. Also note that although bandwidth allocation has no effect on the LC latency and the GC latency, the LC latency and the GC latency will affect the outcome of the bandwidth allocation.

In synchronized FL, the parameter server updates the global model until it has received the local updates from all participating clients. Hence, the length of a FL round of service n is determined by the total latency of the *slowest* client, i.e. $t_n = \max_{k\in\mathcal{K}_n}(t_{n,k}^{\mathrm{DT}} + t_{n,k}^{\mathrm{LC}} + t_{n,k}^{\mathrm{UT}} + t_n^{\mathrm{GC}})$. To minimize the FL round length t_n of service n so that more FL rounds can be executed in a period, one has to optimally allocate bandwidth b_n among the clients of service n. Given b_n, the *intra-service bandwidth allocation* problem can be formulated as

$$\min_{b_{n,1},\ldots,b_{n,K}} \quad t_n(\{b_{n,k}\}_{k\in\mathcal{K}_n}) \quad \text{subject to} \sum_{k\in\mathcal{K}_n} b_{n,k} = b_n \quad (1)$$

Let $t_n^*(b_n)$ denote the optimal solution to Eqn. (1). Then the optimal FL frequency of service n is $f_n^*(b_n) = 1/t_n^*(b_n)$, which is used to represent the FL speed of service n. Note that this means $T \cdot f_n^*(b_n)$ FL rounds can be performed in one period.

B. Inter-Service Bandwidth Allocation

In a period, multiple active FL services may be active and require wireless bandwidth to carry out learning. Since they share a total bandwidth B, how this bandwidth is allocated among different services will determine their achievable learning frequencies $f_n^*(b_n)$, thus the convergence speed in terms of the wall clock time. In this paper, we consider two scenarios depending on the goals of the FLSPs and how inter-service bandwidth allocation is implemented. In the first scenario, all FLSPs are *cooperative*, and their goal is to maximize the FL performance of the overall system. Therefore, it is equivalent to the NO solving a system-wide optimization problem. In the second scenario, FLSPs are *selfish* who only care about their own FL performance. As these FLSPs are competing for the limited bandwidth resource, addressing their incentive issues is crucial. In this paper, we design a fairness-adjusted multi-bid auction mechanism for this case. In the following two sections, we discuss these two scenarios separately.

IV. Cooperative Service Providers

In the cooperative FLSPs scenario, the NO directly decides the bandwidth allocation to maximize the overall system performance. As in any multi-user network, bandwidth allocation for multi-service FL has to address both efficiency and fairness – every active FL service should get a reasonable share of the bandwidth. Thus, we adopt the notion of *proportional fairness* [36], a metric widely used in multi-user resource allocation, and aim to solve the following optimization problem:

$$\max_{b_1,\ldots,b_N} \quad \sum_{n=1}^{N} \log(1 + f_n^*(b_n))$$

$$\text{subject to} \sum_{n=1}^{N} b_n = B \text{ and } f_n^*(b_n) \text{ solves (1)}, \quad \forall n \quad (2)$$

where we drop the period index i and let N be the number of active FL services in the period for conciseness. The objective function adds a "1" inside the logarithmic to ensure that the function value is always non-negative. This change has very little impact on the final allocation since the frequency is often much larger than 1 in a period. Note also that the above inter-service bandwidth allocation problem Eqn. (2) implicitly incorporates the intra-service problem as $f_n^*(b_n)$ is the solution to Eqn. (1).

A. Optimal Solution to the Intra-Service Problem

We first investigate the optimal solution to the intra-service bandwidth allocation problem and see how it can be used to solve the inter-service problem. According to our system model and Eqn. (1), the intra-service bandwidth allocation is equivalent to

$$\min_{b_{n,1},\ldots,b_{n,K},t_n} \quad t_n \tag{3}$$

$$\text{subject to } t_{n,k}^{\mathrm{C}} + \alpha_{n,k}/b_{n,k} \le t_n \tag{4}$$

$$\sum_k b_{n,k} = b_n \tag{5}$$

where we let $t_{n,k}^{\mathrm{C}} \triangleq t_{n,k}^{\mathrm{LC}} + t_n^{\mathrm{GC}}$ and $\alpha_{n,k} \triangleq s_n^{\mathrm{DT}}/r_k^{\mathrm{DT}} + s_n^{\mathrm{UT}}/r_k^{\mathrm{UT}}$ for notational convenience. Clearly, the optimal solution t^* must satisfy $t_{n,k}^{\mathrm{C}} + \alpha_{n,k}/b_{n,k} = t_n^*, \forall k$. Therefore, the optimal t_n^* solves the following equality,

$$\sum_k \frac{\alpha_{n,k}}{t_n^* - t_{n,k}^{\mathrm{C}}} = b_n \tag{6}$$

Although we do not have a closed-form solution of $t_n^*(b_n)$, a bi-section algorithm can be constructed to easily solve the above problem to obtain the optimal $t_n^*(b_n)$ and consequently the optimal frequency $f_n^*(b_n) = 1/t_n^*(b_n)$ as a function of b_n. Furthermore, the property of $f_n^*(b_n)$ can be characterized in the following lemma.

Lemma 1: $f_n^*(b_n)$ is a differentiable, increasing and concave function for $b_n > 0$.

Proof: Let us consider the inverse function $b_n(f_n)$ defined by Eqn. (6). It is easy to see that for $f_n \in [0, 1/\max_k t_{n,k}^{\mathrm{C}})$, $b_n(F_n)$ is a monotonically increasing function in f_n with $b_n(0) = 0$ and $b_n(f_n) \to \infty$ as $f_n \to 1/\max_k t_{n,k}^{cp}$. Therefore, for $b_n \ge 0$, $f_n(b_n)$ is also monotonically increasing. The first-order derivative of $b_n(f_n)$ is, $\forall f_n \in [0, 1/\max_k t_{n,k}^{\mathrm{C}})$,

$$b_n' = \frac{db_n}{df_n} = \frac{db_n}{dt_n}\frac{dt_n}{df_n} = \sum_k \frac{\alpha_{n,k}}{(1 - t_{n,k}^{\mathrm{C}} f_n)^2} > 0 \tag{7}$$

Therefore, $f_n(b_n)$ is differentiable for $b_n \ge 0$ and

$$f_n' = \frac{df_n}{db_n} = \left(\sum_k \frac{\alpha_{n,k}}{(1 - t_{n,k}^{\mathrm{C}} f_n)^2}\right)^{-1} > 0, \quad \forall b_n \ge 0 \tag{8}$$

The second-order derivative f_n'' can also be computed as follows: $\forall B \ge 0$,

$$f_n'' = -\left(\sum_k \frac{\alpha_{n,k}}{(1 - t_{n,k}^{C} f_n)^2}\right)^{-2} \left(\sum_k \frac{\alpha_{n,k} t_{n,k}^{\mathrm{C}}}{(1 - t_{n,k}^{\mathrm{C}} f_n)^3}\right) < 0 \tag{9}$$

This proves that $f_n(b_n)$ is a concave function for $b_n \ge 0$. □

With Lemma 1, it is straightforward to see that the inter-service bandwidth allocation problem (2) is a convex optimization problem.

Proposition 1: The inter-service bandwidth allocation problem (2) *is an equality-constrained convex optimization problem.*

Proof: Because f^* is concave, $\log$ is concave and increasing, the composition $\log(1 + f^*)$ is also a concave function. Then it is straightforward to see that the problem is a concave maximization problem with an equality constraint. □

B. Distributed Algorithm for Inter-Service Bandwidth Allocation

We now proceed with solving the inter-service bandwidth allocation problem. While various centralized convex optimization algorithms, such as the Newton's method, can efficiently solve the inter-service problem Eqn. (2), we prefer a distributed algorithm where individual FLSPs do not share their FL algorithm details and client-level information with each other or the NO. This way reduces the communication overhead and preserves privacy of the client devices of individual FLSPs. Our algorithm is developed based on dual decomposition [37] as follows.

We first relax the total bandwidth constraint $\sum_n b_n = B$ to be $\sum_n b_n \le B$, and then form the Lagrangian by relaxing the coupling constraint:

$$\begin{aligned} L(b_1, \ldots, b_N, \lambda) &= \sum_n \log(1 + f_n^*(b_n)) - \lambda\left(\sum_n b_n - B\right) \\ &= \sum_n L_n(b_n, \lambda) + \lambda B \end{aligned} \tag{10}$$

where λ is the Lagrange multipier associated with the total bandwidth constraint, and $L_n(b_n, \lambda) = \log(1 + f_n^*(b_n)) - \lambda b_n$ is the Lagrangian to be maximized by FLSP n. Such dual decomposition results in each FLSP n solving, for a given λ, the following problem

$$\begin{aligned} b_n^*(\lambda) &= \arg\max_{b_n \ge 0} L_n(b_n, \lambda) \\ &= \arg\max_{b_n \ge 0} (\log(1 + f_n^*(b_n)) - \lambda b_n) \end{aligned} \tag{11}$$

where the solution is unique due to the strict concavity of f_n^* according to Lemma 1. Specifically, to solve this maximization problem, we only need to solve its first-order condition,

$$f_n^{*\prime}(b_n)/(1 + f_n^*(b_n)) = \lambda \tag{12}$$

which can be converted to solve f^* using

$$(1 + f_n^*) \sum_{k \in \mathcal{K}_n} \frac{\alpha_{n,k}}{(1 - t_{n,k}^{\mathrm{C}} f_n^*)^2} = \lambda^{-1} \tag{13}$$

Clearly, the left-hand side is an increasing function of f_n^* for $f_n^* \in [0, 1/\max_k t_{n,k}^{\mathrm{C}})$ and thus, a simple bi-section algorithm can be devised to solve Eqn. (13) to obtain $f_n^*(\lambda)$. Then plugging $f_n^*(\lambda)$ (hence $t_n^*(\lambda)$) into Eqn. (6) yields the optimal $b_n^*(\lambda)$.

Let $g_n(\lambda) = \max_{b_n \geq 0} L_n(b_n, \lambda) = L_n(b_n^*(\lambda), \lambda)$ be the local dual function for FLSP n. Then the master dual problem is

$$\min_{\lambda} g(\lambda) = \sum_n g_n(\lambda) + \lambda B \quad \text{subject to } \lambda \geq 0 \tag{14}$$

Since $b_n^*(\lambda)$ is unique, it follows that the dual function $g_n(\lambda)$ is differentiable and the following gradient method can be used to iteratively update λ:

$$\lambda(j+1) = \left[\lambda(j) - \gamma\left(B - \sum_n b_n^*(\lambda(j))\right)\right]^+ \tag{15}$$

where j is the iteration index, $\gamma > 0$ is a sufficiently small positive step-size, and $[\cdot]^+$ denotes the projection onto the non-negative orthant. The dual variable $\lambda(j)$ will converge to the dual optimum λ^* as $j \rightarrow \infty$. Since the duality gap for the inter-service problem is zero and the solution to Eqn. (11) is unique, $b_n^*(\lambda(j))$ will also converge to the primal optimal variable b_n^*.

Algorithm 1 summarizes the proposed DISBA algorithm. The computation complexity in each iteration is as follows: each FLSP (in parallel to other FLSPs) computes $b_n^*(\lambda(j))$ by solving Equation (11). As we mentioned, this can be done by using a bi-section algorithm, which requires $O(\log(\max_k 1/(t_{n,k}^C)))$ iterations. The NO uses Eqn. (15) to update the dual variable, whose complexity is $O(1)$. Our algorithm uses a gradient method with a constant step size to update the dual variable, which is proven to converge to the optimal value [42].

Algorithm 1 Distributed Inter-Service Bandwidth Allocation (DISBA)

```
Input to NO: total bandwidth $B$, step size $\gamma$, convergence gap $\epsilon$
Input to FLSP $n$: FL service $n$ parameters $\langle s_n^{\text{DT}}, \{w_{n,k}^{\text{LC}}\}_{k \in \mathcal{K}_n}, s_n^{\text{UT}}, w_n^{\text{GC}} \rangle$, channel gains and computing speed of its clients $\mathcal{K}_n$.
Initialization: set $j = 0$ and $\lambda(0)$ equal to some non-negative value
while $\lambda(j) - \lambda(j-1) > \epsilon$ do
    NO sends $\lambda(j)$ to all FLSPs
    Each FLSP $n$ obtains $b_n^*(\lambda(j))$ by solving Eqn. (11) using bi-section
    Each FLSP $n$ sends $b_n^*(\lambda(j))$ to NO
    NO updates $\lambda(j+1)$ according to Eqn. (15)
    $j \leftarrow j+1$
end while
```

V. Selfish Service Providers

In the previous section, DISBA works by letting each FLSP compute the allocated bandwidth $b_n^*(\lambda(j))$ given $\lambda(j)$. This, however, creates an opportunity for a selfish FLSP to mis-report its computation result that favors itself but reduces the system performance as a whole. In fact, even if the inter-service bandwidth allocation problem (2) is solved in a centralized way, similar selfish behavior may still undermine the efficient system operation as a selfish FLSP may mis-report its FL service and client parameters (e.g., FL workload, client computing power and channel gains etc.), which will alter the frequency function f_n^* used at the NO side. With a wrong frequency function f_n^*, the NO will not be able to determine the *true* optimal bandwidth allocation.

In this section, we address the selfishness issue in inter-service bandwidth allocation by designing a multi-bid auction mechanism. This auction mechanism will ensure that the FLSPs are using their true FL frequency functions f_n^* when making bandwidth bids.

A. Multi-Bid Auction

First, we describe the general rules of the multi-bid auction mechanism.

1) Bidding: At the beginning of each bandwidth allocation period, each FLSP n submits a set of M bids $s_n = \{s_n^1, \ldots, s_n^M\}$. For each $m \in \{1, \ldots, M\}$, $s_n^m = (b_n^m, p_n^m)$ is a two-dimensional bid, where b_n^m is the requested bandwidth and p_n^m is the unit price that FLSP n is willing to pay to get the requested bandwidth b_n^m. Without loss of generality, we assume that bids are sorted according to the price such that $p_n^1 \leq p_n^2 \leq \ldots \leq p_n^M$. Let $S \in \mathbb{R}^+ \times \mathbb{R}^+$ denote the set of multi-bids that a FLSP can submit.

2) Bandwidth Allocation and Charges: Once the NO collects all multi-bids from all FLSPs, denoted by $s = \{s_n\}_{n \in \mathcal{N}}$, it computes and implements the inter-service bandwidth allocation $(b_1, \ldots, b_N)$. Each FLSP n then further allocates b_n to its clients to perform FL. At the end of the period, the NO determines the charges $(c_1, \ldots, c_N)$ for all FLSPs depending on the allocated bandwidth and the realized FL performance.

Now, a couple of issues remain to be addressed. First, how to compute the bandwidth allocation and determine the charges given the FLSP-submitted multi-bids? Second, do the FLSPs have incentives to truthfully report their valuations of the bandwidth? These are the questions to be addressed in the next subsections.

B. Market Clearing Prices With Full Information

We first consider a simpler case where the FLSPs *truthfully* report the *complete* FL frequency function $f_n^*(b), \forall n$ to the NO. This analysis will provide us with insights on how to design bandwidth allocation and charging rules in the more difficult multi-bid auction case.

Recall that $f_n^*(b)$ is the optimal FL frequency of service n if it has bandwidth b. Taking into account the price paid to obtain this bandwidth, the (net) utility of FLSP n is

$$u_n(b;p) = f_n^*(b) - p \cdot b \tag{16}$$

Now, if the bandwidth were sold at the unit price p, then FLSP n would buy $b_n(p) = \arg\max_b u_n(b;p)$ bandwidth in order to maximize its utility. We call $b_n(p)$ the **bandwidth demand function** (BDF), and it is easy to show that $b_n(p) = (f_n^{*\prime})^{-1}(p)$ by checking the first-order condition of Eqn. (16). On the other hand, if FLSP n requires a bandwidth b, then the FLSP would pay a unit price no more than $p_n(b) = f_n^{*\prime}(b)$. We call $p_n(b)$ the **marginal valuation function** (MVF).

1) Market Clearing Price: With the complete information of $f_n^*(b)$ and hence BDF $b_n(p)$ for all FLSPs, the NO can compute the **market clearing price** (MCP) ρ so that $\sum_{n=1}^N b_n(\rho) = B$. One can prove that the MCP is unique and optimal in the sense that it maximizes the total (equivalently, average) FL frequency.

Proposition 2: The market clearing price ρ is unique and maximizes the total FL frequency $\sum_{n=1}^N f_n^*(b_n)$.

Proof: According to Lemma 1, $f_n^{*\prime}(b)$ is an increasing function. Therefore, the BDF, which is the inverse function of $f_n^{*\prime}(b)$ is also increasing. As a result, there exists a unique solution to the increasing function $\sum_{n=1}^N b_n(p_n) = B$. To show that $\bar{p}$ maximizes $\sum_{n=1}^N f_n^*(b_n)$, consider the following maximization problem

$$\max_{b_1,\ldots,b_N} \sum_{n=1}^N f_n^*(b_n) \quad \text{subject to} \sum_{n=1}^N b_n = B \tag{17}$$

This is clearly a convex optimization problem. Consider its Karush-Kuhn-Tucker conditions. In particular, the stationarity condition is

$$\nabla \sum_{n=1}^N f_n^*(b_n) + \lambda \nabla (\sum_{n=1}^N b_n - B) = 0 \tag{18}$$

where λ is the Lagrangian multiplier associated with the constraint. The solution requires

$$f_n^{*\prime}(b_n) = \lambda, \quad \forall n \tag{19}$$

Together with the feasibility constraint, this is equivalent to imposing a homogeneous market clearing price. □

Because $b_n(p)$ is a monotonically decreasing function in p, a bi-section algorithm can be easily designed to find the unique market clearing price so that $\sum_{n=1}^N b_n(\rho) = B$.

2) Fairness-Adjusted Costs: One major issue with the above pricing scheme is that it ignores fairness among the FLSPs: although it maximizes efficiency in terms of the average FL frequency according to Proposition 2, it is possible that the average FL frequency is maximized at an operating point where a few FLSPs are allocated with most of the bandwidth while some FLSPs obtain very little. In this paper, we design and incorporate a fairness-adjusted charging scheme. The payment of FLSP n now consists of two parts as follows:

- The first part of the payment depends on the amount of bandwidth b_n allocated to the FLSP n, and the unit price p set by the NO. Specifically, this payment is $p \cdot b_n$.
- The second part of the payment depends on the realized FL frequency f_n of FLSP n. Specifically, FLSP n will be charged a **fairness-adjusted cost** of $\alpha \cdot (f_n - \log(1+f_n))$ at the end of the period once f_n has been realized, where $\alpha \in [0,1]$ is a tunable parameter.

With these payments, FLSP n's utility becomes

$$\begin{aligned} u_n(b;p) &= f_n^*(b) - p \cdot b - \alpha \cdot (f_n^*(b) - \log(1 + f_n^*(b))) \\ &= g_n(b) - p \cdot b \end{aligned} \tag{20}$$

where $g_n(b) \triangleq (1-\alpha) f_n^*(b) + \alpha \log(1 + f_n^*(b))$. Comparing this new utility function Eqn. (20) with Eqn. (16), we make the following remarks. First, the fairness-adjusted cost essentially replaces $f_n^*(b)$ with $g_n(b)$. The decision problem remains largely the same except that now we have a different benefit function. Second, in the new utility function Eqn. (16), given any allocated bandwidth b, it is still in the FLSP's interest to perform the optimal client-level bandwidth allocation to maximize $f_n(b)$. This is because $g_n(b)$ is an increasing function in $f_n(b)$ for $\alpha \in [0,1]$. Therefore, we can directly write $g_n(b)$ as a function of the optimal FL frequency $f_n^*(b)$. Third, to charge the fairness-adjusted cost, the NO does not need to know the exact function $f_n^*(b)$. Rather, it only has to know the realized FL frequency f_n at the end of the current period. This is key to achieving fairness in multi-bid auction where FLSPs do not report the complete FL frequency function $f_n^*(b)$.

We call $d_n(p) = (g_n')^{-1}(p)$ the modified bandwidth demand function (mBDF). Likewise, we call $q_n(b) = g_n'(b)$ the modified marginal valuation function (mMVF). The NO can similarly compute the modified market clearing price (mMCP) ζ so that $\sum_{n=1}^N d_n(\zeta) = B$. Using a similar argument that proves Proposition 2, one can prove Proposition 3 as follows.

Proposition 3: The mMCP ζ is unique and the resulting bandwidth allocation $(b_1, \ldots, b_N)$ maximizes $\sum_{n=1}^N [(1-\alpha) f_n^*(b_n) + \alpha \log(1 + f_n^*(b))]$.

Proof: Because $f_n^*(b)$ is a concave increasing function, $\log(1 + f_n^*(b))$ is also concave and increasing. This further shows that $g_n(b)$ is concave and increasing. The rest follows similar arguments in the proof of Theorem 2. □

The parameter α makes a tradeoff between efficiency and fairness. On the one hand, setting $\alpha = 0$ reduces the problem to the total FL frequency maximization problem. On the other hand, setting $\alpha = 1$ achieves proportional fairness among the FLSPs.

C. Bandwidth Allocation and Charging Rules

Now, we are ready to describe the bandwidth allocation and charging rules in fairness-adjusted multi-bid auction. In this subsection, each FLSP n submits only a multi-bid $s_n = (s_n^1, \ldots, s_n^M)$ instead of the complete FL frequency function $f_n^*(b)$. However, we will assume that the FLSPs are *truthfully* submitting their bids, which will be proven indeed true in the next subsection. Specifically, we say that a bid $s_n^m = (b_n^m, p_n^m)$ is truthful if the bandwidth demand b_n^m and the price p_n^m that FLSP n is willing to pay satisfy the mBDF because it reveals FLSP n's true valuation of bandwidth after taking into consideration the fairness-adjusted costs. A multi-bid is truthful if all bids are truthful.

Definition 1 (Truthful Multi-Bid): A multi-bid $s_n = (s_n^1, \ldots, s_n^M)$ is truthful if $\forall m$, $s_n^m = (b_n^m, p_n^m)$ is such that $p_n^m = g_n'(b_n^m)$.

The NO does not know the BDF (and hence the mBDF) of each FLSP n because it does not have access to the FL frequency function f_n^*. Nonetheless, suppose FLSP n submitted a truthful multi-bid s_n, then the NO can compute a **pseudo-mBDF** using these bids to have some idea of the actual mBDF. Specifically, given the submitted multi-bid s_n, a left-continuous step function can be used to describe the

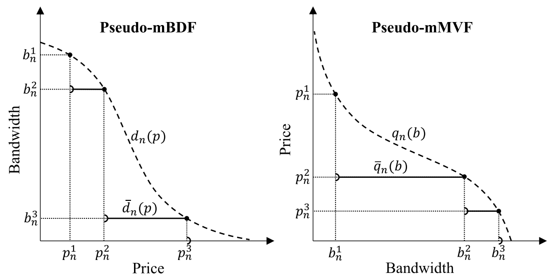

Fig. 3. Pseudo-mBDF and Pseudo-mMVF.

pseudo-mBDF as follows,

$$\bar{d}_n(p) = \begin{cases} 0, & \text{if } p_n^M < p \\ \max\limits_{1\le m\le M}\{b_n^m : p_n^m \ge p\}, & \text{otherwise} \end{cases} \tag{21}$$

Essentially, the pseudo-mBDF uses b_n^m to approximate the bandwidth demand for prices in the range $(p_n^{m-1}, p_n^m]$. Similarly, the NO can also construct a **pseudo-mMVF** (pseudo-MVF), an approximation of FLSP n's actual mMVF using the submitted multi-bid, as follows,

$$\bar{q}_n(b) = \begin{cases} 0, & \text{if } b_n^1 < b \\ \max\limits_{1\le m\le M}\{p_n^m : b_n^m \ge b\}, & \text{otherwise} \end{cases} \tag{22}$$

In other words, the pseudo-mMVF uses p_n^m to approximate the marginal value for bandwidth allocation in the range $[b_n^m, b_n^{m+1})$. We illustrate pseudo-mBDF and pseudo-mMVF in Figure 3.

The **aggregated pseudo-mBDF** is the sum of pseudo-mBDFs of all FLSPs:

$$\bar{d}(p) = \sum_{n=1}^{N} \bar{d}_n(p) \tag{23}$$

The **pseudo-mMCP** $\bar{\zeta}$ is the largest possible price so that the aggregated pseudo-mBDF exceeds the total available bandwidth, i.e.,

$$\bar{\zeta} = \sup\{p : \bar{d}(p) > B\} \tag{24}$$

This implies that reducing the mMCP by just a little bit will result in the supply (i.e., the total available bandwidth B) being no greater than the demand. Because every individual pseudo-mBDF function is a step function with K steps, the aggregated pseudo-mBDF is also a step function with at most NK steps. Therefore, the complexity of computing $\bar{\zeta}$ is at most $O(NK)$.

Next, we describe our bandwidth allocation and charging rules. For notational convenience, let $y(x^+) = \lim_{z\to x, z>x} y(x)$ when this limit exists for a function $y : \mathbb{R} \to \mathbb{R}$ and all $x \in \mathbb{R}$.

1) Bandwidth Allocation: With the pseudo-mMCP $\bar{\zeta}$, our bandwidth allocation rule is as follows: if FLSP n submits the multi-bid s_n (and thereby declares the associated functions $\bar{d}_n$ and $\bar{q}_n$), then it receives bandwidth $b_n(s_n, s_{-n})$, with

$$b_n(s_n, s_{-n}) = \bar{d}_n(\bar{\zeta}^+) + \frac{\bar{d}_n(\bar{\zeta}) - \bar{d}_n(\bar{\zeta}^+)}{\bar{d}(\bar{\zeta}) - \bar{d}(\bar{\zeta}^+)}\left(B - \bar{d}(\bar{\zeta}^+)\right) \tag{25}$$

In other words: (1) Each FLSP n receives an amount of bandwidth it asks for at the lowest price $\bar{\zeta}^+$ for which supply exceeds the pseudo-bandwidth demand. (2) If all bandwidth is not allocated yet, the surplus $B - \bar{d}(\bar{\zeta}^+)$ is shared among FLSPs. This sharing is done proportionally to $\bar{d}_n(\bar{\zeta}) - \bar{d}_n(\bar{\zeta}^+)$ as we notice that $\bar{d}(\bar{\zeta}) - \bar{d}(\bar{\zeta}^+) = \sum_{n=1}^{N}(\bar{d}_n(\bar{\zeta}) - \bar{d}_n(\bar{\zeta}^+))$, and ensures that all bandwidth is allocated.

2) Charging: Given the submitted multi-bids s, each FLSP n is charged $c_n(s)$ as follows,

$$\begin{aligned} &c_n(s_n, s_{-n}) \\ &\quad = \sum_{j\ne n} \int_{b_j(s)}^{b_j(s_{-n})} \bar{q}_j(b)db + \alpha \cdot (f_n^*(b_n) - \log(1 + f_n^*(b_n))) \end{aligned} \tag{26}$$

The first term on the right-hand side is based on the *exclusion-compensation* principle in second-price auction mechanisms [43]: FLSP n pays so as to cover the "social opportunity cost", namely the loss of utility it imposes on all other FLSPs by its presence. The second term on the right-hand side is the fairness-adjusted cost, which is charged at the end of each period after the actual FL frequency is realized and observed.

Considering both the achieved FL frequency and the payment, FLSP n's utility is therefore

$$u(s) = f_n^*(b_n(s)) - c_n(s) \tag{27}$$

The multi-bid auction-based inter-service bandwidth allocatio (MISBA) algorithm is summarized in Algorithm 2.

Algorithm 2 Multi-Bid Auction-Based Inter-Service Bandwidth Allocation (MISBA)

```
Input to NO: total bandwidth $B$, fairness parameter $\alpha$, number of bids $M$.
Input to FLSP $n$: FL service $n$ parameters $s_n^{\text{DT}}, r_n^{\text{DT}}, s_n^{\text{UT}}, r_n^{\text{UT}}$, and number of clients in FL service $n$ as $K_n$.
Each FLSP $n$ sends multi-bid $s_n = (s_n^1, \dots, s_n^M)$ where $s_n^m = (b_n^m, p_n^m)$ to NO
NO calculates the clearing price according to Eqn.(21) - Eqn.(24)
NO allocates the bandwidth $b_n$ to each FLSP $n$ according to Eqn.(25)
Each FLSP $n$ solves the intra-service problem with allocated bandwidth $b_n$
NO charges each FLSP according to Eqn. (26)
```

D. Incentives of Truthful Reporting

In the previous subsection, we assumed that the every FLSP truthfully submits its bid. Now, we prove that this assumption indeed "approximately" holds under the designed bandwidth allocation and charging rules. We first study the individual rationality of the designed mechanism.

Definition 2: A mechanism is said to be ***individual rational*** *if no FLSP can be worse off from participating in the auction than if it had declined to participate.*

Proposition 4: If FLSP n submits a truthful multi-bid s_n, then $u_n(s) \geq 0$.

Proof: By Lemma 1, it is straightforward to see that $g_n(b)$ has the following properties:

- $g_n(b)$ is differentiable and $g_n(0) = 0$
- $g'_n(b)$ is positive, non-increasing and continuous
- $\exists \gamma_n > 0$, $\forall b \geq 0$, $g'_n(b) = 0 \Rightarrow \forall \tilde{b} < b$, $g'_n(b) \leq g'_n(\tilde{b}) - \gamma_n(b - \tilde{b})$.

Therefore, $g_n(b)$ satisfies [Assumption 1, [5]]. According to [Property 10, [5]], we have

$$\sum_{j \neq n} \int_{b_j(s)}^{b_j(s_{-n})} \bar{q}_j(b) db \leq g_n(b_n(s_n, s_{-n})) \tag{28}$$

which is equivalent to $c_n(s_n, s_{-n}) \leq f^*_n(b_n(s_n, s_{-n}))$. Therefore, $u(s) \geq 0$. □

Next, we show that truthful reporting is approximately **incentive compatible**, i.e., a FLSP cannot do much better than simply revealing its true valuation.

Proposition 5: Consider any truthful multi-bid s_n for FLSP n, and any other multi-bid $\tilde{s}_n \neq s_n$, $\forall s_{-n}$, we have $u_n(s_n, s_{-n}) \geq u_n(\tilde{s}_n, s_{-n}) - \Delta_n$, where $\Delta_n = \max_{0 \leq m \leq M} \int_{d_n(p_n^{m+1})}^{d_n(p_n^m)} (q_n(b) - p_n^m) db$ with $p_n^{M+1} = q_n(0)$ and $p_n^0 = p_0$.

Proof: The proof follows [Proposition 2, [5]]. □

The above proposition shows that if FLSP n submits a truthful multi-bid s_n, then every other multi-bid $\tilde{s}_n$ necessarily corresponds to an increase of utility no larger than Δ_n. In other words, a truthful bidding brings FLSP n the best utility possible up to a gap Δ_n. Importantly, this value does not depend on the number of other FLSPs or the multi-bids they submit.

E. An Uniform Multi-Bidding Example

To conclude the multi-bid auction mechanism design, we illustrate a uniform multi-bidding approach as an example of how to decide the multi-bid of an individual FLSP. Instead of having the FLSP submit both prices and bandwidth requests, the NO can announce M prices $(p_n^1, \ldots, p_n^M)$ to FLSP n and let FLSP n report its requested bandwidth $(b_n^1, \ldots, b_M^M)$ at these price points. This way, the NO has a better control over how the FLSPs make multi-bids to avoid multi-bids that may result in a large Δ_n, which may reduce FLSP's incentives to truthfully report. Because the NO does not know the demand function of FLSP n, a natural approach is to uniformly distribute these M prices in the range $[p_0, p_n^{\max}]$ where $p_n^{\max}$ is the largest price at which the FLSP may still request a positive amount of bandwidth. Specifically,

$$\begin{aligned} p_n^{\max} &= p_n(0) = f_n^{*\prime}(0) \\ &= \left(\sum_{k=1}^{K_n} \alpha_{n,k}\right)^{-1} = \left(\sum_{k=1}^{K_n} (\frac{s_n^{\text{DT}}}{r_k^{\text{DT}}} + \frac{s_n^{\text{UT}}}{r_k^{\text{UT}}})\right)^{-1} \end{aligned} \tag{29}$$

Assume that the NO has prior knowledge $\underline{K}_n$, $\underline{s}_n^{\text{DT}}$, $\underline{s}_n^{\text{UT}}$, $\bar{r}_n^{\text{DT}}$ and $\bar{r}_n^{\text{UT}}$ on the lower/upper bounds on the parameters, then $p_n^{\max}$ can be upper bounded by

$$p_n^{\max} \leq \underline{K}_n^{-1} \cdot \left(\frac{\underline{s}_n^{\text{DT}}}{\bar{r}_n^{\text{DT}}} + \frac{\underline{s}_n^{\text{UT}}}{\bar{r}_n^{\text{UT}}}\right)^{-1} \triangleq \bar{p}_n^{\max} \tag{30}$$

Thus, the NO can set the uniform prices as

$$p_n^m = p_0 + m \cdot \frac{\bar{p}_n^{\max} - p_0}{M + 1}, \quad \forall m = \{1, \ldots, M\} \tag{31}$$

Note that there is an intrinsic trade-off on the choice of M. On the one hand, a larger M allows the pseudo-BDF and pseudo-MVF to more accurately reflect the true BDF and MVF at an increased complexity and signaling overhead. On the other hand, a smaller M makes multi-biding easier but the discrepancy between the pseudo functions and the true functions will introduce a larger performance loss.

VI. Simulations

In this section, we conduct simulations to evaluate the performance of the proposed methods. Since our algorithm does not touch the actual FL learning procedure, it will not affect the FL performance in terms of accuracy or function loss in convergence, or even the number of learning rounds to meet a pre-selected convergence criterion. Our algorithm will only affect the absolute time length of a learning round and hence the absolute time (i.e. wall clock time) to converge. For this reason, using simulated FL convergence traces is sufficient to evaluate our algorithm.

A. Simulation Setup

The simulated wireless network has a total bandwidth of $B = 10$ MHz. The period length is set as $T = 20$ s. The number of clients of a FL service is drawn from a Gaussian distribution with mean 25. In every period, a new FL task may start following a scheduled plan, which is defined by a Poisson distribution with the mean interval p_{arrive}. By tuning p_{arrive}, we adjust the FL service demand, and a smaller p_{arrive} will more likely lead to more concurrent FL services in a period as an FL service often lasts multiple periods. Each FL service has a pre-determined target training accuracy, and when the accuracy reaches the target, the FL service terminates and exits the wireless network. In our simulation, each FL service is considered as being converged after 2000 learning rounds. The clients' wireless channel gain is modeled as independent free-space fading where the average path loss is from a Gaussian distribution with different mean and variance in different circumstances. The variance of the complex white Gaussian channel noise is set as 10^{-12}. For each client, the local training time is uniformly randomly drawn from $[0.01, 0.05]$ sec. We fix the global aggregation time to be 1×10^{-5} sec. We consider typical neural network sizes in the range of $[0.2, 0.5]$ Mbits. The upload transmission power is uniformly randomly between 0.05 and 0.15 W, and the download transmission power is uniformly randomly between 0.1 and 0.3 W.

B. Convergence of DISBA in the Cooperative Case

We first illustrate the convergence behavior of DISBA in the cooperative FLSPs case in a representative period with 5 concurrent FL services. These services have 10, 12, 14, 16, 18 clients, respectively. In Figure 4, we show the computed FL frequency for each service before convergence. As Figure 5

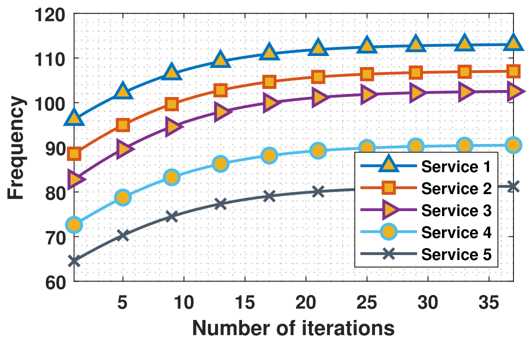

Fig. 4. Frequency of each service before convergence.

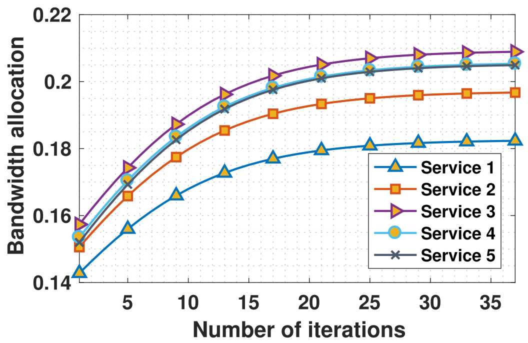

Fig. 5. Bandwidth of each service before convergence.

TABLE II

RESULTED BANDWIDTH ALLOCATION AND FREQUENCY OF EACH FL SERVICE (COOPERATIVE)

Service Index	Number of Clients	Bandwidth Ratio	Frequency
1	10	0.182	113
2	12	0.196	107
3	14	0.209	102.6
4	16	0.205	90.4
5	18	0.205	81.2

TABLE III

COMPUTATIONAL COMPLEXITY FOR THE COOPERATIVE PROVIDER CASE. THE TIME VALUES ARE MEASURED ON A DESKTOP COMPUTER WITH INTEL CORE I5-9400 2.9GHZ GPU AND 16GB MEMORY

Tolerated Gap	Step Size	# of Iterations	Time(s)
1e-3	0.1	131	0.332
1e-3	0.5	37	0.094
5e-3	0.1	72	0.169
5e-3	0.5	26	0.069

shows, the bandwidth allocation quickly converges to the optimal allocation for a convergence tolerance gap $\epsilon = 1e-3$. Eventually, the resulting FL frequencies of these FL services in this period are reported in Table II. We further show in Table III the computation time of DISBA for different values of the tolerance gap and the step size.

C. Fairness-Adjusted Multi-Bid Auction in the Selfish Case

We perform fairness-adjusted multi-bid auction in the same representative period as in the last subsection, with $M = 5$ and $\alpha = 0.5$. The pseudo-mBDFs of the FLSPs and the aggregated pseudo-mBDF are illustrated in Figures 6 and 7, respectively. The pseudo-MCP is also shown in Figure 7. Table IV reports the resulting bandwidth allocation and achieved FL frequency.

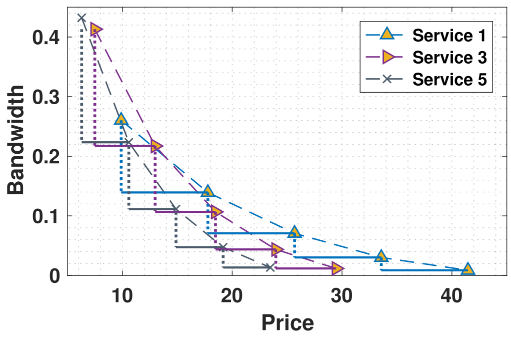

Fig. 6. Pseudo-mBDF of individual FL services.

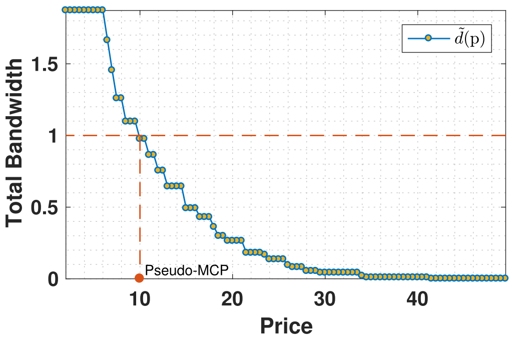

Fig. 7. Aggregated pseudo-mBDF and pseudo-MCP.

TABLE IV

OPTIMAL BANDWIDTH AND FREQUENCY OF EACH SERVICE (SELFISH)

Service Index	Number of Clients	Bandwidth Ratio	Frequency
1	10	0.164	105.82
2	12	0.177	99.52
3	14	0.217	105.46
4	16	0.218	94.4
5	18	0.223	86.56

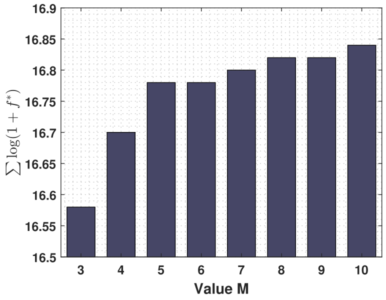

Fig. 8. Overall performance in the selfish FLSPs case with different M.

As we mentioned in Section V, there is a trade-off when selecting the number of bids M. In Figure 8, we demonstrate the overall performance by varying M. As can be seen, as M increases, the overall performance will increase while each FLSP needs to submit more bids to the server which will cause transmission delays and data backlogs.

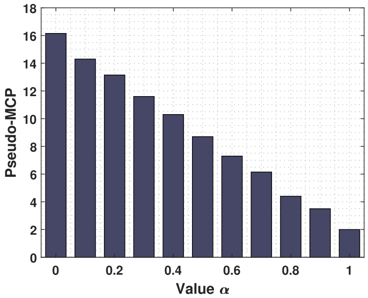

Fig. 9. Pseudo-MCP with different α.

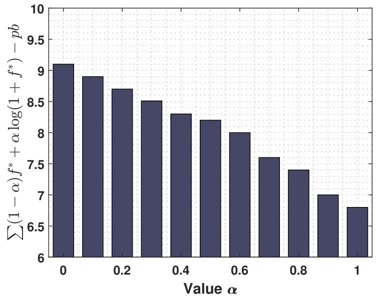

Fig. 10. Total utility with different α.

The parameter α plays an important role in the selfish FLSP case, which makes a tradeoff between efficiency and fairness. With a larger α, the whole system sees fairness as more important, and conversely, the whole system is more concerned with the overall efficiency. The market clearing price is reflected in Figure 9 and the overall utility is shown in Figure 10. With the increase of α, the market clearing price and the total utility will decrease, which can be treated as a compromise to achieving fairness between different FL services.

D. Performance Comparison

In the following experiments, we compare our proposed algorithms with three benchmark algorithms.

- **Equal-Client (EC)**: Bandwidth is equally allocated to the clients. Therefore, each client gets a bandwidth of $B/\sum_n K_n$.
- **Equal-Service (ES)**: Bandwidth is equally allocated to the FL services. That is, each FL service gets a bandwidth of B/N. However, each FLSP still performs the optimal intra-service bandwidth allocation among its clients.
- **Proportional (PP)**: Each FL service obtains a bandwidth that is proportional to the number of its client. That is, FL service n obtains a bandwidth of $\frac{K_n}{\sum_j K_n}B$. This bandwidth is further allocated among its clients following the optimal intra-service bandwidth allocation.

We start by comparing the proposed algorithms with benchmarks in the per-period setting. The overall performance is shown in Figure 11. In this setting, there are five FL services with a random number of clients drawn from a Gaussian distribution with mean 20 and variance 10 and random channel conditions drawn from a Gaussian distribution with mean 85 and variance 15, and the result is averaged over 20 runs. As can be seen, DISBA for the cooperative case (labeled as Coop) has the best performance, and the MISBA for the selfish case (labeled as Self) also outperforms the other benchmarks. Although ES and PP also solve the optimal intra-service bandwidth allocation, the heterogeneity of the client number and channel conditions render them suboptimal for the two-level problem.

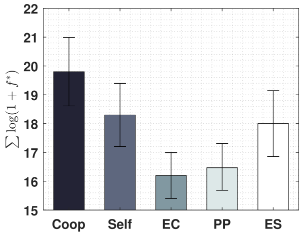

Fig. 11. Per-period FL performance of different algorithms.

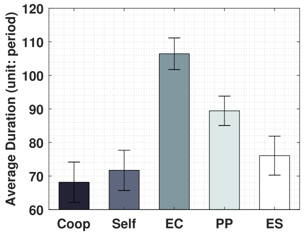

Fig. 12. Average duration period of FL services.

Because FL is a long-term process, we further investigate the long-term performance of the proposed algorithms. In the long-term setting, 10 FL services join the wireless network at different times controlled by the p_{arrive}-parameterized Poisson process and the FL service will be removed from the wireless network when it has converged. Although the convergence of FL is complexly affected by many factors including the adopted FL algorithm, dataset and the selected clients, we consider that each of these 10 FL services requires 2000 FL rounds to converge, which is a typical value observed in the literature [3] to reach convergence. This way provides a meaningful comparison of the algorithms in a controlled environment.

Figure 12 illustrates the average duration (in terms of the number of periods) of all FL services by running different algorithms for $p_{\text{arrive}} = 5$, where the client number of a FL service is drawn from a Gaussian distribution with mean 25 and variance 15 and the channel condition of a FL service is drawn from a Gaussian distribution with mean 85 and variance 15. The results are averaged over 20 runs. We can see that the proposed algorithms achieve the smallest average duration compared to the benchmarks, confirming their fast FL convergence even in the long-run.

Next, we study the impact of the client number heterogeneity (which reflects the FL service scale heterogeneity) on the performance of different algorithms. To this end, the client number of a FL service is drawn from a Gaussian distribution with mean 25 and we change the variance between 0 and 15 to adjust the heterogeneity degree. The result is shown in Figure 13: as the variance increases (i.e. a higher degree of heterogeneity), the mean of the average duration decreases, while the standard deviation of average duration

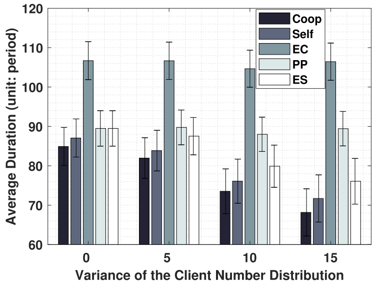

Fig. 13. Impact of client number heterogeneity.

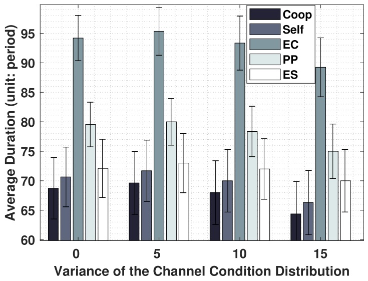

Fig. 14. Impact of the channel condition heterogeneity.

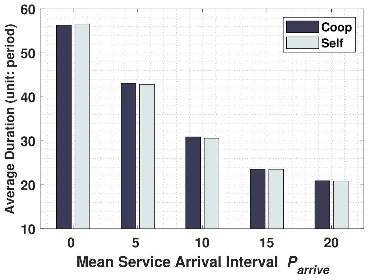

Fig. 15. Average duration with varying P_{arrive}.

increases. This is understandable because a higher degree of heterogeneity causes wireless bandwidth to be more unevenly distributed among the FL services, thereby degrading the overall FL performance. Notably, the performance gain of our proposed algorithms increases as the variance increases, which demonstrates the superior ability of our algorithms to handle the heterogeneous case.

Furthermore, we also investigate the impact of the channel condition heterogeneity on the FL performance. In these simulations, the average channel condition of a FL service is drawn from a Gaussian distribution with mean 85 and we change the variance between 0 and 15 to adjust the heterogeneity degree. The channel conditions of clients of this FL are further drawn from a Gaussian distribution with a mean being the instantiated average channel condition. In Figure 14, we observe a similar phenomenon as in Figure 13, which further confirms the advantage of adopting our proposed algorithms.

Finally, we study the influence of the arrival interval parameter p_{arrive} on the resulting average FL duration. in Figure 15, with the increase of p_{arrive}, the average duration of the FL services decreases. This is because when p_{arrive} is small, many FL services pile up and co-exist in the wireless network, thereby reducing the wireless bandwidth an individual FL service can receive.

VII. Conclusion

This paper studied a bandwidth allocation problem for multiple FL services in a wireless network, a new topic in the literature. The considered problem consists of two interconnected subproblems, intra-service resource allocation, and inter-service resource allocation. By solving these problems, we optimally allocate bandwidth resources to multiple FL services and their corresponding clients to speed up the training process and meanwhile guarantee fairness for both cooperative and selfish FLSPs cases.

We note that FL is captured in our work via its unique workflow rather than its specific learning procedure. Precisely, FL is an iterative process where every round involves four steps (DT, LC, UT and GC), and the slowest client determines the length of a round and hence the speed of learning. Such a workflow results in a new two-level bandwidth allocation problem where bandwidth needs to be allocated among not only clients but also the simultaneous services to ensure an efficient and fair system operation. We did not consider the learning-specific factor in our bandwidth allocation problem because it does not matter in determining the time needed to complete one learning round. We are more inclined to consider this as a merit of our approach rather than a drawback, because it decouples the wireless resource allocation from the learning algorithm design. Thus our approach is able to handle a wide range of FL algorithms that adopt the common workflow. In particular, one does not have to re-design the entire wireless system and the bandwidth allocation scheme for every single FL algorithm. However, tailoring the bandwidth allocation design to specific learning factors may further improve the FL performance, likely with added complexity. This could be an interesting future research direction to explore. In addition, bandwidth allocation can be further performed in conjunction with client selection to deal with cases where some clients experience extremely low computation capacity and/or extremely poor wireless channel condition. This could be another interesting future research topic.

References

[1] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, “Wireless network intelligence at the edge,” *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[2] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, “Toward an intelligent edge: Wireless communication meets machine learning,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.

[3] J. Konevcný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” 2016, *arXiv:1610.05492*. [Online]. Available: https://arxiv.org/abs/1610.05492

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, PMLR, 2017, pp. 1273–1282.

[5] P. Maille and B. Tuffin, “Multi-bid auctions for bandwidth allocation in communication networks,” in *Proc. IEEE INFOCOM*, vol. 1, Mar. 2004, pp. 1–12.

[6] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, “SCAFFOLD: Stochastic controlled averaging for federated learning,” 2019, *arXiv:1910.06378*. [Online]. Available: https://arxiv.org/abs/1910.06378

[7] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” 2018, *arXiv:1812.06127*. [Online]. Available: https://arxiv.org/abs/1812.06127

[8] F. Haddadpour and M. Mahdavi, “On the convergence of local descent methods in federated learning,” 2019, *arXiv:1910.14425*. [Online]. Available: http://arxiv.org/abs/1910.14425

[9] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on non-IID data,” 2019, *arXiv:1907.02189*. [Online]. Available: https://arxiv.org/abs/1907.02189

[10] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-IID data,” 2018, *arXiv:1806.00582*. [Online]. Available: https://arxiv.org/abs/1806.00582

[11] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, “Robust and communication-efficient federated learning from non-iid data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2019.

[12] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4424–4434.

[13] C. Fung, C. J. M. Yoon, and I. Beschastnikh, “Mitigating sybils in federated learning poisoning,” 2018, *arXiv:1808.04866*. [Online]. Available: https://arxiv.org/abs/1808.04866

[14] H. Kim, J. Park, M. Bennis, and S.-L. Kim, “Blockchained on-device federated learning,” *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[15] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” 2019, *arXiv:1902.00146*. [Online]. Available: https://arxiv.org/abs/1902.00146

[16] T. Li, M. Sanjabi, A. Beirami, and V. Smith, “Fair resource allocation in federated learning,” 2019, *arXiv:1905.10497*. [Online]. Available: https://arxiv.org/abs/1905.10497

[17] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[18] W. Y. B. Lim *et al.*, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[19] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[20] C. Shen, J. Xu, S. Zheng, and X. Chen, “Resource rationing for wireless federated learning: Concept, benefits, and challenges,” 2021, *arXiv:2104.06990*. [Online]. Available: https://arxiv.org/abs/2104.06990

[21] T. Chen, G. Giannakis, T. Sun, and W. Yin, “Lag: Lazily aggregated gradient for communication-efficient distributed learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5050–5060.

[22] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” 2017, *arXiv:1712.01887*. [Online]. Available: https://arxiv.org/abs/1712.01887

[23] A. F. Aji and K. Heafield, “Sparse communication for distributed gradient descent,” 2017, *arXiv:1704.05021*. [Online]. Available: https://arxiv.org/abs/1704.05021

[24] L. Liu, J. Zhang, S. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[25] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[26] G. Zhu, Y. Wang, and K. Huang, “Broadband analog aggregation for low-latency federated edge learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[27] S. Wang *et al.*, “Adaptive federated learning in resource constrained edge computing systems,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 1205–1221, Jun. 2019.

[28] Y. Zhan, P. Li, and S. Guo, “Experience-driven computational resource allocation of federated learning by deep reinforcement learning,” in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2020, pp. 234–243.

[29] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, “Federated learning over wireless networks: Optimization model design and analysis,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 1387–1395.

[30] J. Xu and H. Wang, “Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2020.

[31] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, “Energy-efficient radio resource allocation for federated edge learning,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[32] W. Shi, S. Zhou, and Z. Niu, “Device scheduling with fast convergence for wireless federated learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[33] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[34] M. Chen, H. V. Poor, W. Saad, and S. Cui, “Convergence time optimization for federated learning over wireless networks,” 2020, *arXiv:2001.07845*. [Online]. Available: https://arxiv.org/abs/2001.07845

[35] M. N. H. Nguyen, N. H. Tran, Y. K. Tun, Z. Han, and C. S. Hong, “Toward multiple federated learning services resource sharing in mobile edge networks,” 2020, *arXiv:2011.12469*. [Online]. Available: https://arxiv.org/abs/2011.12469

[36] L. Massoulié and J. Roberts, “Bandwidth sharing: Objectives and algorithms,” in *Proc. 18th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 3, Mar. 1999, pp. 1395–1403.

[37] D. P. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[38] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y.-C. Liang, “Joint service pricing and cooperative relay communication for federated learning,” in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2019, pp. 815–820.

[39] Y. Sarikaya and O. Ercetin, “Motivating workers in federated learning: A stackelberg game perspective,” *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.

[40] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, “Incentive design for efficient federated learning in mobile networks: A contract theory approach,” in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, Aug. 2019, pp. 1–5.

[41] T. H. T. Le, N. H. Tran, Y. K. Tun, Z. Han, and C. S. Hong, “Auction based incentive design for efficient federated learning in cellular wireless networks,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[42] D. P. Bertsekas, “Nonlinear programming,” *J. Oper. Res. Soc.*, vol. 48, no. 3, p. 334, 1997.

[43] W. Vickrey, “Counterspeculation, auctions, and competitive sealed tenders,” *J. Finance*, vol. 16, no. 1, pp. 8–37, 1961.

Jie Xu (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2008 and 2010, respectively, and the Ph.D. degree in electrical engineering from UCLA in 2015. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Miami. His research interests include mobile edge computing/intelligence, machine learning for networks, and network security. He received the NSF CAREER Award in 2021.

Heqiang Wang received the B.S. degree in electrical and computer engineering from the University of Kentucky in 2016 and the M.S. degree in electrical and computer engineering from the University of Connecticut in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Miami. His research interests include communication efficiency and client scheduling federated learning problems.

Lixing Chen received the B.S. and M.S. degrees from the College of Information and Control Engineering, China University of Petroleum, Qingdao, China, in 2013 and 2016, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Miami in 2020. He is currently an Assistant Professor with Shanghai Jiao Tong University. His research interests include mobile edge computing and machine learning for networks.