Client Selection and Bandwidth Allocation in Wireless Federated Learning Networks: A Long-Term Perspective

Jie Xu, *Member, IEEE*, and Heqiang Wang

***Abstract*— This paper studies federated learning (FL) in a classic wireless network, where learning clients share a common wireless link to a coordinating server to perform federated model training using their local data. In such wireless federated learning networks (WFLNs), optimizing the learning performance depends crucially on how clients are selected and how bandwidth is allocated among the selected clients in every learning round, as both radio and client energy resources are limited. While existing works have made some attempts to allocate the limited wireless resources to optimize FL, they focus on the problem in individual learning rounds, overlooking an inherent yet critical feature of federated learning. This paper brings a new long-term perspective to resource allocation in WFLNs, realizing that learning rounds are not only temporally interdependent but also have varying significance towards the final learning outcome. To this end, we first design data-driven experiments to show that different temporal client selection patterns lead to considerably different learning performance. With the obtained insights, we formulate a stochastic optimization problem for joint client selection and bandwidth allocation under long-term client energy constraints, and develop a new algorithm that utilizes only currently available wireless channel information but can achieve long-term performance guarantee. Experiments show that our algorithm results in the desired temporal client selection pattern, is adaptive to changing network environments and far outperforms benchmarks that ignore the long-term effect of FL.**

***Index Terms*— Federated learning (FL), client selection, resource allocation, wireless networks.**

I. Introduction

MOBILE devices nowadays generate a massive amount of data each day. This rich data has the potential to power a wide range of machine learning (ML)-based applications, such as learning the activities of smart phone users, predicting health events from wearable devices or adapting to pedestrian behavior in autonomous vehicles. 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 each mobile device. The aim of "federated learning" (FL) [1] is to enable mobile devices to collaboratively learn a shared ML model with the coordination of a central server while keeping all the training data on device, thereby decoupling the ability to do ML from the need to upload/store the data in the cloud.

This paper focuses on FL in a classic wireless network setting where the clients, e.g., mobile devices, share a common wireless link to the server. We call this system a *wireless federated learning network* (WFLN). The network operates for a number of learning rounds as follows: 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 via the wireless link; the server then aggregates the local updates to improve the shared model. Similar to a traditional throughput-oriented wireless network, the limited wireless network resources require the WFLN to determine in each round which clients access the wireless channel to upload the model updates and how much bandwidth is allocated to each client. However, due to the specific application in consideration, namely FL, the resource allocation objective and consequently the outcome can be very different from, e.g., throughput maximization.

Optimizing WFLNs faces unique challenges compared to optimizing either FL or the traditional wireless networks. On the one hand, the wireless network sets resource constraints on performing FL as the finite wireless bandwidth limits the number of clients that can be selected in each round, and the selection must be adaptive to the highly variable wireless channel conditions. On the other hand, FL is likely to change the way wireless networks should be optimized as model training is a complex *long-term* process where decisions across rounds are interdependent and collectively decide the final training performance. Further, since mobile devices often have finite energy budgets due to, e.g., a finite battery, the number of rounds each individual mobile device can participate during the entire course of FL is also limited. An extremely crucial yet largely overlooked question is: does learning in different rounds contribute the same or differently to the final learning outcome and hence should the wireless resources be allocated discrepantly across rounds? Without a good understanding of its answer, conventional wireless network optimization approaches that treat each time slot independently and equally may lead to considerably suboptimal FL performance.

Manuscript received April 15, 2020; revised August 27, 2020; accepted October 9, 2020. Date of publication October 22, 2020; date of current version February 11, 2021. This work was supported in part by the NSF under Grant ECCS-2033681, Grant CNS-2006630, and Grant ECCS-2029858. The associate editor coordinating the review of this article and approving it for publication was K. Choi. *(Corresponding author: Jie Xu.)*

The authors are with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146 USA (e-mail: jiexu@miami.edu).

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

Digital Object Identifier 10.1109/TWC.2020.3031503

1536-1276 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

This paper aims to formalize this fundamental problem of client selection and bandwidth allocation in WFLNs and derive critical knowledge to enable the efficient operation of these networks. We study how resources (i.e., bandwidth and energy) should be allocated among clients in each learning round as well as *across* rounds given finite client energy budgets in a volatile network environment. Our main contributions are summarized as follows.

- While existing works [2], [3] have shown that selecting more clients generally improves the FL performance, there is little understanding of how this improvement depends on different learning rounds. If the total number of selected clients during the entire course of FL is fixed, should client selection be uniform across rounds or biased toward the early/late FL rounds? Through experiments, we show in two representative ML tasks, namely image classification and text generation, a general "later-is-better" phenomenon in terms of higher accuracy, lower training loss and better robustness. This finding, to our best knowledge, is the first that relates the temporal client selection pattern to the final FL performance.
- With "later-is-better", we formulate a *long-term* client selection and bandwidth allocation problem for a finite number of FL rounds under finite energy constraints of individual clients. To deal with time-varying but unpredictable wireless channel conditions, a new Lyapunov-based online optimization algorithm called OCEAN is proposed, where in each FL round, a new Set Expansion Algorithm (SEA) is developed to efficiently solve the per-round problem. We prove that OCEAN demonstrates an $[O(1/V), O(\sqrt{V})]$ learning-energy tradeoff where V is an algorithm parameter.
- We characterize the structure of the client selection and bandwidth allocation outcome. In each FL round, clients are selected according to a priority metric, which is the ratio of the client's current energy deficit queue length and its current wireless channel state. However, among the selected ones, more bandwidth is allocated to clients with a lower priority (i.e., worse channel and larger energy deficit queue length). This is in stark contrast to a traditional throughput-oriented wireless network where more bandwidth is allocated to clients with a better channel condition in order to maximize throughput.

II. Related Work

Since the proposal of FL [1], [4], a lot of research effort has been devoted to tackling various challenges in this new distributed machine learning framework, including developing new optimization and model aggregation algorithms [5]–[7], handling non-i.i.d. and unbalanced datasets [8]–[10], and preserving model privacy [11]–[15] etc. Among these challenges, improving the communication efficiency of FL has been a key challenge due to the tension between uploading a large amount of data for model aggregation and the limited network resource to support this transmission. In this regard, a strand of literature focuses on modifying the FL algorithm itself to reduce the communication burden on the network, e.g., updating clients with significant training improvement [16], compressing the gradient vectors via quantization [17], or accelerating training using sparse or structured updates [1], [18]. Hierarchical FL networks [19] have also been proposed where multiple edge servers perform partial model aggregation first, whose outputs are further aggregated by a cloud server. Recognizing the unique physical property of wireless transmission, [2], [20]–[22] propose analog model aggregation over the air, provided that a very stringent synchronization is available.

As wireless networks are the envisioned main deployment scenario of FL, how to optimally allocate the limited bandwidth and energy resources for FL has also received much attention. Many existing works [23]–[26] study the inherent trade-off between local model update and global model aggregation, e.g., to adapt the frequency of global aggregation [23] or to optimize uplink transmission power/rate and the local update CPU frequency [25], [26]. In all these works, all clients participate in every FL round. Although both empirical studies [2], [3] and theoretical analysis [27] show that including more clients improves the FL convergence speed, the limited bandwidth of wireless networks cannot support many clients to upload their local updates at the same time. For FL at scale, client scheduling policies, which select only a subset of clients in every round, are necessary. In [28], the convergence performance of FL under three basic scheduling policies, namely random, round-robin and proportional fair, is analyzed. Different types of joint bandwidth allocation and client scheduling policies, e.g., [3], [29]–[33], have been proposed to either minimize the learning loss or the training time. However, their optimization problems are formulated by considering individual FL rounds separately or treating every FL round equally, and hence the same network resources are allocated across learning rounds. Our paper differs from these works in that we explicitly consider the varying significance of FL rounds and study a long-term bandwidth allocation and client selection problem under long-term energy constraints and with uncertain wireless channel information.

In our experiments, an empirical "later-is-better" convergence phenomenon is observed. A thorough theoretical convergence analysis of this phenomenon seems a very difficult task, and likely would require techniques beyond those used in existing convergence analysis for FL, e.g., [34], [35], as they do not explicitly differentiate the effects of different learning rounds. A possible direction is to relate the approximation error of the gradient estimate to the number of selected clients in every learning round, which shares a similar principle to stochastic gradient descent with adaptive batches [36]. The theoretical convergence analysis is beyond the scope of the current paper, and represents an important future work direction.

III. System Model

Consider a WFLN with one server and K clients, indexed by the set $\mathcal{K} = \{1, \ldots, K\}$. Each participating client $k \in \mathcal{K}$ has a local dataset $\mathcal{D}_k$. In the supervised learning case, $\mathcal{D}_k$ defines the collection of data samples given as a set of input-output

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

pairs $\{x_i, y_i\}_{i=1}^{D_k}$, where $x_i \in \mathbb{R}^d$ is a d-dimensional input feature vector, and $y_i \in \mathbb{R}$ is the ground-truth output label. This data can be generated through the usage of the client via mobile applications and can be employed for various ML tasks, e.g., user activity prediction or health event prediction.

FL iterates between two steps: 1) the server updates the global model by aggregating local models transmitted over a multi-access channel by the clients; 2) the clients update their local models using the global model broadcasted by the server. We call each iteration a *learning round*. WFLN has to decide in each round which clients upload their local model updates depending on their wireless channel condition and remaining battery to maximize the learning performance. We use $a_k^t \in \{1, 0\}$ to denote whether or not client k is selected in round t, and $\boldsymbol{a}^t = (a_1^t, \ldots, a_K^t)$ collects the overall client selection decisions.

A. Client Energy Consumption

For a selected client k in round t (i.e., $a_k^t = 1$), it incurs energy consumption due to local training and uploading the local updates to the edge server via the wireless channel. For each client k, let E_k^{tr} denote its local training energy consumption in every round, which depends on its computing architecture, hardware and dataset. We consider a specific wireless multi-access scheme, i.e., orthogonal frequency-division multiple access (OFDMA) for local model uploading with a total bandwidth B. Let $b_k^t \in [0, 1]$ be the bandwidth allocation ratio for client k in round t, and hence its allocated bandwidth is $b_k^t B$. Let $\boldsymbol{b}^t = (b_1^t, \ldots, b_K^t)$. Bandwidth allocation must satisfy $\sum_{k\in\mathcal{K}} b_k^t = 1, \forall t$. Clearly, if $a_k^t = 0$, namely client k is not selected in round t, then it is the best not to allocate any bandwidth to this client, i.e., $b_k^t = 0$. On the other hand, if $a_k^t = 1$, then we require that at least a minimum bandwidth b_{min} is allocated to client k, i.e., $b_k^t \geq b_{min}$. This is because practical systems cannot assign an arbitrarily small bandwidth to an individual client due to, e.g., a finite resource block size. In addition, a close-to-zero bandwidth allocation will require an extremely high transmit power and hence result in an extremely high energy consumption to achieve a target transmission rate. To make the problem feasible, we assume $b_{min} \leq 1/K$.

Let p_k^t denote the transmission power (in Watt/Hz) of client k in round t. The achievable rate (in bit/s), denoted by r_k^t, can be written according to the Shannon's formula as

$$r_k^t = b_k^t B \log_2 \left(1 + \frac{p_k^t (h_k^t)^2}{N_0}\right) \tag{1}$$

where N_0 is the variance of the complex white Gaussian channel noise and h_k^t is the channel state of client k in round t. Let L denote the data size of the adopted machine learning model (in bit), then the time needed to upload the local model update to the edge server is $\tau_k = L/r_k^t$. For a target upload time deadline $\bar{\tau}$, the required transmission power can be derived using (1) and hence the transmission energy consumption of client k is

$$E^{\text{tx}}(a_k^t, b_k^t | h_k^t) = \frac{\bar{\tau} N_0 B b_k^t}{(h_k^t)^2} \left(2^{\frac{L}{\bar{\tau} B b_k^t}} - 1\right) a_k^t \tag{2}$$

The total energy consumption of client k is therefore

$$E(a_k^t, b_k^t | h_k^t) = \left[\frac{\bar{\tau} N_0 B b_k^t}{(h_k^t)^2} \left(2^{\frac{L}{\bar{\tau} B b_k^t}} - 1\right) + E_k^{\text{tr}}\right] a_k^t \tag{3}$$

B. System Learning Performance

Existing works [2], [3] have shown that the FL performance can be improved by selecting more clients in each round. However, selecting more clients is not always possible if each client is subject to a long-term energy constraint due to, e.g., a finite battery: selecting more clients in early learning rounds depletes the battery of the clients and hence fewer clients can be selected in later learning rounds. Hence, even with the same average number of selected clients, the temporal pattern can be considerably different, yet there is little understanding of how the temporal pattern affects the final FL outcome. In this subsection, we design experiments for two tasks (i.e., image classification and text prediction) on three datasets (i.e., MNIST, CIFAR-10 and Shakespeare) to show that the temporal client pattern indeed has a considerable impact on the final FL performance. In all experiments, we consider FL on 20 clients and investigate three temporal selection patterns: **Uniform** – in each round, 10 clients are randomly selected to participate in FL; **Ascend** – the number of selected clients gradually increases from 1 to 20 with an average number of 10 clients selected per round; **Descend** – the number of selected clients gradually decreases from 20 to 1 with an average number of 10 clients selected per round. All experiments are conducted using the TensorFlow Federated (TFF) framework [37] and the datasets have been pre-processed as non-i.i.d. federated datasets.

Figure 2 illustrates the test accuracy of MNIST, CIFAR-10 and Shakespeare, respectively. As can be seen, although the average number of selected clients is the same, different temporal patterns result in different test accuracy by the end of training. In particular, **Ascend** results in the best performance compared to **Uniform** and **Descend**. There is a good reason behind such a "later-is-better" phenomenon: early learning rounds are "easy" rounds where the learning performance is less sensitive to the number of selected clients. Therefore, even if **Ascend** selects fewer clients in the early rounds, learning speed is minimally affected. However, the later learning rounds are the more "difficult" rounds, and to push accuracy even higher requires more clients to update the shared model using their data. In fact, not only **Ascend** wins in training loss and accuracy, but also it is often much more robust as the standard deviation is much smaller. This is again because more clients participate in model updating towards the end of learning, which can smooth out abrupt changes in the learned model of individual clients.

Based on the empirical "later-is-better" observation, we introduce the following metric to describe the FL performance in round t:

$$U^t(\boldsymbol{a}^t) = \eta^t \sum_{k=1}^{K} D_k a_k^t \tag{4}$$

where η^t is a temporal weight to capture the varying significance of selecting more clients in different learning rounds.

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

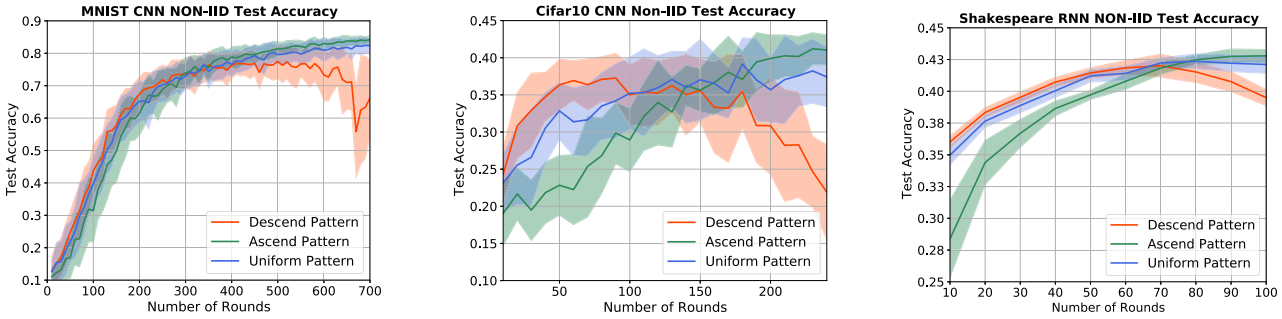

Fig. 1. Test accuracy on MNIST, CIFAR-10 and Shakespeare datasets.

An increasing sequence of η^t often results in better FL performance as more clients are likely to be selected in later rounds of learning.

We note, however, although the above metric will facilitate our subsequent resource allocation, it does not exactly characterize the FL speed or accuracy, which is extremely difficult, if not impossible, to model due to the complex and non-convex nature of many ML algorithms.

C. Problem Formulation

As we emphasize the long-term performance and final outcome of FL, the goal is to maximize the weighted sum of selected clients defined in (4) for a total number of T learning rounds while satisfying the long-term energy budget constraints of individual clients, through joint client selection $\boldsymbol{a}^t$ and bandwidth allocation $\boldsymbol{b}^t$ in every round $t = 0, \ldots, T-1$. Although the performance metric defined in (4) is artificial, we will relate it to the actual FL performance in experiments. Formally, the problem that we aim to solve is

$$\mathbf{P1} \quad \max_{\boldsymbol{a}^0, \boldsymbol{b}^0, \ldots, \boldsymbol{a}^{T-1}, \boldsymbol{b}^{T-1}} \sum_{t=0}^{T-1} U^t(\boldsymbol{a}^t) \tag{5}$$

$$\text{s.t.} \sum_{t=0}^{T-1} E(a_k^t, b_k^t | h_k^t) \leq H_k, \forall k \tag{6}$$

$$b_{min} \leq b_k^t \leq 1, \quad \forall k, \forall t, \quad \sum_{k=1}^{K} b_k^t = 1, \forall t \tag{7}$$

$$a_k^t \in \{0, 1\}, \quad \forall k, \forall t \tag{8}$$

Constraint (6) requires that the total energy consumption over the T rounds for each client k does not exceed an energy budget H_k (e.g., battery capacity or energy limit set by the client). Constraint (7) is the feasibility condition on the bandwidth allocation. Constraint (8) is the feasibility condition on the client selection.

So far we have formulated a long-term optimization problem for client selection and bandwidth allocation in WFLNs. However, several challenges impede the derivation of the optimal solution to **P1**. The first is the lack of future information: optimally solving **P1** requires complete offline information (i.e., channel conditions) over the entire FL period (i.e., T learning rounds) that is very difficult to accurately predict in advance. Furthermore, **P1** belongs to mixed-integer nonlinear programming and is difficult to solve, even if the long-term future information is accurately known a priori. Thus, these challenges demand an online approach that can efficiently make joint client selection and bandwidth allocation decisions without foreseeing the far future.

D. Offline Benchmark: R-Round Lookahead Algorithm

Before we propose the online algorithm, we first introduce an offline algorithm with R-round lookahead information (i.e., the channel information in the next R learning rounds are *assumed* to be known) as a benchmark. Specifically, we divide the entire FL period into $M \geq 1$ frames, each having $R \geq 1$ learning rounds such that $T = MR$, and present the following problem formulation:

$$\mathbf{P2}: \max_{\boldsymbol{a}^0, \boldsymbol{b}^0, \ldots, \boldsymbol{a}^{T-1}, \boldsymbol{b}^{T-1}} \sum_{t=mR}^{(m+1)R-1} U^t(\boldsymbol{a}^t) \tag{9}$$

$$\text{s.t.} \sum_{t=mR}^{(m+1)R-1} E(a_k^t, b_k^t | h_k^t) \leq H_k/M, \quad \forall k$$
$$\text{Constraints (7), (8)} \tag{10}$$

Essentially, **P2** defines a family of offline algorithms parameterized by the lookahead window size R. Clearly, there exists at least one sequence of joint client selection and bandwidth allocation decisions that satisfies all constraints of **P2** (e.g., no client is selected in any round in each frame). We denote the optimal learning performance for the m-th frame by U_m^*, for $m = 0, \ldots, M-1$, considering all the decisions that satisfy the constraints and have perfect information over the frame. Thus, the optimal long-term learning performance achieved by the oracle's optimal R-round lookahead algorithm is given by $\sum_{m=0}^{M-1} U_m^*$.

We note that because of the assumed lookahead information, the R-round lookahead algorithms are impractical (unless $R = 1$). The purpose of introducing these algorithms is only to use them as a benchmark for our practical online algorithm to be proposed in the next section.

IV. Online Client Selection and Bandwidth Allocation

In this section, we develop the Online Client sElection and bAndwidth allocatioN algorithm, called OCEAN, and then characterize its structural properties. We also prove that it

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

is efficient compared to the optimal offline algorithm with R-round lookahead information. Our OCEAN algorithm has an input parameter R related to the lookahead window size, so the R-round lookahead algorithms are natural benchmarks.

A. The OCEAN Algorithm

A major challenge of directly solving **P1** is that the long-term energy constraint of the clients couples the client selection and bandwidth allocation decisions across different learning rounds: selecting more clients in the current round reduces the bandwidth allocated to each individual client, thereby increasing the energy consumption of these clients; furthermore, more energy consumption in the current round potentially reduces the energy budget available for future FL rounds, and yet the decisions have to be made without foreseeing the future. To address this challenge, we leverage the Lyapunov technique and construct a virtual energy deficit queue $q_k(t)$ for each client k to guide the client selection and bandwidth allocation decisions to follow the long-term energy constraint. The virtual energy queue of client k starts with $q_k(0) = 0, \forall k$, and is updated at the end of each round t as follows

$$q_k(t+1) = [E(a_k^t, b_k^t|h_k^t) - H_k/T + q_k(t)]^+ \tag{11}$$

where $[\cdot]^+ = \max\{\cdot, 0\}$. Hence, $q_k(t)$ is the queue length indicating the deviation of the current energy consumption of client k from its long-term energy constraint H_k. Let $\boldsymbol{q}(t) = (q_1(t), q_2(t), \ldots, q_K(t))$ collect the energy deficit queues for all clients.

Algorithm 1 OCEAN

1: **Input**: $q_k(0) = 0, \forall k$ and R
2: **for** $t = 1, 2, \ldots, T$ **do**
3: **if** $t = mR, \forall m = 1, \ldots, M-1$ **then**
4: $q_k(t) \leftarrow 0, \forall k$ and $V \leftarrow V_m$
5: **end if**
6: Observe the current channel state $h_k^t, \forall k$
7: Solve **P3**
8: Update energy queue according to (11)
9: **end for**

We now present OCEAN in Algorithm 1. OCEAN is purely online and requires only the currently available channel state information as inputs (i.e. $h_k(t), \forall k$). We use $V_0, V_1, \ldots, V_{M-1}$ to denote a sequence of positive control parameters to dynamically adjust the tradeoff between maximizing the number of selected clients and minimizing energy consumption over the M frames, each having R communication rounds. The importance of the control parameters will be revisited in Section IV.C. In every round t, we aim to solve the following per-round problem:

$$\textbf{P3} \ \max_{\boldsymbol{a}^t, \boldsymbol{b}^t} V \cdot U^t(\boldsymbol{a}^t) - \sum_{k=1}^{K} q_k(t) E(a_k^t, b_k^t|h_k^t) \tag{12}$$

$$\text{s.t. } (7), (8) \tag{13}$$

By considering the additional term $\sum_{k=1}^{K} q_k(t) E(a_k^t, b_k^t|h_k^t)$, the system takes into account the energy deficit of the clients during the current round's client selection and bandwidth allocation. As a consequence, when $q_k(t)$ is larger, minimizing the energy deficit is more critical. Thus, our algorithm works following the philosophy of "if violate the energy constraint, then use less energy", and the energy deficit queue maintained without foreseeing the future guides the system towards meeting the energy constraints of the clients. OCEAN decomposes the long-term optimization problem into a series of per-round problems **P3**. For a more rigorous derivation of this decomposition, please refer to the proof of Theorem 1. Now, to complete OCEAN, it remains to solve **P3**, which however is still very difficult.

B. Solving the Per-Round Problem

The per-round problem **P3** is a difficult mixed-integer problem. To see more clearly how the objective function depends on $\boldsymbol{a}^t$ and $\boldsymbol{b}^t$, we write it out and rearrange it as follows

$$\sum_{k=1}^{K} \left[V\eta^t D_k - q_k(t) \cdot \left(\frac{\bar{\tau} N_0 B b_k^t}{(h_k^t)^2} \left(2^{\frac{L}{\bar{\tau} B b_k^t}} - 1 \right) + E_k^{\text{tr}} \right) \right] a_k^t$$
$$\boldsymbol{e} \triangleq W(\boldsymbol{a}^t, \boldsymbol{b}^t) \tag{14}$$

Notice that a_k^t is a binary integer variable and b_k^t is a continuous variable in $[b_{\min}, 1]$. In general, mixed-integer problems are difficult to solve and often there is no polynomial-time optimal algorithm. In what follows, we develop an efficient algorithm to solve **P3**. To simplify the notations, we drop the index t in this subsection.

Our algorithm to solve **P3**, called SEA (Set Expansion Algorithm), incrementally adds clients into the selection set S based on a metric $\rho_k \triangleq \frac{q_k(t)}{(h_k^t)^2}$, which we call the *selection priority* (the lower value, the higher priority). Initially, all clients with $\rho_k = 0$ (which also means $q_k(t) = 0$ as $(h_k^t)^2$ is always positive) are added into S. We denote this initial set by S^0. Then, clients with $\rho_k > 0$ are added into S one by one in the ascending order of ρ_k, and for each possible selection set, the corresponding bandwidth allocation is computed by solving the following optimization problem

$$\textbf{P4} \ \max_{\{b_k^t\}_{k \in S - S^0}} \sum_{k \in S - S^0} \big(V\eta^t D_k$$
$$- \rho_k N_0 \bar{\tau} B b_k \left(2^{\frac{L}{\bar{\tau} B b_k}} - 1 \right) - q_k(t) E_k^{\text{tr}} \big) \tag{15}$$

$$\text{s.t.} \sum_{k \in S - S^0} b_k^t = 1 - |S^0| \cdot b_{min} \tag{16}$$

$$b_k \geq b_{min}, \quad \forall k \in S - S^0 \tag{17}$$

Let $\boldsymbol{b}^*(S)$ be the optimal bandwidth allocation for a given selection set S, and $W^*(S)$ be the optimal value. Clearly, for the initial set S^0, $W^*(S^0) = \eta^t |S^0|$ as $\rho_k = 0, \forall k \in S^0$. The number of selection sets that can possibly emerge following the above set expanding rule, which are collected in $\mathcal{S}$, is at most K. Finally, the implemented optimal selection is $S^* = \arg\max_{S \in \mathcal{S}} W^*(S)$ and the implemented optimal bandwidth allocation is $\boldsymbol{b}^* = \boldsymbol{b}^*(S^*)$.

Because there are K clients and in every iteration, one more client is added into S, we ensure that the algorithm only needs to solve at most K optimization problems **P4** to return

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Set Expansion Algorithm (SEA)

Input: $q_k(0) = 0, \forall k$
Rank the clients according to ρ. Hence we have $\rho_1 \leq \rho_2 \leq \ldots \leq \rho_K$
Set $S^0 = \{k : \rho_k = 0\}$, $S = S^0$, and $\mathcal{S} = \{S^0\}$.
for $k = |S^0| + 1, \ldots, K$ **do**
  Update $S = S \cup \{k\}$
  Solve **P4** and obtain $\boldsymbol{b}^*(S)$ and $W^*(S)$
  if $V\eta^t - \rho_k N_0 \tilde{\tau} B b_k^t \left(2^{\frac{L}{\tilde{\tau} B b_k^t}} - 1\right) < 0$ **then**
    Stop iteration
  else
    Add S to $\mathcal{S}$, i.e. $\mathcal{S} = \mathcal{S} \cup \{S\}$
  end if
end for
Find $S^* = \arg\max_{S \in \mathcal{S}} W^*(S)$
Return $\boldsymbol{a}^*$ where $a_k^* = \mathbf{1}\{k \in S^*\}, \forall k$ and $\boldsymbol{b}^* = \boldsymbol{b}^*(S^*)$

the optimal solution. In fact, we can reduce the number of times for solving **P4** by adding a termination condition: if for some S, its optimal bandwidth allocation $\boldsymbol{b}^*(S)$ results in $V\eta^t D_k - \rho_k N_0 \tilde{\tau} B b_k \left(2^{\frac{L}{\tilde{\tau} B b_k^t}} - 1\right) - q_k(t) E_k^{\text{tr}} < 0$ for the last added client k, then the algorithm stops adding more clients into the selection set. This termination condition can significantly reduce the number of convex optimization problems to be solved when K is large. The pesudocode of SEA is given in Algorithm 2.

Now, we analyze the optimality of SEA. To this end, we first define Δ-optimality.

Definition 1: A joint client selection and bandwidth allocation action $(\boldsymbol{a}^\dagger, \boldsymbol{b}^\dagger)$ is Δ-optimal if $W(\boldsymbol{a}^\dagger, \boldsymbol{b}^\dagger) \geq W(\boldsymbol{a}^*, \boldsymbol{b}^*) - \Delta$, where Δ is some positive constant and $(\boldsymbol{a}^*, \boldsymbol{b}^*)$ is the optimal joint client selection and bandwidth allocation action.

*Theorem 1: SEA returns an Δ-optimal solution to the per-round problem **P3** by solving at most K convex optimization problems, where $\Delta = K \cdot \max_{k_1,k_2}(V\eta|D_{k_1} - D_{k_2}| + |q_{k_1} E_{k_1}^{tr} - q_{k_2} E_{k_2}^{tr}|)$.*

Proof: See Appendix A. □

The following corollary is straightforward by relaxing the heterogeneity of dataset size and the local training energy consumption so that $\Delta = 0$.

*Corollary 1: Assume that all clients have the same local data size, i.e., $D_{k_1} = D_{k_2}, \forall k_1 \neq k_2$, and the local training energy consumption is negligible, i.e., $E_k^{tr} \to 0, \forall k$, then SEA returns the optimal solution to the per-round problem **P3** by solving at most K convex optimization problems.*

Remark: The complexity of SEA is determined by two factors: the number of times to solve **P4**; the time complexity to solve each **P4**. First, SEA solves **P4** at most K times, but the termination condition can significantly reduce the actual number of times to solve P4. Secondly, since **P4** is a convex optimization problem, many efficient algorithms [38] and mature software tools (such as CVX [39] and SciPy [40]) exist to solve it with low complexity. In addition, since SEA for round t can run while the parameter server is aggregating local models for round $t-1$, it has a minimal impact on the training complexity of FL. Numerical results are reported in Section VI. D.

C. Structural Results and Performance Analysis

In this subsection, we first investigate the structure of the optimal solution produced by SEA in every round, and then characterize the performance of OCEAN.

In Theorem 1, we have already proven a thresholding result on the client selection, namely only clients whose selection priority ρ_k is below a threshold are selected to participate in a FL round. Proposition 1 characterizes how bandwidth is allocated among the selected clients and their incurred energy consumption.

Proposition 1: In any learning round t, the allocated bandwidth $b_k^{t,*}$ of a selected client k and its weighted energy consumption $q_k(t)E_k^t(b_k^{t,*})$ are non-decreasing with ρ_k^t.

Proof: See Appendix B. □

Theorem 1 and Proposition 1 together show that a client with a smaller energy deficit $q_k(t)$ and a better channel condition $(h_k^t)^2$ (and hence a smaller ρ_k^t) is more likely to be selected to participate in the current FL round; however, among the selected clients, a client with a smaller ρ_k^t is allocated with less bandwidth. This is because although allocating more bandwidth to client k with a smaller ρ_k^t reduces the energy consumption and deficit of this client, it reduces the bandwidth that can be allocated to clients with larger ρ, which leads to even higher increased energy consumption and deficit of those clients. Moreover, in the optimal solution, the overall effect of energy deficit and consumption, namely $q_k(t)E_k^t(b_k^*)$, is still increasing in ρ_k^t.

Next, we prove the performance guarantee of OCEAN. The key idea of our proof relies on the drift-plus-penalty technique in Lyapunov analysis [41]. The biggest difference is that our formulation considers a finite number of T learning rounds, while standard Lyapunov analysis assumes $T \to \infty$. This leads to a different proof and result as shown in Theorem 2.

Theorem 2: For any $R \in \mathbb{Z}^+$ and $M \in \mathbb{Z}^+$ such that $T = MR$, when comparing OCEAN with the R-round lookahead algorithm, the following statements hold:

(a) The energy constraint of every client k is approximately satisfied with a bounded deviation:

$$\sum_{t=0}^{T} E_k(a_k^t, b_k^t | h_k^t) \leq H_k + \sum_{m=0}^{M-1} \sqrt{\frac{2(V_m \eta^t K + C_1)}{R}}, \quad \forall k \tag{18}$$

where $C_1 \triangleq K(E^{max} - H^{min}/T)^2/2$.

(b) The federated learning performance satisfies:

$$\sum_{t=0}^{T-1} U(\boldsymbol{a}^t) \geq \sum_{m=0}^{M-1} U_m^* - C_2 \sum_{m=0}^{M-1} \frac{1}{V_m} \tag{19}$$

where $C_2 \triangleq C_1 R + \frac{R(R-1)K}{2}(E^{max})^2$ and U_m^* is the optimal value achieved by the R-round lookahead algorithm in frame m.

Proof: See Appendix C. □

Theorem 2 shows that, given a fixed value of R and M, OCEAN is $O(1/V)$-optimal with respect to the FL performance against the optimal R-lookahead policy, while the

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

energy consumption is guaranteed to be approximately satisfied with a bounded factor $O(\sqrt{V})$. Thus, OCEAN demonstrates an $[O(1/V), O(\sqrt{V})]$ learning-energy tradeoff. Note that when $R = T$, the T-lookahead benchmark has complete future information of the entire T rounds. Even in this case, the $[O(1/V), O(\sqrt{V})]$ tradeoff still holds.

V. Simulation Results

In this section, we simulate a WFLN to evaluate the performance of OCEAN.

Federated Dataset: To simulate FL, we leverage the TensorFlow Federated (TFF) framework and the MNIST dataset for hand-written digit classification. Each client's local dataset is keyed by the original writer of the digits. Since each writer has a unique style, this dataset exhibits the kind of non-i.i.d. behavior expected of federated datasets. We use the first 10 clients in the MNIST dataset to conduct our simulation with each client having about 100 training data samples. Because clients have a similar local dataset size, we take $D_k = 1, \forall k$ after normalization for simplicity. Since the hand-written digit classification is a relatively easy image classification task, we follow TFF's tutorial to construct a simple three-layer neural network with the first layer being input, the second containing 10 neurons and the third performing the softmax operation. This neural network's model size is $L = 3.4 \times 10^5$ bits. FedAvg [1] is used as the learning algorithm.

Wireless Network: To simulate the wireless network, we consider an OFDMA system where the total bandwidth $B = 10$ MHz. Each client's wireless channel gain is modelled as independent free-space fading with average path loss 36dB. The variance of the complex white Gaussian channel noise is set as $N_0 = 10^{-12}$ W. To ensure timely model update, we set the target uploading time in each round to be $\bar{\tau} = 300$ ms. The minimal bandwidth b_{min} is set as 2×10^5 Hz. For ch client k, the energy budget is set as $H_k = 0.15$ J. The network runs for $T = 300$ rounds.

A. Benchmarks

We compare the performance of OCEAN with the following three benchmark algorithms.

- **Select-All**: All 10 clients are selected in every learning round. Bandwidth is allocated to minimize the total energy consumption while satisfying the upload deadline requirement.
- **Static Myopic Optimal (SMO)**: In every learning round, SMO uses only currently available information independently across rounds (which is equivalent to the 1-Round Lookahead algorithm) to solve

$$\max_{\boldsymbol{a}^t, \boldsymbol{b}^t} \sum_k a_k^t \tag{20}$$

$$\text{s.t. } E(a_k^t, b_k^t | h_k^t) \le H_k/T, \quad \forall k$$

$$\text{Constraints (7), (8)} \tag{21}$$

This problem is easy to solve: for each client k, first compute the required bandwidth $b_k^\dagger \ge b_{min}$ so that using H_k/T energy can meet the upload time target $\bar{\tau}$; then rank $b_k^\dagger$ in the ascending order and select clients until the total required bandwidth exceeds B. SMO mimics existing approaches that solves bandwidth allocation and client selection independently across learning rounds.

- **Adaptive Myopic Optimal (AMO)**: SMO has a clear deficiency which can result in energy under-utilization: when a client is not selected in a round, its energy H_k/T is wasted and will not be used in future rounds. To address this issue, we also consider a modified version of SMO, which recycles previously unused energy budget for future rounds. In particular, the energy budget for client k in round t is modified to $(H_k - \sum_{\tau=0}^{t-1} E_k^t)/(T-t)$.
- **Weighted Sum Static Myopic Optimal (WS-SMO)**: This is the formulation adopted in [3], which aims to minimize $\sum_k E(a_k^t, b_k^t | h_k^t) - \lambda \sum_k a_k^t$ in every FL round.

For OCEAN, we let $R = T$ and hence the sequence $V_1, \ldots, V_M$ becomes a single scalar V. Moreover, we implement three variants using different temporal importance sequences η_t: Ascending (OCEAN-a); Descending (OCEAN-d); and Uniform (OCEAN-u).

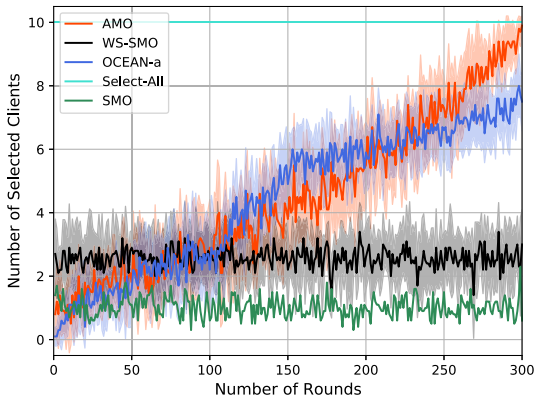

Fig. 2. Temporal client selection patterns of OCEAN and benchmarks.

B. Performance Comparison

Figure 2 shows the number of selected clients in every round for different approaches, which is obtained by averaging over 10 runs. As the name suggests, Select-All selects all 10 clients in every round, resulting in the ideal optimal client selection for FL. SMO selects much fewer clients due to the hard energy budget allocation in every round. Many clients do not get to upload their local model updates due to the bad channel state that they are experiencing. Likewise, WS-SMO also does not select sufficiently many clients because the used $\lambda = 0.2$ is small. Note that choosing an appropriate λ requires a careful retrospective tuning, which is impractical. AMO starts with selecting few clients due to the same reason as SMO. However, as time goes on, energy budget not used in the previous rounds accumulates. This allows the client to transmit at the desired rate using a higher transmission power in later rounds, especially in those towards the very end, thereby countering the effects of bad channel states.

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

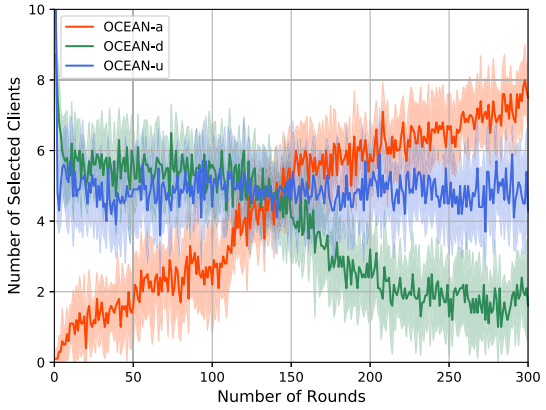

Fig. 3. Temporal client selection patterns of OCEAN variants.

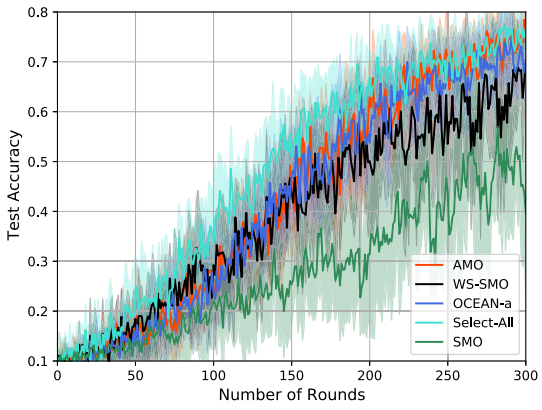

Fig. 5. Test accuracy of OCEAN-a and benchmarks.

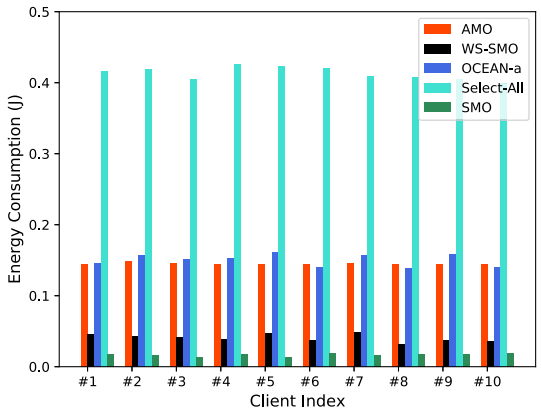

Fig. 4. Per-client energy consumption comparison.

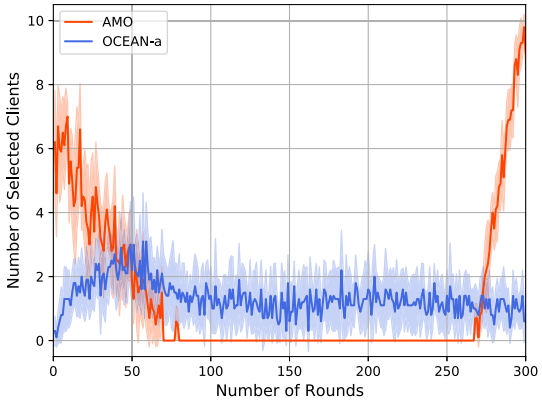

Fig. 6. Client selection of scenario 1.

As a (fortunate) by-product, AMO also achieves an ascending pattern of client selection. Our proposed algorithm, OCEAN-a, is able to select many more clients than SMO because it uses energy as needed without imposing a hard per-round energy constraint. Compared to AMO, it is able to fine-tune the temporal pattern of client selection by using different sequences of temporal weights η. As can be seen in Figure 3, OCEAN-a results in an increasing number of selected clients, OCEAN-d results in a decreasing number of selected clients, while OCEAN-u keeps the number of selected clients almost the same across rounds.

Figure 4 shows the actual energy consumption of individual clients by the end of 300 learning rounds for different approaches in a particular run. Because Select-All completely ignores the energy budgets of the clients, it results in a very large energy consumption, far exceeding the energy budgets. On the other hand, SMO and WS-SMO didn't fully utilize the client's energy budget because in many learning rounds the client is not selected. Both AMO and OCEAN-a incur a total energy consumption close to the given energy budget (i.e. 0.15) for individual clients.

As our ultimate goal is to improve the FL performance, we show the test accuracy for different approaches in Figure 5. Select-All, as expected, results in the best FL performance, with the highest accuracy and the fastest convergence among all approaches. Due to the insufficient selection of clients in the course of learning, SMO's learning performance is considerably inferior to all other approaches. Thanks to the fortunate by-product of AMO, AMO's FL performance is comparable to OCEAN-a in this specific setting, which is close to the ideal case Select-All. However, we will show in the next set of experiments that AMO's "luck" does not extend to other more complex network environments.

C. Adaptability to Varying Network Condition

Although the performance of AMO seems comparable to OCEAN-a in the last experiment, it is achieved in a relatively easy network, where the wireless channel is relatively stable. In this set of experiments, we simulate more challenging network environments where the wireless channel can vary considerably due to, e.g., client mobility. In particular, we simulate two scenarios. In Scenario 1, the average path loss gradually increases from 32 dB to 45 dB, mimicking a scenario where clients move away from the server over time. In Scenario 2, the average path loss gradually decreases from 45 dB to 32 dB, mimicking a scenario where clients move towards the server over time.

Scenario 1: Figure 6 shows the number of selected clients over 300 rounds for OCEAN-a and AMO and Figure 7 shows the their FL accuracy. In the early rounds when the wireless channel is good, AMO selects some clients. However, as the channel gain degrades, AMO is not able to adapt to this change

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

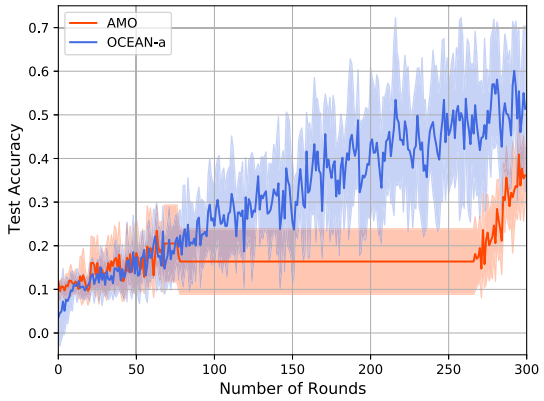

Fig. 7. Test accuracy of scenario 1.

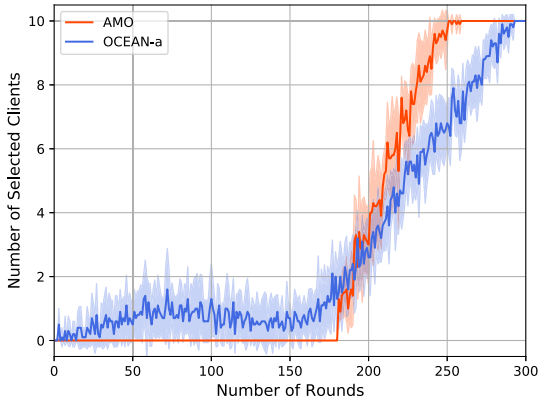

Fig. 8. Client selection of scenario 2.

as the pre-allocated energy budget (even if the unused budget from the previous rounds is incorporated) cannot support even a single client to finish uploading the local model before the deadline $\bar{\tau}$. Only in the rounds towards the very end does the energy budget become sufficient and hence, some clients again are selected to upload their local model updates. Because of the long idle period in the middle when no clients are selected, the learning performance of AMO is significantly worse than OCEAN.

Scenario 2: Figure 8 shows the number of selected clients over 300 rounds for OCEAN-a and AMO and Figure 9 shows the their federated learning accuracy. In this scenario, the channel state in the early rounds is bad and hence, hardly any client can be selected to upload its local model update due to insufficient energy budget in AMO. As the channel state improves, AMO starts to select some clients but it becomes too late to do so.

In both scenarios, OCEAN is able to adapt its client selection decision because of its soft per-round energy budget allocation, yet the total consumed energy is still made close to the total energy budget. The per-client total energy consumption of OCEAN-a is shown in Figure 10 for the two considered scenarios.

D. Features of OCEAN

1) Client Selection and Bandwidth Allocation Outcomes: To have a deeper understanding of how OCEAN works, we illustrate, in one specific round, how clients are selected and bandwidth is allocated depending on the clients' channel condition and energy deficit queue in that round. In Figure 11, the top subplot shows the current channel condition and energy deficit queue for each client. The middle subplot shows the computed selection priority ρ, with shaded bars indicating the selected clients. The bottom subplot shows the bandwidth allocation among the selected clients. As can be seen, a better channel condition and a larger deficit queue result in a higher priority (i.e., a smaller value of ρ). However, among the selected clients, more bandwidth is allocated to clients of a lower priority (i.e., a larger value of ρ).

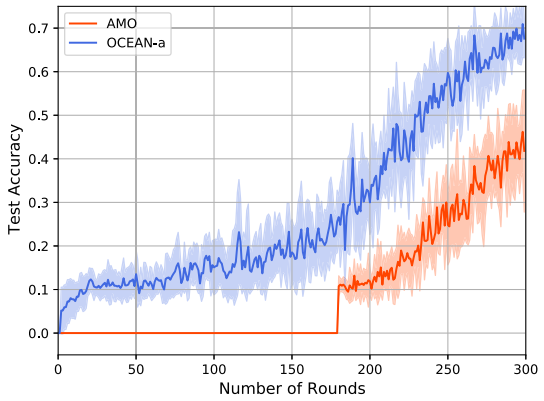

Fig. 9. Test accuracy of scenario 2.

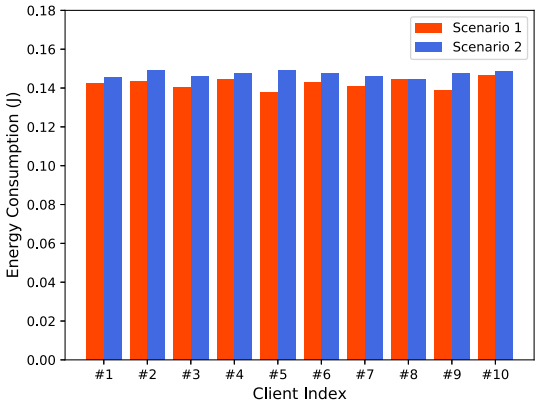

Fig. 10. Energy consumption of OCEAN-a for the two scenarios.

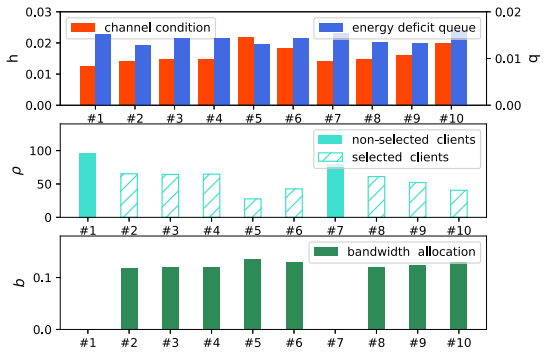

Fig. 11. Client selection and bandwidth allocation outcomes.

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

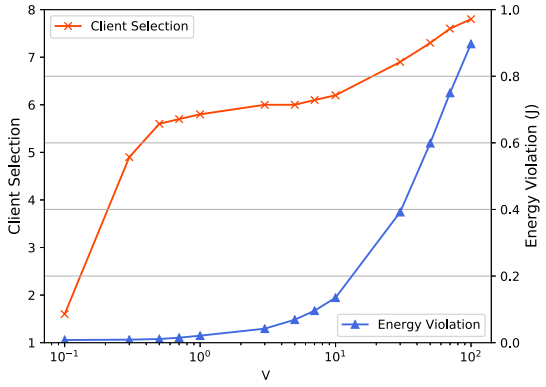

Fig. 12. Tradeoff between learning and energy.

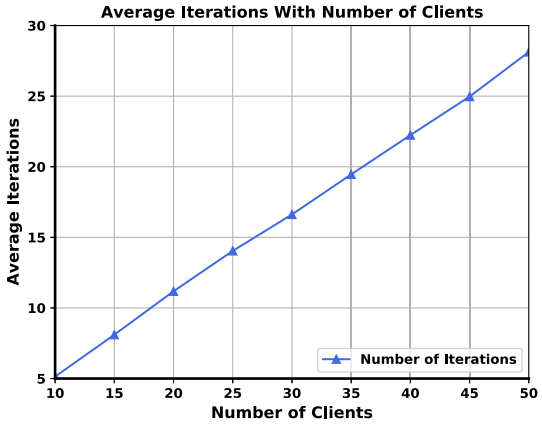

Fig. 13. Actual number of times needed to solve P4 v.s. the number of clients K.

TABLE I
COMPUTATIONAL COMPLEXITY FOR SOLVING P4

Tolerance Gap	# Iterations	# Function Evaluations	# Gradient Evaluations	Time(s)
1e-5	1	2	1	0.002
1e-8	2	2	2	0.015
1e-10	2	2	2	0.022
1e-12	2	13	2	0.13

2) Learning - Energy Tradeoff: Next, we show the impact of the algorithm parameter V on achieving different learning v.s. energy tradeoff of OCEAN. Figure 12 shows the number of selected clients, the learning accuracy and the per-client energy consumption violation as a function of V. As can be seen, a larger V emphasizes more on the learning performance, resulting in more selected clients and higher accuracy. On the other hand, a smaller V emphasizes more on the energy consumption, resulting in a smaller violation (if any) on the total energy budget.

3) Complexity of SEA: Finally, we show numerical results on the complexity of SEA. Figure 13 shows the actual number of times needed to solve **P4** for different K with different number of total clients. In our implementation, we used the SLSQP algorithm [42] to solve each **P4**. Table I shows the number of iterations, the number of function evaluations, the number of gradient evaluations and the wall clock time to solve a single **P4** for different tolerated gap values. (Wall clock time is obtained on a computer with Intel Core i5-9400 2.9GHz CPU with 16GB memory).

VI. CONCLUSION

Resource allocation in wireless networks is an old topic, but it also faces constantly changing new challenges as new applications emerge. With FL being the trending new wireless network application, the old mindset of resource allocation for traditional applications such as file downloading or video streaming must be changed. This paper identifies a key property of FL, namely the temporal dependency and varying significance of learning rounds, that may significantly reshape how wireless resources should be allocated for optimized network and learning performance, yet is largely overlooked in the literature. While our formulation and algorithm have shown superior performance in real-world FL experiments, there are several future research directions that may extend the impact of this work. For example, we showed that an ascending client selection pattern is generally desired, but it is still not clear what the optimal pattern is. A theoretical understanding of why "later-is-better" is an important future research direction.

APPENDIX A
PROOF OF THEOREM 1

The key is to prove that there exists at least one Δ-optimal solution that has the following thresholding structure: there is one $k^\dagger$ so that $a_k^\dagger = 1, \forall k \leq k^\dagger$ and $a_k^\dagger = 0, \forall k > k^\dagger$. Next, we prove this by contradiction. Suppose all $\Delta/2$-optimal solutions do not have a thresholding structure. Otherwise, the above is already true because $\Delta/2 < \Delta$. Consider any particular $\Delta/2$-optimal solution $(\boldsymbol{a}^\dagger, \boldsymbol{b}^\dagger)$, then there must exist some $L \leq K$ and $k_L^- < \ldots < k_1^- < k_1^+ < \ldots, k_L^+$ so that $a_{k_l^-}^\dagger = 0$ and $a_{k_l^+}^\dagger = 1$, and by swapping the decision for k_l^- and k_l^+ for all l, the solution becomes a thresholding solution. Let us study this thresholding solution $(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}})$ after the swap. Specifically,

$$\tilde{a}_{k_l^-} = a_{k_l^+}^\dagger = 1, \ \tilde{b}_{k_l^-} = b_{k_l^+}^\dagger \tag{22}$$

$$\tilde{a}_{k_l^+} = a_{k_l^-}^\dagger = 0, \ \tilde{b}_{k_l^+} = b_{k_l^-}^\dagger \tag{23}$$

Since the decisions for clients other than $k_1^-, \ldots, k_L^-$ and $k_1^+, \ldots, k_L^+$ remain the same, the difference in the objective function value satisfies

$$\begin{aligned}
&W(\boldsymbol{a}^*, \boldsymbol{b}^*) - W(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}) \\
&= W(\boldsymbol{a}^*, \boldsymbol{b}^*) - W(\boldsymbol{a}^\dagger, \boldsymbol{b}^\dagger) + W(\boldsymbol{a}^\dagger, \boldsymbol{b}^\dagger) - W(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}) \\
&\leq \Delta/2 + W(\boldsymbol{a}^\dagger, \boldsymbol{b}^\dagger) - W(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}) \\
&= \Delta/2 + \sum_{l=1}^{L} [V\eta(D_{k_l^-} - D_{k_l^+}) \\
&\quad + (\rho_{k_l^-} - \rho_{k_l^+}) N_0 \tilde{\tau} B f(b_{k_l^+}^\dagger) \\
&\quad + (q_{k_l^-} E_{k_l^-}^{\text{tr}} - q_{k_l^+} E_{k_l^+}^{\text{tr}})] \\
&\leq \Delta/2 + K \cdot \max_{k_1, k_2} (V\eta |D_{k_1} - D_{k_2}| \\
&\quad + |q_{k_1} E_{k_1}^{\text{tr}} - q_{k_2} E_{k_2}^{\text{tr}}|) = \Delta
\end{aligned} \tag{24}$$

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

where the last inequality is because $\rho_{k_l^-} - \rho_{k_l^+} < 0$. Therefore, we proved that at least one Δ-optimal solution has the thresholding structure. Hence, by checking sequentially the thresholding solutions, SEA must return a Δ-optimal solution.

Next, we prove that the termination condition is correct. Let i^* be the first client with $V\eta D_{i^*} - \rho_{i^*} N_0 \tilde{\tau} B f(b^*_{i^*}[i^*]) - q_{i^*} E^{\text{tr}}_{i^*} < 0$ where we use $b^*_k[i]$ to denote the optimal bandwidth allocation for the selection set $S = \{1, \ldots, i\}$. Clearly, when only clients $\{1, \ldots, i^* - 1\}$ are selected, we obtain a higher total utility because each client has more bandwidth. Adding more clients into the selection set only reduces the total utility. This proves the correctness of the termination condition.

Finally, we remain to prove that **P4** is a convex optimization problem. This is easy to check because the function $f(x) = x(2^{\frac{\beta}{x}} - 1)$ where $\beta > 0$ is decreasing and convex in $x \in (0, \infty)$.

Appendix B
Proof of Proposition 1

It suffices to consider the following optimization problem with two clients

$$\min_{b_1, b_2} \ \rho_1 f(b_1) + \rho_2 f(b_2) \tag{25}$$

$$\text{s.t. } b_1 + b_2 = \delta, \quad b_1, b_2 \geq b_{min} \tag{26}$$

where δ is any constant in $(2\ b_{\min}, 1]$. Let $\rho_1 < \rho_2$. Suppose the optimal bandwidth allocation satisfies $b_1^* > b_2^*$, then by Lemma 1, we know $f(b_1^*) < f(b_2^*)$. Let us construct a different bandwidth allocation solution $\tilde{\boldsymbol{b}}$ where $\tilde{b}_1 = b_2^*$ and $\tilde{b}_2 = b_1^*$. In other words, the bandwidth allocation decisions are swapped. This solution also satisfies all constraints. We compare the respective objective values and have

$$\begin{aligned} &\rho_1 f(b_1^*) + \rho_2 f(b_2^*) - (\rho_1 f(\tilde{b}_1) + \rho_2 f(\tilde{b}_2)) \\ &\boldsymbol{e}= (\rho_1 - \rho_2)(f(b_1^*) - f(b_2^*)) > 0 \end{aligned} \tag{27}$$

This contradicts the optimality of $\boldsymbol{b}^*$. Therefore, we must have $b_1^* \leq b_2^*$.

To prove $\rho_1\ f(b_1^*) \leq \rho_2\ f(b_2^*)$, let $b_2 = \delta - b_1$ and ignore the constraint that $b_1, b_2 \geq b_{min}$ for now. The first-order condition requires

$$\rho_1 df(b_1)/db_1 + \rho_2 df(b_2)/db_2 \cdot db_2/db_1 = 0 \tag{28}$$

This leads to

$$\rho_1 f'(b_1^*) = \rho_2 f'(b_2^*) \tag{29}$$

Because $f'(x) < 0$, we can instead prove $f(b_1^*)/f'(b_1^*) \geq f(b_2^*)/f'(b_2^*)$. Let us define $g_1(x) \triangleq f(x)/f'(x)$. Since we have proven $b_1^* \leq b_2^*$ in the above, we only need to prove that $g_1(x)$ is a non-increasing function in $x > 0$. To this end, consider the first order derivative of $g_1(x)$,

$$g_1'(x) = \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} \tag{30}$$

Let $g_2(x) \triangleq (f'(x))^2 - f(x)f''(x)$. We have to prove $g_2(x) \leq 0$ for $x > 0$.

$$\begin{aligned} &g_2(x) \\ &= \left(2^{\frac{\beta}{x}} \left(1 - \ln 2 \cdot \frac{\beta}{x}\right) - 1\right)^2 - x(2^{\frac{\beta}{x}} - 1) \cdot (\ln 2)^2 2^{\frac{\beta}{x}} \frac{\beta^2}{x^3} \\ &= (2^{\frac{\beta}{x}} - 1)^2 - 2(2^{\frac{\beta}{x}} - 1)2^{\frac{\beta}{x}} \ln 2 \cdot \frac{\beta}{x} + (\ln 2)^2 2^{\frac{\beta}{x}} \frac{\beta^2}{x^2} \end{aligned} \tag{31}$$

To simplify notations, we use a change of variable by letting $y = \beta/x$. Then proving $g_2(y) \leq 0$ for $y > 0$ is equivalent to proving $g_2(x) \leq 0$ for $x > 0$. We rewrite $g_2(y)$ below:

$$g_2(y) \triangleq (2^y - 1)^2 - 2(2^y - 1)2^y \ln 2 \cdot y + (\ln 2)^2 2^y y^2 \tag{32}$$

Clearly, $g_2(0) = 0$. In order to prove $g_2(y) \leq 0$, we prove $g_2(y)$ is decreasing in $y \geq 0$.

$$g_2'(y) = -(\ln 2)^2 y 2^y \left(4(2^y - 1) - \ln 2 \cdot y\right) \tag{33}$$

It is easy to verify that $g_3(y) \triangleq 4(2^y - 1) - \ln 2 \cdot y$ is increasing in $y > 0$ and $g_3(0) = 0$, which means $g_3(y) \geq 0$ for $y \geq 0$. Hence, $g_2'(y) < 0$ for $y > 0$. This concludes the proof for $\rho_1\ f(b_1^*) \leq \rho_2\ f(b_2^*)$ by ignoring the constraint $b_1, b_2 \geq b_{min}$.

When the constraint $b_1, b_2 \geq b_{min}$ is considered, there are two cases. In the first case, $b_{min} \leq b_1^* \leq b_2^*$. In this case, the constraint is automatically satisfied and hence, our above conclusion holds. In the second case, $b_1^* \leq b_{min} \leq b_2^*$. In this case, the optimal allocation is modified to $\tilde{b}_1^* = b_{min} \geq b_1^*$ and $\tilde{b}_2^* = 1 - b_{min} \leq b_2^*$. Since $f(x)$ is a decreasing function, $\rho_1\ f(\tilde{b}_1^*) \leq \rho_1\ f(b_1^*) \leq \rho_2\ f(b_2^*) \leq \rho_2\ f(\tilde{b}_2^*)$. This completes the proof.

Appendix C
Proof of Theorem 2

We define the quadratic Lyapunov function $L(\boldsymbol{q}(t)) \triangleq \frac{1}{2}\sum_{k=1}^K q_k^2(t)$. Let $\Delta_1(t)$ be the 1-round Lyapunov drift yielded by some control decisions over one round: $\Delta_1(t) \triangleq L(\boldsymbol{q}(t+1)) - L(\boldsymbol{q}(t))$. Similarly, let $\Delta_R(t)$ be the R-round Lyapunov drift: $\Delta_R(t) \triangleq L(\boldsymbol{q}(t+R)) - L(\boldsymbol{q}(t))$. Based on the queue dynamics, we have

$$\frac{1}{2}\sum_k^K q_k^2(t+1) \leq \frac{1}{2}\sum_{k=1}^K [E_k(a_k^t, b_k^t | h_k^t) - H_k/T + q_k(t)]^2$$

Then, it can be easily show that

$$\Delta_1(t) \leq C_1 + \sum_{k=1}^K q_k(t) \cdot [E_k(a_k^t, b_k^t | h_k^t) - H_k/T] \tag{34}$$

where C_1 is a constant satisfying $C_1 \geq \frac{1}{2}\sum_{k=1}^K (E^{\max} - H^{\min}/T)^2, \forall t$, which is finite due to the boundedness of the channel condition h_k^t and the minimum bandwidth allocation requirement b_{min}. Next, it is straightforward that $\forall m$ and $\forall t = mR, \ldots, (m+1)R - 1$

$$\begin{aligned} &V_m \cdot U(\boldsymbol{a}^t) - \Delta_1(t) \\ &\boldsymbol{e}\geq V_m \cdot U(\boldsymbol{a}^t) - \sum_{k=1}^K q_k(t) \cdot [E_k(a_k^t, b_k^t | h_k^t) - H_k] - C_1 \end{aligned} \tag{35}$$

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

As we can see, by solving **P3**, SEA actually maximizes a lower bound of $V_m \cdot U(\boldsymbol{a}^t) - \Delta_1(t)$. Let $\hat{\boldsymbol{a}}^0, \hat{\boldsymbol{b}}^0, \ldots, \hat{\boldsymbol{a}}^{T-1}, \hat{\boldsymbol{b}}^{T-1}$ be the sequence of decisions derived by the online algorithm.

(a) Consider a specific sequence of decisions where $\tilde{a}_k^t = 0, \forall t, k$. Clearly, in this case, $U(\tilde{\boldsymbol{a}}^t) = 0$ and $E_k(\tilde{a}_k^t, \tilde{b}_k^t | h_k^t) - H_k/K = -H_k/K$. Because $\hat{a}^t, \hat{b}$ maximizes the right-hand side of (35), we have

$$V_m \cdot U(\hat{\boldsymbol{a}}^t) - \Delta_1(t) \geq V_m \cdot 0 + \sum_{k=1}^{K} \tilde{q}_k(t) H_k - C_1 \geq -C_1$$

Therefore, $\Delta_1(t) \leq V_m \cdot U(\hat{\boldsymbol{a}}^t) + C_1 \leq V_m \eta^t K + C_1$. As enforced by the online algorithm,

$$\begin{aligned} \Delta_R(mR) &= \tfrac{1}{2} \textstyle\sum_{k=1}^{K} (q_k^2(mR+R) - q_k^2(mR)) \\ &= \tfrac{1}{2} \textstyle\sum_{k=1}^{K} q_k^2(mR+R) \end{aligned} \tag{36}$$

is the R-round drift calculated after the m-th rest but before the $(m+1)$-th reset of the energy deficit queue (so $q_k(mR) = 0$). Thus, before the $(m+1)$-th reset of the energy deficit queue:

$$\begin{aligned} &\sum_{k=1}^{K} q_k^2(mR+R) = 2\Delta_R(mR) \\ &\boldsymbol{e}= 2 \sum_{t=mR}^{mR+R-1} \Delta_1(t) \leq 2R(V_m \eta^t K + C_1) \end{aligned} \tag{37}$$

Therefore, $\forall k$,

$$q_k(mR+R) \leq \sqrt{2R(V_m \eta^t K + C_1)} \tag{38}$$

On the other hand, according to (11), we have

$$q_k(t+1) - q_k(t) \geq E_k(a_k^t, b_k^t | h_k^t) - H_k/T \tag{39}$$

Summing both sides over the rounds in the m-th frame, namely $t = mR, \ldots, (m+1)R - 1$, and dividing by R, we have

$$\begin{aligned} &\frac{1}{R} \sum_{t=mR}^{(m+1)R-1} (E_k(a_k^t, b_k^t | h_k^t) - H_k/T) \\ &\boldsymbol{e}\leq \frac{q_k((m+1)R) - q_k(mR)}{R} = \frac{q_k((m+1)R)}{R} \end{aligned} \tag{40}$$

Plugging (38) into (40), we have

$$\frac{1}{R} \sum_{t=mR}^{(m+1)R-1} (E_k(\hat{a}_k^t, \hat{b}_k^t | h_k^t) - H_k/T) \leq \sqrt{\frac{2(V_m \eta^t K + C_1)}{R}}$$

Considering all M frames, we obtain

$$\sum_{t=0}^{T} E_k(\hat{a}_k^t, \hat{b}_k^t | h_k^t) \leq H_k + \sum_{m=0}^{M-1} \sqrt{\frac{2(V_m \eta^t K + C_1)}{R}}, \forall k$$

(b) Consider R-round weighted learning utility minus drift:

$$\begin{aligned} &V_m \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^t) - \Delta_R(mR) \\ &\geq V_m \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^t) - C_1 R \\ &\quad - \sum_{t=mR}^{mR+R-1} \sum_{k=1}^{K} q_k(t) \cdot [E_k(a_k^t, b_k^t | h_k^t) - H_k/T] \end{aligned} \tag{41}$$

Because $\hat{\boldsymbol{a}}^0, \hat{\boldsymbol{b}}^0, \ldots, \hat{\boldsymbol{a}}^{T-1}, \hat{\boldsymbol{b}}^{T-1}$ explicitly maximizes the right-hand side of the above equation, the following must also hold

$$V_m \sum_{t=mR}^{mR+R-1} U(\hat{\boldsymbol{a}}^t) - \Delta_R(mR) \tag{42}$$

$$\geq V_m \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^{*,t}) - C_1 R$$

$$- \sum_{t=mR}^{mR+R-1} \sum_{k=1}^{K} q_k(t) \cdot [E_k(a_k^{*,t}, b_k^{*,t} | h_k^t) - H_k/T] \tag{43}$$

$$\geq V_m \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^{*,t}) - C_1 R$$

$$- \sum_{t=mR}^{mR+R-1} \sum_{k=1}^{K} (t - mR) E^{\max} \cdot [E_k(a_k^{*,t}, b_k^{*,t} | h_k^t) - H_k/T]$$

$$- \sum_{k=1}^{K} q_k(mR) \sum_{t=mR}^{mR+R-1} [E_k(a_k^{*,t}, b_k^{*,t} | h_k^t) - H_k/T] \tag{44}$$

$$\geq V_m \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^{*,t})$$

$$- \sum_{k=1}^{K} q_k(mR) \sum_{t=mR}^{mR+R-1} [E_k(a_k^{*,t}, b_k^{*,t} | h_k^t) - H_k/T]$$

$$- \left(C_1 R + \frac{R(R-1)K}{2} (E^{\max})^2 \right) \tag{45}$$

$$\geq V_m \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^{*,t}) - \left(C_1 R + \frac{R(R-1)K}{2} (E^{\max})^2 \right) \tag{46}$$

where in $\hat{\boldsymbol{a}}^{*,0}, \hat{\boldsymbol{b}}^{*,0}, \ldots, \hat{\boldsymbol{a}}^{*,T-1}, \hat{\boldsymbol{b}}^{*,T-1}$ is the optimal decision that solves the R-round lookahead problems. Notice that $q_k(t)$ in the above equation is still derived by SEA. The first inequality holds because SEA maximizes the lower bound. The last inequality holds because $q_k(mR)$ is reset to zero as enforced by SEA.

Noting $\Delta_R(mR) \geq 0$ and dividing both sides by V_m yields:

$$\begin{aligned} &\sum_{t=mR}^{mR+R-1} U(\hat{\boldsymbol{a}}^t) \\ &\geq \sum_{t=mR}^{mR+R-1} U(\boldsymbol{a}^{*,t}) - \frac{1}{V_m} \left(C_1 R + \frac{R(R-1)K}{2} (E^{\max})^2 \right) \end{aligned} \tag{47}$$

By summing over $m = 0, \ldots, M-1$, we have

$$\sum_{t=0}^{T-1} U(\hat{\boldsymbol{a}}^t) \geq \sum_{m=0}^{M-1} U_m^* - C_2 \sum_{m=0}^{M-1} \frac{1}{V_m} \tag{48}$$

where $C_2 \triangleq C_1 R + \frac{R(R-1)K}{2} (E^{\max})^2$.

REFERENCES

[1] J. Kone$\boldsymbol{v}$cný, 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: http://arxiv.org/abs/1610.05492

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.

[2] 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.

[3] 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.

[4] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*. [Online]. Available: http://arxiv.org/abs/1610.02527

[5] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning," in *Proc. ICML*, 2020, pp. 1–12.

[6] 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: http://arxiv.org/abs/1812.06127

[7] 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

[8] 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: http://arxiv.org/abs/1806.00582

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

[10] L. Corinzia and J. M. Buhmann, "Variational federated multi-task learning," 2019, *arXiv:1906.06268*. [Online]. Available: http://arxiv.org/abs/1906.06268

[11] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," 2018, *arXiv:1812.00984*. [Online]. Available: http://arxiv.org/abs/1812.00984

[12] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*. [Online]. Available: http://arxiv.org/abs/1712.07557

[13] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, 2019, pp. 1–11.

[14] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.

[15] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning," in *Proc. IEEE Symp. Secur. Privacy*, May 2019, pp. 739–753.

[16] 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.

[17] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. ICLR*, 2018, pp. 1–14.

[18] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. EMNLP*, 2017, pp. 440–445.

[19] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," 2019, *arXiv:1905.06641*. [Online]. Available: http://arxiv.org/abs/1905.06641

[20] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[21] 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.

[22] G. Zhu, Y. Du, D. Gunduz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," 2020, *arXiv:2001.05713*. [Online]. Available: http://arxiv.org/abs/2001.05713

[23] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[24] 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 INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1387–1395.

[25] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," 2020, *arXiv:2003.00199*. [Online]. Available: http://arxiv.org/abs/2003.00199

[26] Y. Zhan, P. Li, and S. Guo, "Experience-driven computational resource allocation of federated learning by deep reinforcement learning," in *Proc. IPDPS*, 2020, pp. 234–243.

[27] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. ICLR*, 2019, pp. 1–17.

[28] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[29] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, early access, Oct. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9210812

[30] 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.

[31] 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.

[32] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," 2019, *arXiv:1911.02417*. [Online]. Available: http://arxiv.org/abs/1911.02417

[33] 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: http://arxiv.org/abs/2001.07845

[34] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. ICLR*, 2020, pp. 1–26.

[35] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local GD on heterogeneous data," 2019, *arXiv:1909.04715*. [Online]. Available: http://arxiv.org/abs/1909.04715

[36] S. De, A. Yadav, D. Jacobs, and T. Goldstein, "Automated inference with adaptive batches," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1504–1513.

[37] M. Reneer. (2020). *TensorFlow Federated*. [Online]. Available: https://www.tensorflow.org/federated

[38] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[39] M. Grant and S. Boyd. (Sep. 2013). *CVX: Matlab Software for Disciplined Convex Programming, Version 2.0 Beta*. [Online]. Available: http://cvxr.com/cvx

[40] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, pp. 261–272, Feb. 2020.

[41] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synth. Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, Jan. 2010.

[42] D. Kraft, *A Software Package for Sequential Quadratic Programming* (Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht). Brunswick, Germany: Wiss. Berichtswesen d. DFVLR, 1988. [Online]. Available: https://books.google.com/books?id=4rKaGwAACAAJ

Jie Xu (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 Assistant Professor with the Electrical and Computer Engineering Department, University of Miami. His primary research interests include machine learning, edge computing, wireless communications, and network security.

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

Authorized licensed use limited to: UNIV OF MIAMI LIB. Downloaded on August 22,2021 at 01:17:37 UTC from IEEE Xplore. Restrictions apply.