entropy

MDPI

Article

Joint Client Selection and CPU Frequency Control in Wireless Federated Learning Networks with Power Constraints

Zhaohui Zhou, Shijie Shi, Fasong Wang, Yanbin Zhang and Yitong Li *

School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China
* Correspondence: ieytli@zzu.edu.cn

Abstract: Federated learning (FL) represents a distributed machine learning approach that eliminates the necessity of transmitting privacy-sensitive local training samples. However, within wireless FL networks, resource heterogeneity introduces straggler clients, thereby decelerating the learning process. Additionally, the learning process is further slowed due to the non-independent and identically distributed (non-IID) nature of local training samples. Coupled with resource constraints during the learning process, there arises an imperative need for optimizing client selection and resource allocation strategies to mitigate these challenges. While numerous studies have made strides in this regard, few have considered the joint optimization of client selection and computational power (i.e., CPU frequency) for both clients and the edge server during each global iteration. In this paper, we initially define a cost function encompassing learning latency and non-IID characteristics. Subsequently, we pose a joint client selection and CPU frequency control problem that minimizes the time-averaged cost function subject to long-term power constraints. By utilizing Lyapunov optimization theory, the long-term optimization problem is transformed into a sequence of short-term problems. Finally, an algorithm is proposed to determine the optimal client selection decision and corresponding optimal CPU frequency for both the selected clients and the server. Theoretical analysis provides performance guarantees and our simulation results substantiate that our proposed algorithm outperforms comparative algorithms in terms of test accuracy while maintaining low power consumption.

Keywords: federated learning; client selection; CPU frequency control; wireless networks

check for updates

Citation: Zhou, Z.; Shi, S.; Wang, F.; Zhang, Y.; Li, Y. Joint Client Selection and CPU Frequency Control in Wireless Federated Learning Networks with Power Constraints. *Entropy* **2023**, *25*, 1183. https://doi.org/10.3390/e25081183

Academic Editor: Jerry D. Gibson

Received: 30 June 2023
Revised: 30 July 2023
Accepted: 5 August 2023
Published: 9 August 2023

1. Introduction

As the rapid expansion of Internet of Things (IoT) communication unfolds, an immense volume of data generated by massive machine-type devices circulates via wireless access technology. This advancement instigates the pervasive utilization of machine-learning-based applications in various aspects of people's everyday life, including smart transportation and smart healthcare. In the traditional centralized learning paradigm, the raw data of each device are initially uploaded to the edge server via a wireless channel, facilitating aggregation for subsequent processing and analysis. This methodology, however, potentially leads to privacy data leakage. Consequently, the exploration of machine learning mechanisms that protect user data privacy is of paramount significance. FL was proposed as a solution to address the limitations of traditional centralized machine learning methods in ensuring user data privacy [1]. This approach permits each device to partake in the collaborative training of a shared model, negating the need to share the device's proprietary data. Since its inception, federated learning has garnered considerable interest from both academia and industry, finding broad applications in fields such as mobile cloud computing [2], the industrial Internet of Things [3], and device-to-device communication [4].

Despite the significant advantages offered by federated learning, its application in wireless networks presents certain challenges that warrant attention. Firstly, the scarcity of wireless resources, such as channel bandwidth, limit the number of clients capable of participating in each learning iteration. Additionally, client selection for each iteration needs to be guided by the real-time state of the channel. Secondly, the computational power and power budget of devices and the edge server are finite, necessitating the optimization of computational power and power allocation throughout the multi-iteration learning process in FL, which is essential for both extending battery life and advancing green communication. A third challenge arises from the latency incurred during the federated learning process, which constrains its use in scenarios that are latency-sensitive. The latency is determined by the time required for straggler clients to upload the local model during each learning iteration, the time spent aggregating the model on the edge server, which is influenced by the transmission power and CPU frequency in each learning iteration. Furthermore, the heterogeneity of client behavior and the variable dynamics of wireless environments may lead to the acquisition of non-independent and identically distributed (non-IID) training data [5]. Some clients may hold data that significantly deviate from independent identically distributed (IID) training data, making it challenging for the model to generalize effectively across all clients.

1.1. Related Work

Ever since the proposition of federated learning, an extensive research effort has been devoted to improving its performance in wireless networks. This advancement primarily hinges on designing appropriate client selection schemes [6,7] or optimizing resource allocation [8–10]. For instance, a client scheduling strategy based on channel and learning qualities was proposed in [7]. Ref. [9] investigated both the CPU frequency and transmission power control strategy of all IoT clients to minimize the energy consumption under latency requirement. The studies presented in [11–13] focused on the optimization of the client selection process and resource allocation simultaneously, with the objective of improving federated learning performance. Specifically, Ref. [11] delved into a joint client selection and resource allocation problem, seeking to optimize the trade-off between the number of selected clients and the total energy consumption, while [12] focused on minimizing training loss while adhering to the constraints of delay and energy consumption. Additionally, Ref. [13] explored the process of client selection and resource allocation under the condition of non-IID data distributions.

The aforementioned studies were formulated in the context of individual global iterations, overlooking the interdependence between different iterations. This neglects the cumulative learning effect over multiple iterations, potentially yielding less effective learning models and limiting overall system performance. Hence, the need for long-term optimization that accounts for the interconnectedness of global iterations becomes evident. Numerous research efforts [14–18] have targeted long-term optimization in federated learning, focusing on various aspects of the problem. For instance, Ref. [14] sought to optimize the client selection process in each learning round with the aim of minimizing training latency under fairness constraints. The study presented in [15] introduced a dynamic scheduling mechanism aimed at optimizing the federated learning process, striking a balance between the enhancement of learning performance and the reduction of training latency. Ref. [16] focused on the optimization of radio transmission parameters and computation resources, attempting to minimize power consumption while upholding learning performance and latency constraints. Refs. [17,18] focused on client selection and bandwidth allocation under energy constraints in wireless FL networks. Specifically, the study in [17] aimed to maximize the weighted sum of selected clients, whereas [18] focused on minimizing the cost of time and accuracy. While the above works explored long-term optimization in federated learning, the optimization of the latency and the impact of its non-IID nature under the long-term power constraints of both clients and the server for FL have not been considered.

1.2. Contribution

In this paper, we consider a client selection and CPU frequency control problem in wireless FL networks. Different from the extant literature, our approach concurrently optimizes the selection of clients and CPU frequency for both clients and the edge server. The objective of the proposed problem is to minimize a predefined cost function, which incorporates latency and model robustness, under long-term power constraints. The main contributions of our work are as follows:

(1) We develop a comprehensive framework for the long-term client selection and CPU frequency control problem, taking into account the interdependence of different global iterations and long-term power consumption constraints for both clients and the server. The aim is to expedite the learning process by incorporating client and server latency, as well as the effect of the non-IID distribution of local training samples.

(2) Leveraging Lyapunov optimization theory, we transform the long-term problem into a set of per-iteration problems. We introduce an algorithm to tackle the per-iteration problem, accompanied by a theoretical performance guarantee.

(3) We conduct extensive experiments, inclusive of several comparative experiments. Simulation results demonstrate that our proposed algorithm can yield superior test accuracy while maintaining low power consumption.

The remainder of this paper is structured as follows. Our proposed framework's system model, along with the optimization problem formulation, is elucidated in Section 2. The solution via the Lyapunov optimization theory, is laid out in Section 3. Section 4 comprises the simulation results, showcasing the superiority of our proposed scheme. Finally, Section 5 concludes and discusses the paper.

2. System Model and Problem Formulation

The proposed federated learning framework is shown in Figure 1, consisting of a set of clients $\mathcal{K} = \{1, \dots, K\}$ and a server, with K indicating the total number of clients. Each client $k \in \mathcal{K}$ possesses a local dataset $\mathcal{D}_k = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{d_k}$, wherein $\boldsymbol{x}_i$ and y_i denote the i-th sample and its associated ground-truth label of client k, respectively, and d_k stands for the dataset size originating from q_k label classes.

2.1. Learning Model

Assuming T global iterations, we adopt $a_k^t = 1$ to represent the selection of client k in global iteration $t = 0, \dots, T-1$, with $a_k^t = 0$ otherwise. The client selection decisions are denoted by $\boldsymbol{a}^t = (a_1^t, \dots, a_K^t)$. The server aims to construct a global model by minimizing the following global loss function:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \frac{\sum_{k=1}^{K} [d_k f_k(\boldsymbol{w})]}{\sum_{k=1}^{K} d_k}, \tag{1}$$

where $f_k(\boldsymbol{w})$ is the local loss function at client k. For instance, the loss function for linear regression is given by:

$$f_k(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{x}_i^T \boldsymbol{w} - y_i)^2. \tag{2}$$

The goal of the training process is to find the optimal model $\boldsymbol{w}^*$ though iteration. A global iteration t consists of four steps:

(1) Each client shares its side's information. Subsequently, the server selects a group of clients and broadcasts the current global model $\boldsymbol{w}^t$ to them.

(2) The selected clients execute a local iteration to update their local models $\boldsymbol{w}_k^t$ based on their respective datasets.

(3) The selected clients upload their newly updated local models to the server.

(4) The server aggregates all the received local models to establish a new global model, as represented by $\boldsymbol{w}^{t+1} = \sum_{k=1}^{K} (a_k^t \boldsymbol{w}_k^t) / \sum_{k=1}^{K} a_k^t$.

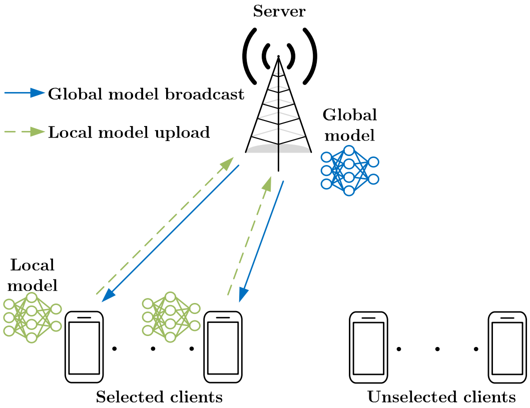


Figure 1. Federated learning framework in wireless networks.

2.2. Power Consumption Model

In each global iteration, the selected clients engage in training and uploading models while the server aggregates these received models. This process contributes to power consumption. We represent the overall CPU frequency control decisions of the clients as $\boldsymbol{f}^t = (f_1^t, \ldots, f_K^t)$, where f_k^t indicates the CPU frequency of client k in the global iteration t. Notably, if $a_k^t = 0$, then f_k^t also equals zero. The power computation of the training model can be expressed as $P_k^{t,tr} = \gamma_1 (f_k^t)^3 a_k^t$, where γ_1 denotes the capacitance coefficient of clients [18]. Let $P_k^{t,up} = p_k^t a_k^t$ denote the power spent for uploading the model; thus, the total power consumption of client k during the global iteration t is given by:

$$P_k^t = P_k^{t,tr} + P_k^{t,up}. \tag{3}$$

On the other hand, the CPU frequency of the server during the global iteration t is represented by f_r^t, and the server's capacitance coefficient is denoted as γ_2. Consequently, the power consumption of the server during the global iteration t can be formulated as follows:

$$P_r^t = \gamma_2 (f_r^t)^3. \tag{4}$$

2.3. Latency Model

Let m represent the number of local iterations in each global iteration, and c_k stand for the number of CPU cycles necessary to process a sample from client k. The local training latency for selected client k in global iteration t can then be calculated as $\tau_k^{t,tr} = m c_k d_k a_k^t / f_k^t$, which will linearly decrease as the allocated local computing power f_k^t increases.

When the local training is finished, the selected clients upload their models to the server via orthogonal frequency-division multiple access (OFDMA). The total available bandwidth is denoted as B, and it is assumed that this bandwidth is equally allocated to the selected clients during the global iteration t. Consequently, the bandwidth allocated to a selected client k in global iteration t can be represented as

$$b^t = B / \sum_{k=1}^{K} a_k^t. \tag{5}$$

The model size is represented as s; therefore, the latency for model uploading is given by

$$\tau_k^{t,up} = \frac{s a_k^t}{b^t \log_2(1 + \frac{h_k^t p_k^t}{N_0 b^t})}, \tag{6}$$

where h_k^t denotes the channel gain between client k and the server during the global iteration t, which is assumed to be available at the transmitter side. N_0 denotes the power spectral density of noise. The total latency of client k can be formulated as:

$$\tau_k^t = \tau_k^{t,tr} + \tau_k^{t,up}. \tag{7}$$

At the server side, let τ_r^t denote the latency of the server in global iteration t, which can be written as:

$$\tau_r^t = \frac{\phi \sum\limits_{k=1}^{K} a_k^t}{f_r^t}, \tag{8}$$

where ϕ is the quantity of processing cycles required to carry out a single summation operation [16].

We assume that the server starts aggregating after receiving all the local models of selected clients. Therefore, the learning latency of global iteration t is bottlenecked by the straggler clients and can be derived as:

$$\tau^t = \max_{k=1,\ldots K}(\tau_k^t) + \tau_r^t. \tag{9}$$

2.4. Cost Model

The non-IID nature of data introduces biases in the training process, which significantly impacts the accuracy of FL. As noted in [13], a larger number of label classes might result in a more robust trained model, and the non-IID nature could decrease when clients possess more label classes. In this paper, we use label classes q_k to quantify the non-IID nature with an aim to minimize both the learning latency and accuracy degradation caused by it. However, reducing the latter could potentially increase the learning latency. Therefore, we propose a cost objective function U^t to balance the two goals during the global iteration t:

$$U^t(\boldsymbol{a}^t, \boldsymbol{f}^t, f_r^t) = \tau^t - \mu \sum_{k=1}^{K} (a_k^t q_k), \tag{10}$$

where μ is a price parameter, which turns the label classes into a cost form [19].

2.5. Problem Formulation

From the aforementioned discussion, we consider an optimization problem that minimizes the time-averaged cost function through joint client selection and CPU frequency control as follows:

$$\text{P1} \min_{\boldsymbol{G}^0,\ldots,\boldsymbol{G}^{T-1}} \frac{1}{T} \sum_{t=0}^{T-1} U^t(\boldsymbol{a}^t, \boldsymbol{f}^t, f_r^t) \tag{11}$$

$$\text{s.t. } f_k^{\min} \le f_k^t \le f_k^{\max}, \forall k, \forall t, \tag{12}$$

$$f_r^{\min} \le f_r^t \le f_r^{\max}, \forall t, \tag{13}$$

$$a_k^t \in \{0, 1\}, \forall k, \forall t, \tag{14}$$

$$\frac{1}{T} \sum_{t=0}^{T-1} P_k^t \le \bar{P}_k, \forall k, \tag{15}$$

$$\frac{1}{T} \sum_{t=0}^{T-1} P_r^t \le \bar{P}_r. \tag{16}$$

where $\boldsymbol{G}^t = (\boldsymbol{a}^t, \boldsymbol{f}^t, f_r^t)$ is the optimization variables in global iteration t, $t = 0, 1, \ldots, T-1$. Constraint (12) and (13) specify the CPU frequency range of each client and the server, respectively. Constraint (14) defines whether each client is selected or not. Constraint (15) guarantees that the average power consumption of each client is limited by $\bar{P}_k$, while constraint (16) guarantees that the average power consumption of the server is limited by $\bar{P}_r$. For clarity, in the following sections of this paper, we succinctly refer to the cost function introduced in Equation (10) as U^t.

3. Problem Solution and Algorithm Design

A direct resolution of problem P1 is not viable due to the time-averaged optimization objective and long-term power constraints. Therefore, in this paper, problem P1 is initially transformed into a per-iteration problem by utilizing Lyapunov optimization theory. Subsequently, this per-iteration problem is decomposed into two distinct subproblems: a CPU frequency control problem, which assumes fixed client selection decisions, and a client selection problem that operates under the optimal CPU frequency setting.

3.1. Problem Transformation via Stochastic Optimization Theory

The resolution of problem P1 necessitates comprehensive information, such as channel gain, pertaining to T global iterations. However, the unavailability of future information in the present moment presents a formidable challenge. To circumvent this issue, P1 is converted into a series of subproblems, the solutions for which do not rely on the knowledge of future iterations. This transformation is achieved through the application of Lyapunov optimization theory [20] and the introduction of virtual queue techniques. For each client, a virtual power deficit queue Z_k^t is established, with an initial condition of $Z_k^0 = 0$, and updated at the end of each global iteration as follows:

$$Z_k^{t+1} = \max\{P_k^t - \bar{P}_k + Z_k^t, 0\}, \tag{17}$$

where Z_k^t encapsulates the disparity between power consumption and the long-term power constraint of client k over T iterations. A similar approach can be used to construct a virtual power deficit queue Y_r^t for the server, as depicted:

$$Y_r^{t+1} = \max\{P_r^t - \bar{P}_r + Y_r^t, 0\}. \tag{18}$$

To maintain the mean rate stability of the queues, we first establish a Lyapunov function in the following form:

$$L(\boldsymbol{\Theta}^t) = \frac{1}{2}[\sum_{k=1}^{K} (Z_k^t)^2 + (Y_r^t)^2], \tag{19}$$

where $\boldsymbol{\Theta^t}$ symbolizes all the virtual deficit queues. Then we formulate Lyapunov drift to measure the expected increase as of $L(\boldsymbol{\Theta}^t)$ as follows:

$$\Delta(\boldsymbol{\Theta}^t) = E[L(\boldsymbol{\Theta}^{t+1}) - L(\boldsymbol{\Theta}^t)|\boldsymbol{\Theta}^t]. \tag{20}$$

With the objective of restricting the growth of virtual deficit queues and minimizing the cost function, the objective function is integrated into the Lyapunov drift. Consequently, the drift-plus-cost function is defined as follows:

$$\Delta(\boldsymbol{\Theta}^t) + VE[U^t|\boldsymbol{\Theta}^t], \tag{21}$$

where V serves as a control parameter that aids in balancing the trade-off between minimizing the objective function and adhering to the power constraints. An observation of (21) indicates that it solely involves the current iteration t, signifying that the original problem P1 can be transitioned into a real-time problem solved on a per-iteration basis. The application of Lyapunov optimization theory provides the following lemma regarding the upper bound of the drift-plus-cost function:

Theorem 1. *Assume $P_k^{\max} \geq P_k^t$ for each client k, and $P_r^{\max} \geq P_r^t$ for the server in global iteration t. The drift-plus-cost function satisfies:*

$$\Delta(\mathbf{\Theta}^t) + VE[U^t|\mathbf{\Theta}^t] \leq C_1 + \sum_{k=1}^{K} Z_k^t E[P_k^t - \bar{P}_k|\mathbf{\Theta}^t] + Y_r^t E[P_r^t - \bar{P}_r|\mathbf{\Theta}^t] + VE[U^t|\mathbf{\Theta}^t], \quad (22)$$

where C_1 is a finite constant, which satisfies $C_1 \geq \frac{1}{2}\sum_{k=1}^{K}(P_k^{\max} - \bar{P}_k)^2 + \frac{1}{2}(P_r^{\max} - \bar{P}_r)^2$.

Proof. The proof is given in Appendix A. □

By minimizing the upper bound in Equation (22), virtual deficit queue stability is achieved concurrently with cost function minimization. Upon excluding all constants (i.e., C, $\bar{P}_k Z_k^t$, $\bar{P}_r Y_r^t$), problem P1 can be transformed into a per-iteration problem P2:

$$\begin{aligned} \text{P2} \ \min_{G^t} \sum_{k=1}^{K} (P_k^t Z_k^t) + P_r^t Y_r^t + VU^t \\ \text{s.t. (12)–(14).} \end{aligned} \quad (23)$$

3.2. Problem Solution

To simplify the complexity, U^t in Equation (21) is substituted with an upper bound $\tilde{U}^t = \sum_{k=1}^{K} \tau_k^t + \tau_r^t - \mu \sum_{k=1}^{K}(a_k^t q_k^t)$, derivable through the application of $\max_{k=1,\ldots K}(\tau_k^t) \leq \sum_{k=1}^{K} \tau_k^t$. Consequently, the resolution of P2 can be reoriented towards the following problem:

$$\begin{aligned} \text{P3} \ \min_{G^t} \sum_{k=1}^{K} (P_k^t Z_k^t) + P_r^t Y_r^t + V\tilde{U}^t \\ \text{s.t. (12)–(14).} \end{aligned} \quad (24)$$

Problem P3 manifests as a mixed-integer problem and poses a significant challenge for direct resolution. However, given any $\boldsymbol{a}^t$, the objective function of P3 transforms into a convex function with respect to the CPU frequency of the selected clients and the server, i.e., f_k^t and f_r^t. Consequently, the optimal CPU frequencies for selected clients and the server can be efficiently procured as

$$(f_k^t)^* = \begin{cases} f_k^{\min}, if \sqrt[4]{\dfrac{Vmc_k d_k}{3Z_k^t \gamma_1}} < f_k^{\min} \\ f_k^{\max}, if \sqrt[4]{\dfrac{Vmc_k d_k}{3Z_k^t \gamma_1}} > f_k^{\max} \\ \sqrt[4]{\dfrac{Vmc_k d_k}{3Z_k^t \gamma_1}}, \text{otherwise}, \end{cases} \quad (25)$$

and

$$(f_r^t)^* = \begin{cases} f_r^{\min}, if \sqrt[4]{\dfrac{V\phi \sum_{k=1}^{K} a_k^t}{3Y_r^t \gamma_2}} < f_r^{\min} \\ f_r^{\max}, if \sqrt[4]{\dfrac{V\phi \sum_{k=1}^{K} a_k^t}{3Y_r^t \gamma_2}} > f_r^{\max} \\ \sqrt[4]{\dfrac{V\phi \sum_{k=1}^{K} a_k^t}{3Y_r^t \gamma_2}}, \text{otherwise}, \end{cases} \quad (26)$$

respectively.

With the optimal CPU frequency established, the objective of problem P2 becomes a function of $\boldsymbol{a}^t$ and can consequently be transformed as follows:

$$\begin{aligned} \text{P4} \min_{\boldsymbol{a}^t} & \sum_{k=1}^{K} (P_k^t Z_k^t) + P_r^t Y_r^t + VU^t \\ \text{s.t. } & f_k^t = (f_k^t)^*, \forall k, \\ & f_r^t = (f_r^t)^*, \\ & (14). \end{aligned} \tag{27}$$

A straightforward strategy to resolve P4 involves traversing all possible client selection scenarios and then selecting the scheme that minimizes the objective function. However, the complexity of this approach escalates rapidly with an increase in the total number of clients. Therefore, we introduce an efficient algorithm designed to address P4 in Algorithm 1. In this proposed algorithm, during each global iteration, clients with $I_k^t = P_k^t Z_k^t - V\mu q_k^t$ lower than 0 are included into the initial set X_0^t. Thereafter, considering that learning latency is determined by straggler clients, these $|X_0^t|$ clients are incorporated one by one into the auxiliary selection set X_a^t in ascending order according to their total latency τ_k^t, thereby generating $|X_0^t|$ auxiliary selection sets. Here $|\cdot|$ signifies the count of elements within the set. These $|X_0^t|$ auxiliary selection sets are subsequently accumulated in the client selection set $\mathcal{X}^t$. We then compute the value of the objective function of P4 for each auxiliary selection set within $\mathcal{X}^t$ and select the optimal auxiliary selection set $(X_a^t)^*$ that minimizes the objective function of P4. Utilizing our proposed algorithm, throughout each global iteration, only $|X_0^t|$ computations of the objective function are required to attain the optimal solution. Consequently, this represents a significantly lower complexity compared to the exhaustive traversal method.

Algorithm 1 Client Selection Algorithm

```
Input: $Z_k^t = 0, \forall k, Y_r^t = 0$
Set $X_0^t = \varnothing, X_a^t = \varnothing, \mathcal{X}^t = \varnothing$
for $k \in \mathcal{K}$ do
    Calculate $I_k^t = P_k^t Z_k^t - V\mu q_k^t$
    if $I_k^t \leq 0$ then
        $X_0^t = X_0^t \cup \{k\}$
    end if
end for
Rank the clients in $X_0^t$ according to their $\tau_k^t$. Therefore we have $\tau_1^t \leq \tau_2^t \leq \ldots \leq \tau_{|X_0^t|}^t$
for $x \in X_0^t$ do
    Update $X_a^t = X_a^t \cup \{x\}$
    Add $X^t$ to $\mathcal{X}^t$, i.e., $\mathcal{X}^t = \mathcal{X}^t \cup \{X^t\}$
    Calculate $J(X_a^t) = \sum_{k \in X_a^t} (P_k^t Z_k^t) + P_r^t Y_r^t + VU^t$
end for
Find $(X_a^t)^* = \arg\min_{X_a^t \in \mathcal{X}^t} (J(X_a^t))$
Return $(\boldsymbol{a}^t)^*$, where $(a_k^t)^* = 1\{k \in (X_a^t)^*\}, \forall k$
```

3.3. Analysis of the Proposed Optimization Scheme's Optimality

Given the trade-off between minimizing the time-averaged cost and reducing power consumption violations, the analysis of the proposed optimization strategy's optimality is provided herein.

Theorem 2. *The average cost function satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1} E[U^t|\mathbf{\Theta}^t] \leq \frac{C_2}{V} + \varphi^*, \tag{28}$$

where $C_2 \geq C_1 + \sum_{k=1}^{K} Z_k^{t,\max} \max(P_k^{\max} - \bar{P}_k) + Y_r^{t,\max} \max(P_r^{\max} - \bar{P}_r)$*, and* φ^* *is the optimal solution of problem P1.*

Proof. The proof is given in Appendix B. □

Theorem 3. *Assume* $E[U^t|\mathbf{\Theta}^t] \geq \varphi^{\min}$*. The power consumption of each client k and the server are bounded by* $T\bar{P}_k + \sqrt{2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}}$ *and* $T\bar{P}_r + \sqrt{2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}}$*, respectively.*

Proof. The proof is given in Appendix C. □

Theorem 2 elucidates that the discrepancy between the objective value yielded by the proposed algorithm and the original optimal value is less than or equal to $O(1/V)$. This suggests that the cost determined by the proposed optimization scheme can approximate the original optimal value to an arbitrary degree through the augmentation of the control parameter V. In accordance with Theorem 3, the energy deficit queues of all clients and the server adhere to an upper limit of $O(\sqrt{V})$ at any iteration, a limit that escalates in accordance with the control parameter V. Nonetheless, an excessively large value of V may result in an unduly large upper boundary for the virtual power deficit queue backlog, which could lead to power consumption surpassing the power budget. In summary, the proposed algorithm delivers a $[O(1/V), O(\sqrt{V})]$ trade-off between cost and power consumption, a balance that can be managed by adjusting the parameter V.

Theorem 4. *Virtual queue of each client k and the server satisfies:*

$$\lim_{T\to\infty} \frac{Z_k^T}{T} = 0, \lim_{T\to\infty} \frac{Y_r^T}{T} = 0. \tag{29}$$

Proof. The proof is given in Appendix D. □

Theorem 4 indicates that the virtual power deficit queue backlog is bounded as the global iteration approaches infinity, i.e., all virtual queues remain mean rate stable across the FL iteration.

4. Experiment Result and Analysis

4.1. Experiment Settings

In the conducted experiment, FL was implemented using PyTorch, considering a system setup in which K clients are randomly positioned within a circular area of a 500 m radius with a central server. The path loss model is defined as $128.1 + 37.6\log_{10} i + \psi$, where i represents the distance between a client and the server in kilometers, while ψ is a Gaussian random variable exhibiting a variance of 8 dB. The total bandwidth, B, is set to 100 MHz, with the noise power spectral density $N_0 = -174$ dBm/Hz.

The power used for uploading the local model is arbitrarily assigned between 10 and 20 dBm. The model size s is set as 1 Mbit. For all clients, the number of local iterations in each global iteration m is set to 1. The number of CPU cycles necessary for processing a sample per client is randomly distributed within the range of $[1, 3] \times 10^4$ cycles/sample. Average power constraints are established at $\bar{P}_k = 100$ mW and $\bar{P}_r = 500$ mW. The decision parameter V is assigned the value of 10, with a justification provided later. The CPU frequency range of the clients and the central server, f_k^t and f_r^t, span from 0.1 GHz to 2.5 GHz and from 0.1 GHz to 3.3 GHz, respectively. Furthermore, the capacitance coefficients for the clients and the server, the price parameter of the cost function, and the number of CPU cycles needed to perform a single summation are set to $\gamma_1 = \gamma_2 = 10^{-28}$, $\mu = 1.6 \times 10^{-3}$, and $\phi = 10^6$.

The MNIST dataset [21] was employed for the experiment, consisting of 60,000 training samples and 10,000 test samples with 10 label classes from 0 to 9. Each client's local dataset was assembled by randomly selecting one or two label classes from the MNIST dataset with $d_k = 100$ samples. A multi-layer perceptron (MLP) model with a single hidden layer containing 64 nodes was utilized, with ReLU as the activation function. The learning rate was set to 0.01, and the batch size was 10.

To demonstrate the advantage of our proposed algorithm, we introduce the following three algorithms as comparison benchmarks:

- Selected All: In this algorithm, all the clients are selected in each global iteration. The CPU frequency for both the clients and the central server is consistently set at their maximum values in every global iteration.
- Greedy: For a rational comparison with our proposed algorithm, the long-term average number of client selected per round is tuned to be consistent with that of our proposed algorithm in this comparative algorithm. As such, we establish a client selection latency threshold. Clients are subsequently chosen one by one in ascending order based on their individual total latency τ_k^t until the learning latency τ^t surpasses the preset client-selection latency threshold. Furthermore, with the prerequisite of adhering to the CPU frequency constraint, all participating clients and servers maintain a constant power level, identical to the long-term power constraint.
- Random: In this comparative algorithm, clients are randomly selected in each round. The number of clients selected is maintained at a constant value, which is equal to the average number per round in our proposed algorithm. Aside from this variation, all other configurations align with those of the Greedy algorithm.

4.2. Analysis of Experimental Results

Conceptually, reducing the time required for each global iteration and minimizing the impact of the non-IID nature on the model convergence speed enables the training model to reach a specific accuracy more rapidly within a given learning time. Figure 2 demonstrates how the test accuracy of our proposed algorithm and comparative algorithms varies with the learning time under the number of clients $K = 100$. It is apparent that the proposed algorithm exhibits a performance almost equivalent to the Selected All algorithm in terms of convergence speed. Even though all the clients participate in each global iteration, fostering a swift convergence speed, the effects of the non-IID nature stemming from each client's dataset cannot be negated, thereby undermining its performance. Conversely, in our proposed model, clients with more label classes in their dataset may inherently have a higher selection priority. Simultaneously, the mean rate stable properties of the virtual queue in the proposed algorithm ensure fairness for clients with fewer label classes. Our proposed model significantly outperforms both the Greedy and Random algorithms. In the Greedy algorithm, while the impact of straggler clients is mitigated, it does not address the influence of the non-IID nature. Since the presence of the non-IID nature and straggler clients are not taken into account, the convergence speed of the Random algorithm is impeded.

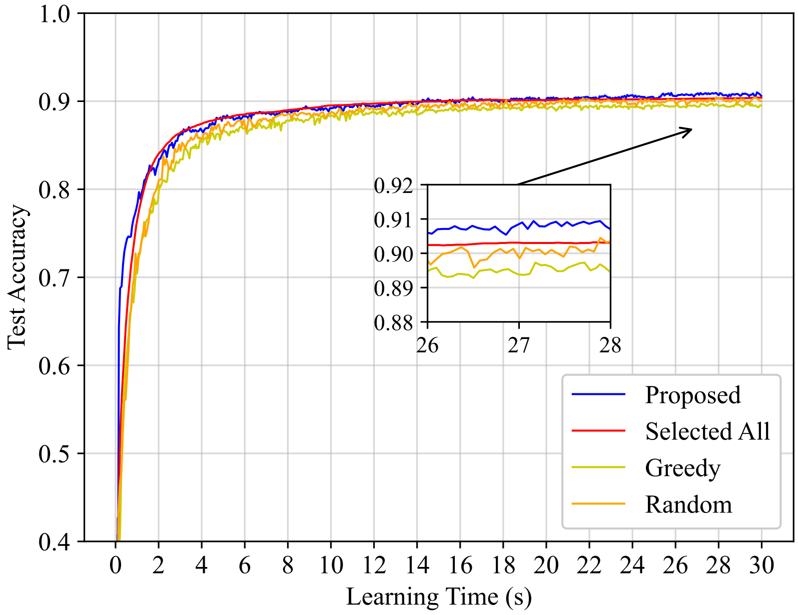


Figure 2. Test accuracy versus learning latency with the number of clients $K = 100$.

Figure 3 demonstrates the corresponding average power consumption of the client side and the server side in each global iteration. The Selected All algorithm, when compared to other methods, exhibits substantially larger power consumption, primarily because it lacks a power constraint. However, due to the mean rate stable characteristic of the virtual queue, as demonstrated by Theorem 4, the power consumption under our proposed algorithm adheres to the long-term power constraint. Notably, the average power consumption under our proposed algorithm is approximately similar to that observed in the Greedy and Random algorithms.

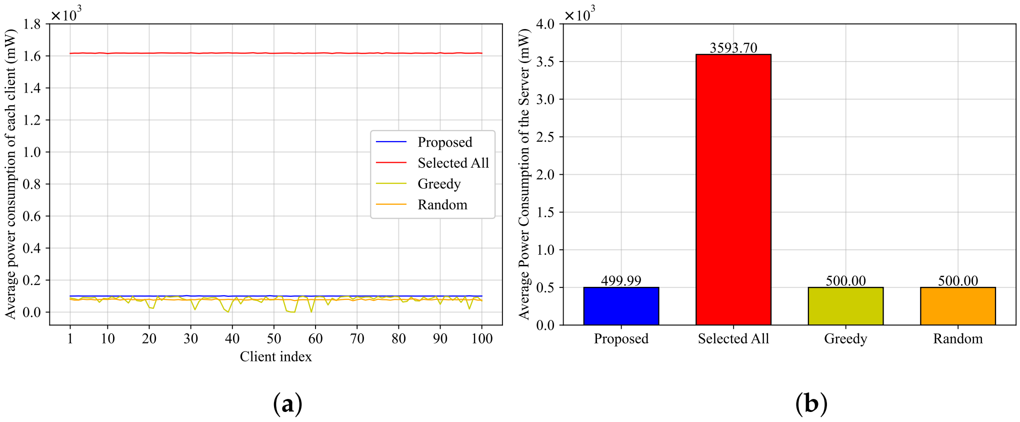


Figure 3. Average power consumption of each client and the server. (**a**) Each client. (**b**) Server.

To further validate the proposed optimization scheme, its performance is examined under a varying total number of clients K, as depicted in Figure 4 and Table 1. In the conducted experiment, we take into account the average test accuracy during the concluding half second when the learning time spans 30 seconds. Accompanying the increase in the total number of clients, the average number of clients selected per iteration also escalates, which, under normal circumstances, should enhance test accuracy. Nonetheless, the increase in the number of clients may cause a corresponding increment in each iteration's training duration. As a consequence, the total iterations that can be carried out within a fixed time duration may decrease, thereby potentially reducing accuracy. Hence, the test accuracy does not bear a linear relationship with the total number of clients, which can be observed from Figure 4. Nevertheless, our proposed algorithm continues to surpass comparative algorithms, as it adeptly manages non-IID characteristics and straggler clients. Furthermore, our proposed algorithm exhibits a consistent ability to maintain low power consumption as the total number of clients increases. This finding aligns with our previous analysis, demonstrating that the virtual power deficit queues are mean rate stable in our proposed algorithm.

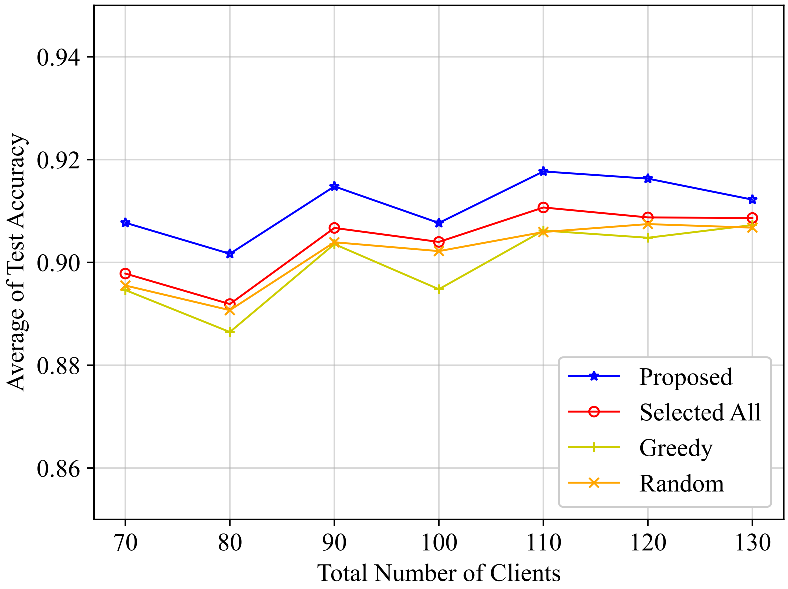
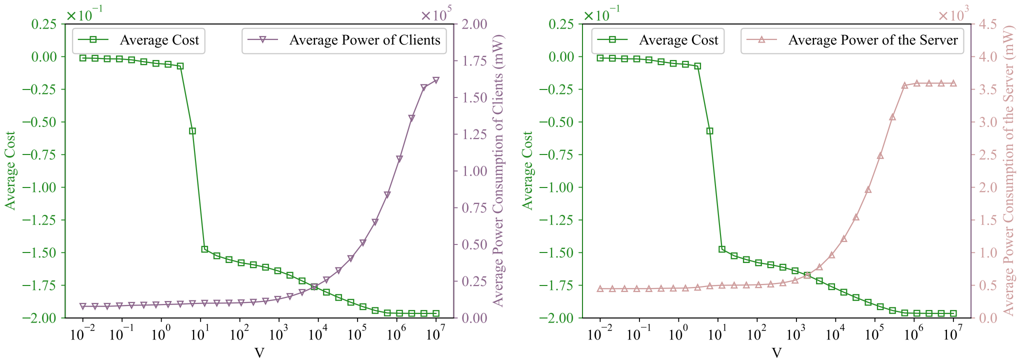


Figure 4. Average of test accuracy versus total number of clients.

Table 1. Average power consumption of clients and the server versus total number of clients.

Average Power Consumption	Total Number of Clients	Proposed	Selected All	Greedy	Random
clients	70	7016.21 mW	113,212.66 mW	5616.25 mW	5600.00 mW
	80	8020.03 mW	129,386.61 mW	6401.84 mW	6400.00 mW
	90	9033.43 mW	145,559.47 mW	7229.53 mW	7200.01 mW
	100	10,036.40 mW	161,733.26 mW	7743.76 mW	7800.00 mW
	110	11,030.69 mW	177,907.72 mW	8439.06 mW	8400.01 mW
	120	12,053.63 mW	194,080.37 mW	9336.41 mW	9300.01 mW
	130	13,047.53 mW	210,254.14 mW	9940.70 mW	9900.01 mW
server	70	499.86 mW	3593.70 mW	500.00 mW	500.00 mW
	80	499.94 mW	3593.70 mW	500.00 mW	500.00 mW
	90	499.94 mW	3593.70 mW	500.00 mW	500.00 mW
	100	499.99 mW	3593.70 mW	500.00 mW	500.00 mW
	110	500.03 mW	3593.70 mW	500.00 mW	500.00 mW
	120	500.04 mW	3593.70 mW	500.00 mW	500.00 mW
	130	500.24 mW	3593.70 mW	500.00 mW	500.00 mW

Figure 5 depicts the variation in the average power consumption and the average cost in correlation with the control parameter V within our proposed optimization scheme. A clear observation is that the average cost experiences a decrease, while the average power consumption undergoes an increase with an escalating control parameter V. This aligns with the $[O(1/V), O(\sqrt{V})]$ cost–power trade-off indicated by Theorems 2 and 3. Figure 6 illustrates the variation in the optimal average number of selected clients per iteration with the control parameter V. As previously stated, a control parameter value of $V = 10$ was selected for the experiment. This choice was made because it yields an appropriate average number of selected clients to effectively address the accuracy degradation incited by non-IID characteristics, along with the low average cost and power consumption.



(**a**) (**b**)

Figure 5. The impact of V. (**a**) Average power consumption of clients and average cost versus control parameter V. (**b**) Average power consumption of the server and average cost versus control parameter V.

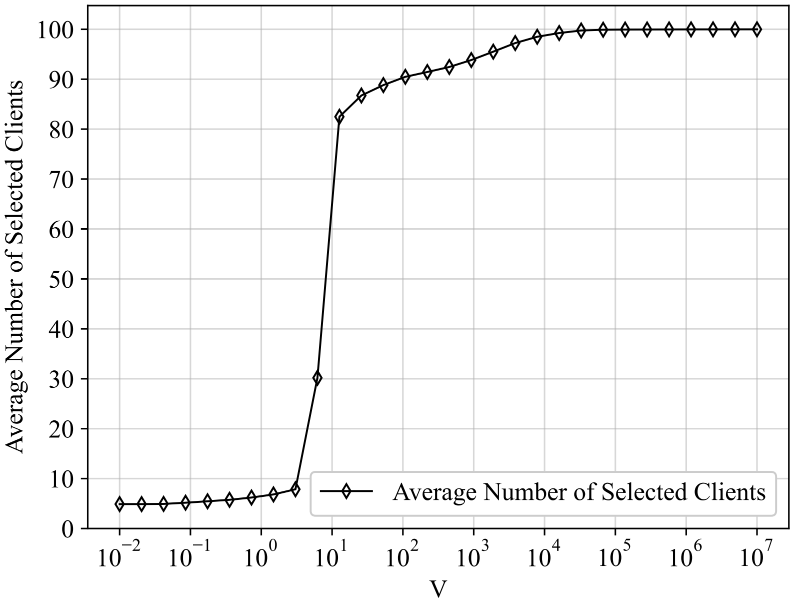


Figure 6. Average number of selected clients versus control parameter V.

5. Discussion

In this paper, we explored a problem involving the selection of clients and the concurrent control of the CPU frequency for both the selected clients and the server within wireless FL networks. Lyapunov optimization theory was applied to transform the original problem into a per-iteration problem, which facilitated the design of an algorithm for problem resolution. Theoretical analysis offers performance guarantees, wherein controlling the parameter V empowers us to reduce cost while minimizing power consumption. Simulation results demonstrated that the proposed algorithm outperforms benchmark algorithms in terms of test accuracy by mitigating the impact of non-IID characteristics and straggler clients. By managing the virtual queues, the proposed algorithm was able to adhere to long-term power constraints. Furthermore, the simulation results verified that our proposed algorithm successfully realized the $[O(1/V), O(\sqrt{V})]$ cost–power trade-off.

It is noteworthy that this study is currently confined to a simple star network topology. Expanding our analysis to encompass more intricate network structures such as hierarchical networks and multi-base station networks would undoubtedly enhance its applicability. Additionally, in practical wireless networks, client participation in learning can be affected by factors such as mobility, network congestion, or power availability fluctuations, potentially leading to client dropouts from the FL process and thereby impacting overall learning performance. Hence, the implications of client dropouts merit further investigation.

Author Contributions: Conceptualization, Z.Z.; methodology, Z.Z., Y.L. and F.W.; software, Z.Z. and S.S.; validation, Z.Z., Y.L. and S.S.; formal analysis, Z.Z., Y.L. and F.W.; investigation, Z.Z., Y.L. and Y.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Y.L. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant 61801433 and Grant 62101505, and in part by the Science and Technology Planning Project of Henan Province under Grant 222102210003 and Grant 222102210278.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FL	Federated learning
IoT	Internet of Things
non-IID	Non-independent and identically distributed
IID	Independent identically distributed
OFDMA	Orthogonal frequency-division multiple access
MLP	Multi-layer perceptron
ReLU	Rectified Linear Unit

Appendix A

Since $\Delta(\mathbf{\Theta}^t)$ plays a crucial role in the proof, we first bound $\Delta(\mathbf{\Theta}^t)$. Plugging (17)–(19) into $\Delta(\mathbf{\Theta}^t)$, we have:

$$\begin{aligned}
\Delta(\mathbf{\Theta}^t) &= \tfrac{1}{2}E[\sum_{k=1}^{K}(Z_k^{t+1})^2 + (Y_r^{t+1})^2 - \sum_{k=1}^{K}(Z_k^t)^2 - (Y_r^t)^2|\mathbf{\Theta}^t] \\
&= \tfrac{1}{2}E[\sum_{k=1}^{K}\{\max\{P_k^t - \bar{P}_k + Z_k^t, 0\}\}^2 - \sum_{k=1}^{K}(Z_k^t)^2 + \{\max\{P_r^t - \bar{P}_r + Y_r^t, 0\}\}^2 - (Y_r^t)^2|\mathbf{\Theta}^t] \\
&\leq \tfrac{1}{2}E[\sum_{k=1}^{K}(P_k^t - \bar{P}_k + Z_k^t)^2 - \sum_{k=1}^{K}(Z_k^t)^2 + (P_r^t - \bar{P}_r + Y_r^t)^2 - (Y_r^t)^2|\mathbf{\Theta}^t] \\
&\leq C_1 + \sum_{k=1}^{K} Z_k^t E[P_k^t - \bar{P}_k|\mathbf{\Theta}^t] + Y_r^t E[P_r^t - \bar{P}_r|\mathbf{\Theta}^t],
\end{aligned} \tag{A1}$$

where the first inequality is due to $\{\max\{a,0\}\}^2 \leq a^2$. Thus, we have:

$$\Delta(\mathbf{\Theta}^t) + VE[U^t|\mathbf{\Theta}^t] \leq C_1 + \sum_{k=1}^{K} Z_k^t E[P_k^t - \bar{P}_k|\mathbf{\Theta}^t] + Y_r^t E[P_r^t - \bar{P}_r|\mathbf{\Theta}^t] + VE[U^t|\mathbf{\Theta}^t], \tag{A2}$$

This concludes the proof.

Appendix B

According to Appendix A, we have:

$$\begin{aligned}
&\Delta(\mathbf{\Theta}^t) + VE[U^t|\mathbf{\Theta}^t] \\
&\leq C_1 + \sum_{k=1}^{K} Z_k^{t,\max}\max(E[P_k^{\max} - \bar{P}_k|\mathbf{\Theta}^t]) + Y_r^{t,\max}\max(E[P_r^{\max} - \bar{P}_r|\mathbf{\Theta}^t]) + V\varphi^* \\
&\leq C_2 + V\varphi^*
\end{aligned} \tag{A3}$$

Summing (A3) over T global iterations, we obtain:

$$\sum_{t=0}^{T-1} VE[U^t|\mathbf{\Theta}^t] \leq TC_2 + TV\varphi^* - \sum_{t=0}^{T-1} \Delta(\mathbf{\Theta}^t). \tag{A4}$$

Next we sum $\Delta(\mathbf{\Theta}^t)$ over T global iterations. The following formula can be derived from (20):

$$\sum_{t=0}^{T-1} \Delta(\mathbf{\Theta}^t) = L(\mathbf{\Theta}^T) - L(\mathbf{\Theta}^0). \tag{A5}$$

Plugging (A5) into (A4), we have:

$$\sum_{t=0}^{T-1} VE[U^t|\mathbf{\Theta}^t] \leq TC_2 + TV\varphi^* - L(\mathbf{\Theta}^T) + L(\mathbf{\Theta}^0). \tag{A6}$$

Dividing by TV, using the fact that $L(\mathbf{\Theta}^T) \geq 0$, $L(\mathbf{\Theta}^0) = 0$, we have:

$$\frac{1}{T}\sum_{t=0}^{T-1} E[U^t|\mathbf{\Theta}^t] \leq \frac{C_2}{V} + \varphi^*. \tag{A7}$$

This concludes the proof.

Appendix C

Rearranging (A6), we obtain:

$$L(\mathbf{\Theta}^T) - L(\mathbf{\Theta}^0) \leq TC_2 + TV\varphi^* - \sum_{t=0}^{T-1} VE[U^t|\mathbf{\Theta}^t]. \tag{A8}$$

Substituting $E[U^t|\mathbf{\Theta}^t] \geq \varphi^{\min}$ into (A8), we have:

$$L(\mathbf{\Theta}^T) \leq TC_2 + TV\varphi^* - TV\varphi^{\min}. \tag{A9}$$

Due to (19), we obtain:

$$(Z_k^T)^2 + (Y_r^T)^2 \leq \sum_{k=1}^{K} (Z_k^T)^2 + (Y_r^T)^2 \leq 2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}. \tag{A10}$$

Thus, we have:

$$(Z_k^T)^2 \leq 2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}, (Y_r^T)^2 \leq 2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}. \tag{A11}$$

Next we bound the virtual power deficit queues. From (17) and (18), we have:

$$Z_k^{t+1} = Z_k^t - \min\{\bar{P}_k - P_k^t, Z_k^t\}, Y_r^{t+1} = Y_r^t - \min\{\bar{P}_r - P_r^t, Y_r^t\}. \tag{A12}$$

Rearranging terms and summing both sides over t global iterations, we have:

$$Z_k^t - Z_k^0 = -\sum_{\tau=0}^{t-1} \min\{\bar{P}_k - P_k^\tau, Z_k^t\} \geq \sum_{\tau=0}^{t-1} (P_k^\tau - \bar{P}_k) = \sum_{\tau=0}^{t-1} P_k^\tau - t\bar{P}_k, \tag{A13}$$

$$Y_r^t - Y_r^0 = -\sum_{\tau=0}^{t-1} \min\{\bar{P}_r - P_r^\tau, Y_k^t\} \geq \sum_{\tau=0}^{t-1} (P_r^\tau - \bar{P}_r) = \sum_{\tau=0}^{t-1} P_r^\tau - t\bar{P}_r. \tag{A14}$$

Thus, we have:

$$Z_k^T \geq \sum_{\tau=0}^{T-1} P_k^\tau - T\bar{P}_k, Y_r^T \geq \sum_{\tau=0}^{T-1} P_r^\tau - T\bar{P}_r. \tag{A15}$$

Thus far, we have bounded the virtual power deficit queues. Plugging (A15) into (A11), we have:

$$\sum_{\tau=0}^{T-1} P_k^\tau \leq T\bar{P}_k + \sqrt{2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}}, \tag{A16}$$

$$\sum_{\tau=0}^{T-1} P_r^\tau \leq T\bar{P}_r + \sqrt{2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}}. \tag{A17}$$

This concludes the proof.

Appendix D

Rearranging terms of (A11), we have:

$$Z_k^T \leq \sqrt{2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}}, Y_r^T \leq \sqrt{2TC_2 + 2TV\varphi^* - 2TV\varphi^{\min}}. \tag{A18}$$

Dividing by T and taking limits of both sides, we have:

$$\lim_{T\to\infty} \frac{Z_k^T}{T} = 0, \lim_{T\to\infty} \frac{Y_r^T}{T} = 0. \tag{A19}$$

This concludes the proof.

References

1. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017.
2. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205–1221. [CrossRef]
3. Gao, W.; Zhao, Z.; Min, G.; Ni, Q.; Jiang, Y. Resource allocation for latency-aware federated learning in industrial internet of things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8505–8513. [CrossRef]
4. Hu, R.; Guo, Y.; Gong, Y. Energy-efficient distributed machine learning at wireless edge with device-to-device communication. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022.
5. Zhao, Z.; Feng, C.; Hong, W.; Jiang, J.; Jia, C.; Quek, T.Q.; Peng, M. Federated learning with non-iid data in wireless networks. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 1927–1942. [CrossRef]
6. Nishio, T.; Yonetani, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019.
7. Leng, J.; Lin, Z.; Ding, M.; Wang, P.; Smith, D.; Vucetic, B. Client scheduling in wireless federated learning based on channel and learning qualities. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 732–735. [CrossRef]
8. Yang, Z.; Chen, M.; Saad, W.; Hong, C.S.; Shikh-Bahaei, M. Energy efficient federated learning over wireless communication networks. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 1935–1949. [CrossRef]
9. Yao, J.; Ansari, N. Enhancing federated learning in fog-aided IoT by CPU frequency and wireless power control. *IEEE Internet Things J.* **2020**, *8*, 3438–3445. [CrossRef]
10. Zhao, Z.; Xia, J.; Fan, L.; Lei, X.; Karagiannidis, G.K.; Nallanathan, A. System optimization of federated learning networks with a constrained latency. *IEEE Trans. Veh. Technol.* **2021**, *71*, 1095–1100. [CrossRef]
11. Yu, L.; Albelaihi, R.; Sun, X.; Ansari, N.; Devetsikiotis, M. Jointly optimizing client selection and resource management in wireless federated learning for internet of things. *IEEE Internet Things J.* **2021**, *9*, 4385–4395. [CrossRef]
12. Chen, M.; Yang, Z.; Saad, W.; Yin, C.; Poor, H.V.; Cui, S. A joint learning and communications framework for federated learning over wireless networks. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 269–283. [CrossRef]
13. Liu, S.; Yu, G.; Chen, X.; Bennis, M. Joint user association and resource allocation for wireless hierarchical federated learning with iid and non-iid data. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 7852–7866. [CrossRef]
14. Huang, T.; Lin, W.; Wu, W.; He, L.; Li, K.; Zomaya, A.Y. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1552–1564. [CrossRef]
15. Guo, K.; Chen, Z.; Yang, H.H.; Quek, T.Q. Dynamic scheduling for heterogeneous federated learning in private 5g edge networks. *IEEE J. Sel. Top. Signal Process.* **2021**, *16*, 26–40. [CrossRef]

16. Battiloro, C.; Di Lorenzo, P.; Merluzzi, M.; Barbarossa, S. Lyapunov-based optimization of edge resources for energy-efficient adaptive federated learning. *IEEE Trans. Green Commun. Netw.* **2022**, *7*, 265–280. [CrossRef]
17. Xu, J.; Wang, H. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 1188–1200. [CrossRef]
18. Ji, Y.; Kou, Z.; Zhong, X.; Li, H.; Yang, F.; Zhang, S. Client Selection and Bandwidth Allocation for Federated Learning: An Online Optimization Perspective. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022.
19. Ji, X.; Tian, J.; Zhang, H.; Wu, D.; Li, T. Joint Device Selection and Bandwidth Allocation for Cost-Efficient Federated Learning in Industrial Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 9148–9160. [CrossRef]
20. Neely, M.J. Stochastic network optimization with application to communication and queueing systems. *Synth. Lect. Commun. Netw.* **2010**, *3*, 1–211.
21. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.