Turkish Journal of Electrical Engineering & Computer Sciences
http://journals.tubitak.gov.tr/elektrik/
Research Article
Turk J Elec Eng & Comp Sci
(2023) 31: 612 – 625

doi:10.55730/1300-0632.4005

Unbiased federated learning in energy harvesting error-prone channels

Zeynep ÇAKIR[1,2,*], **Elif Tuğçe CERAN**[2]
[1]Roketsan Missiles Industries and Trade Inc., Ankara, Turkey
[2]Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey

Received: 18.12.2022 • **Accepted/Published Online:** 30.04.2023 • **Final Version:** 28.05.2023

Abstract: Federated learning (FL) is a communication-efficient and privacy-preserving learning technique for collaborative training of machine learning models on vast amounts of data produced and stored locally on the distributed users. This paper investigates unbiased FL methods that achieve a similar convergence as state-of-the-art methods in scenarios with various constraints like an error-prone channel or intermittent energy availability. For this purpose, we propose FL algorithms that jointly design unbiased user scheduling and gradient weighting according to each user's distinct energy and channel profile. In addition, we exploit a prevalent metric called the age of information (AoI), which quantifies the staleness of the gradient updates at the parameter server and adaptive momentum attenuation to increase the accuracy and accelerate the convergence for nonhomogeneous data distribution of participant users. The effect of AoI and momentum on fair FL with heterogeneous users on various datasets is studied, and the performance is demonstrated by experiments in several settings.

Key words: Federated learning, energy harvesting, age of information, momentum, wireless communications

1. Introduction

The need to store, process, and use the big data produced by various types of devices is one of the main focuses of up-to-date machine learning (ML) applications. While traditional ML approaches require an orchestral server that collects, stores, and processes the data produced by the devices, it is neither feasible nor efficient for a centralized server to work with that substantial amount of data. In addition, the data produced by a device can be sensitive and private, and privacy violations may occur because of the need to upload it. Motivated by providing a solution to these problems, Google researchers introduced a concept named *"federated learning"* (FL) [1], and then it became a popular and promising method for private and communication-efficient machine learning/deep learning to train ML models on vast amounts of data produced and stored locally on the participant users. It allows users to be part of a global machine learning model training without sharing their local data. Training is performed using distributed stochastic gradient descent (SGD) coordinated by a central server responsible for the global model. To train the global model, each user uses their local dataset, and the goal is to train a machine learning model on the combined dataset.

While designing a method for an FL setup with several communicative constraints, such as error-prone wireless channels and intermittent availability of energy, ensuring that there is no bias between heterogeneous users is an important performance indicator in terms of convergence and accuracy. A prevalent metric called the Age of Information (AoI) quantifies the staleness of the information at the destination. It is defined as the time elapsed since the generation time of the most recent status update packet successfully received at the destination

*Correspondence: cakirzeynepp@gmail.com

[2, 3]. Especially for applications that require timely updates, such as IoT, machine-type communications, and FL, AoI is a critical performance indicator. In the FL framework, AoI can be considered as the time elapsed between receiving the local updates from a participant user. It is an essential and unique metric for increasing the performance of FL algorithms and provides a new perspective to existing methods and applications.

In this work, we consider AoI-aware FL algorithms that jointly design unbiased user scheduling and gradient weighting according to the energy and channel profile of each user. Extensive experiments show that the proposed algorithms provide high test accuracy and convergence guarantees comparable to the state-of-the-art algorithms with no energy or channel constraints.

2. Related work

Since first introduced by Google researchers in 2016 [1], FL and its applications have become a popular approach for privacy and energy concerns. A training scenario where K users work together is considered to train a model in FL. Each user has a local dataset, and the goal is to train a machine learning model on the combined dataset. Training is performed using distributed stochastic gradient descent (SGD) coordinated by a central server responsible for the global model. The server sends users the current estimate of the model parameters after each training round. Users then update their model by calculating a local gradient on local datasets. The server then collects users' local updates, updates the global model, and returns the updated model to the users. Along with the first introduction of FL by Konecny et al. [4], [1], another reference guide is presented by McMahan et al. [5] in 2017. This study explains the concept of FL, and the *FederatedAveraging* algorithm, which forms the basis of many following studies, is introduced. Each participant user performs local training on the current global model in this algorithm using its local dataset. The parameter server takes a weighted average of the locally trained model parameters. The convergence analysis of this algorithm was carried out by Li et al. in [6], and it was carried out separately for the datasets that are equally and not equally distributed. Additionally, a method called *"momentum"* for increasing the efficiency of stochastic gradient descent can be applied in FL [1]. It is an extension of the stochastic gradient descent method and is essential in accelerating the convergence or increasing the accuracy for nonhomogeneous data distribution on participant users. There are many applications of momentum in FL [7–9]. Xu et al. studied expanding the *FederatedAveraging* algorithm introduced in [5], by adding a momentum factor to it, supported by the convergence analysis and experiment [7]. Kim et al. proposed another method named *FedAGM*, to deal with the challenge of low convergence rate [8]. *FedAGM* uses momentum to accelerate the model training process and aims to improve the convergence and accuracy of the model.

Energy harvesting, which comprises the gathering of electrical energy without wires using time-dependent electric, magnetic, or electromagnetic fields, has been indicated as a feasible preference for numerous communication systems [10–12]. Güler et al. [13, 14] studied a sustainable FL framework by considering energy harvesting users, which motivated adding new measures to the FL. Gündüz et al. [15] studied federated edge learning (FEEL), inspected the divergence of existing coding and communication schemes and learning algorithms, and suggested new approaches to combine these concepts. Özfatura et al. [16] studied the demonstration of how taking wireless channel characteristics such as resource allocation, scheduling, etc. into consideration may considerably enhance the speed and overall performance of distributed learning approaches. In [17], we studied an FL setup for IID datasets in which users harvest energy from the environment and collaboratively train

[1] Brownlee J (2021). Gradient Descent With Momentum from Scratch [online]. Website https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/ [accessed 12 03 2023].

a machine learning model under the constraints of intermittent energy arrivals and channel availability. The main focus was to develop an algorithm that achieves a similar convergence as modern FL methods. FEEL is a version of FL performed by wireless devices, with constrained energy and bandwidth, on their local datasets, supported by a remote parameter server. A concept of weighting model differences by a *"cooldown multiplier,"* based on the time elapsed between the two most recent energy arrivals, is introduced [18].

The AoI was introduced by Kaul et al. in [2] and [3], to adjust the freshness of information in status-update systems. The AoI quantifies the staleness of the information at the destination and is defined as the time elapsed since the generation time of the most recent status update packet successfully received at the destination. In the FL area, AoI can be considered as the time elapsed between receiving the local updates from a participant user. Yang et al. studied a metric called *"age of update"* (AoU) and proposed a scheduling policy to find the minimum AoU, with the constraints of maximum transmit power, avoiding interference and rate exceeding a threshold [19], Büyükateş et al. studied the metric of the average AoI of each client to define the AoI-optimal number of total and earliest participant users. They suggested a strategy that ensures timeliness and reduces average iteration durations [20]. Liu et al. studied an age-aware communication method for FL over wireless networks to achieve efficient model training over non-IID data [21].

This work explores FL strategies that achieve a similar convergence as state-of-the-art FL methods in contexts with diverse restrictions, such as error-prone channels or intermittent energy availability. To the best of our knowledge, this is the first work that studies the effect of AoI on FL with channel and energy-aware scheduling combined with the dynamic weighting of the updates and the acceleration of AoI-aware momentum.

3. System model

We consider an FL system with K distinct users on the network, which collaborate through a centralized parameter server to train an ML model. These users are connected to the central parameter server over error-prone channels and receive intermittent energy through energy harvesting. Energy can be provided by the environment in various ways, where the arrival of energy can be either deterministic or stochastic. The channels between the users and the parameter server are either available or unavailable in each time slot, depending on the channel error probabilities. A user should have sufficient energy to compute local model updates and transmit those updates to the parameter server. In addition to the energy constraint, channel availability is also a criterion for the user to participate in the training. A user can join the training only if there is enough energy and the channel is available. In such terms, the parameter server may give particular importance to users who can participate in the process more frequently and produce better results than the other users. This situation leads to a bias in FL. It is a situation not desired because the parameter server would prefer the users that are more advantageous than the other users in terms of participation, resulting in a performance loss due to heterogeneous data distribution. The aim is to minimize a global loss function under the energy and channel state awareness, guaranteeing that there is no bias or unfairness between users. The illustration of the system model is in Figure 1.

3.1. Federated learning model

Assuming that a user $i \in \{1, \ldots, K\}$ has D_i data points in its local dataset, the total number of data points for all users can be defined as D. With these definitions, the global loss function can be defined as follows:

$$F(w) = \sum_{i=1}^{K} p_i F_i(w), \tag{1}$$

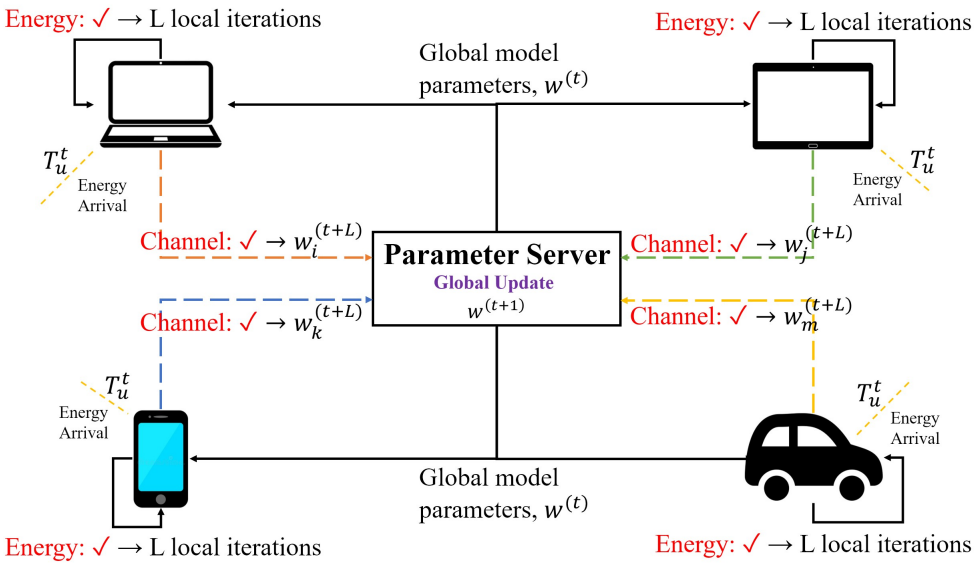

Figure 1. System model.

In this equation, K is the number of users, p_i is the ratio of the user i's local dataset size to the entire dataset size ($p_i = \frac{D_i}{D}, \sum_{i=1}^{K} p_i = 1$), w is the up-to-date estimation of the model parameters, and the function $F_i(w)$ represents the local loss function. The local loss function of user i is defined as follows:

$$F_i(w) = \frac{1}{D_i} \sum_{j=1}^{D_i} l(w, x_{ij}), \tag{2}$$

The value $l(w, x_{ij})$ in this equation indicates the loss of the point x_{ij} in user i in the local dataset.

Training is performed by using the distributed SGD method. In this method, the model parameters are constantly updated in the negative direction of the gradient. Estimation of the model parameters for the global round $t \in \{0, 1, 2, ..\}$ is represented by $w^{(t)}$. The parameter server sends the value $w^{(t)}$ to participating users. The number of local training iterations (local rounds) performed by the participant user is defined by L. Users $i \in \{1, 2, ..., K\}$ calculate a local stochastic gradient with L local iterations:

$$g_i(w^{(t)}, \xi_i^t) = \nabla F_i(w^{(t)}, \xi_i^{(t)}), \tag{3}$$

The value $\xi_i^{(t)}$ specifies a uniformly random sample from the local dataset. This ensures that the stochastic gradient is not biased. Under this assumption, the actual gradient value of user i can be defined as:

$$E_{\xi_i^{(t)}}[\nabla F_i(w^{(t)}, \xi_i^{(t)})] = \nabla F_i(w^{(t)}), \tag{4}$$

In this equation, the value $\nabla F_i(w^{(t)})$ specifies the gradient of the local loss function. The gradient of the global loss function is defined as follows:

$$\nabla F(w^{(t)}) = \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)}) \tag{5}$$

After users complete their local calculations, local gradient values are sent to the parameter server. The parameter server updates the model as follows:

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^{K} p_i g_i(w^{(t)}, \xi_i^{(t)}), \tag{6}$$

where η denotes the learning rate. After the update, the model is sent back to participating users, and the cycle continues until the global training process is complete. The proposed method in this work considers scheduling of the participating users and also modifies the Equation (6) in the sense that the local updates received from the participant users are calculated by using the momentum, resulting in the global update equation being altered, which is going to be explained in Section 4.

3.2. Energy and channel model

In this study, we consider users powered by energy harvested from the environment by energy-scavenging devices. It is assumed that a step in the SGD method, including the local gradient computation and transmission to the parameter server, costs each user a unit amount of energy. It is also assumed that each user has a unit battery that stores enough energy for one SGD step. The energy arrival process of user i at global round t is denoted by E_i^t: if there is an energy arrival, $E_i^t = 1$, otherwise $E_i^t = 0$. The distribution of energy arrivals varies depending on whether the harvesting process is deterministic or stochastic. In the case of deterministic energy arrival, users know when the energy will arrive. It is assumed that there is only one unit of energy arriving in the same global round. In the case of stochastic energy arrival, users do not know exactly when the energy will arrive, but they know the probabilistic model of the energy arrival process. As a stochastic energy arrival method, binary energy arrival is defined as a Bernoulli process with energy arrival probabilities as β_i for each user i. The user i receives one unit amount of energy with probability β_i per global round. The value of β_i is between 0 and 1 and may vary from user to user.

$$E_i^t = \begin{cases} 1, & \text{with the probability } \beta_i \\ 0, & \text{with the probability } 1-\beta_i \end{cases} \tag{7}$$

Users are assumed to deliver their local updates to the remote parameter server through an imperfect wireless channel. The channel state of each user is assumed to change randomly at each global round. Let Q_i^t denote whether the channel of user i is available at global round t or not, and is assumed to follow a Bernoulli process with channel error probabilities as q_i, $i \in \{1, \ldots, K\}$. The channel of user i is available with probability $1-q_i$. The value of q_i is between 0 and 1, which may vary from user to user.

$$Q_i^t = \begin{cases} 0, & \text{with the probability } q_i \\ 1, & \text{with the probability } 1-q_i. \end{cases} \tag{8}$$

T_i, referred to as the *"energy arrival parameter"*, specifies the frequency of the energy arrivals of the user i. For deterministic energy arrivals, it is a constant integer; but for stochastic energy arrivals, it is determined by dividing 1 by the channel availability probability $1-q_i$.

4. Proposed methods

This section presents the proposed FL algorithms for deterministic and stochastic energy arrival scenarios. We utilize momentum as an extension of the gradient descent optimization aiming to accelerate the convergence or

increase the accuracy for nonhomogeneous data distribution on participant users. AoI metric is also exploited in gradient update and momentum attenuation factor computation to train the model in a more balanced and accurate way. Next, two FL algorithms are going to be explained in detail.

4.1. Federated learning with deterministic energy arrivals

The scheduling process starts by checking the energy status of the user. If energy arrival occurs, the user randomly chooses a global round to join the training process and determines an integer J with a certain probability in the range of 0 to $T_i - 1$, and the user is scheduled for $(t+J)^{th}$ round. The value of this integer depends on the energy arrival parameter of the participating user and the error probability of the channel, which will be determined using the following probability distribution.

$$P(J=0) = \frac{1}{T_i - T_i q_i + q_i} \tag{9}$$

$$P(0 < J \leq T_i - 1) = \frac{1 - q_i}{T_i - T_i q_i + q_i} \tag{10}$$

The derivation of these probabilities is provided in Appendix S1. At each global round, it is checked whether users can participate in the learning process in accordance with the scheduling, taking the channel availability into account. The user can participate in the learning process if the channel is available at the current global round. If not, the user is scheduled to participate in the next global round.

The parameter server sends the model parameters to the participating users, and the users perform the learning process by making local gradient calculations. The information gathered from the user that participates in the training process less often is more critical and must be included in the process on a greater scale. In other words, if the time elapsed between the two most recent participation of the corresponding user is long, the scaling coefficient must be greater than the other users; since that user cannot participate in the training process frequently, the information obtained from that user must be important. To ensure this assumption, while scaling the local gradients, the AoI of the user is taken into account. AoI, denoted as Δ_i^t, is defined as the time elapsed between the most recent participation in the training process and the current global round t for user i. The scaling coefficient is defined as the ratio of the AoI of the corresponding user to the total AoI of the users, $\frac{\Delta_i^t}{\Delta^t}$. During the local training, user-specific local gradient values are scaled as follows:

$$g_i^{(t+L)} = \frac{\Delta_i^t}{\Delta^t}(g_i(w^{(t)}, \xi_i^t)) \tag{11}$$

We also introduce a parameter named *"momentum attenuation factor,"* aiming to regulate the quantity of previous data to incorporate in the update equation: a large value of the momentum attenuation factor means that the current update is strongly affected by the previous update, whereas a lower value means the reverse [2]. The momentum attenuation factor ranges from 0 to 1, and 0 corresponds to gradient descent without momentum. The momentum update is given as:

$$m_i(t+1) = \delta_i^t m_i(t) - \eta * g_i^{(t+L)} \tag{12}$$

[2]Brownlee J (2021). Gradient Descent With Momentum from Scratch [online]. Website https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/ [accessed 12 03 2023].

where the term $m_i(t)$ is generally referred to as *"velocity"*, which is an instrument of the momentum-included SGD calculations and includes the previous and current information provided by the user i. In addition, the momentum attenuation factor of participant user i is denoted as δ_i^t. Similar to the approach explained for the gradient scaling factor, if the user cannot participate in the training process because of its energy arrival or channel availability processes that user's previous data must be taken into account on a grander scale. To achieve that, the momentum attenuation factor must be set at a higher value, because the information gathered from that user becomes more critical due to the lack of participation in the process. On the other hand, if the user participates in the training process more frequently, the momentum attenuation factor must be determined as a smaller value. Consequently, the momentum attenuation factor is determined in the scheduling process according to the AoI of the corresponding user:

$$\delta_i^t = \begin{cases} 0.1, & \text{if } \Delta_i^t = 1 \\ 0.5, & \text{if } 1 < \Delta_i^t \leq T_i \\ 0.9, & \text{if } \Delta_i^t > T_i \end{cases} \tag{13}$$

where T_i is the energy arrival parameter. The participant user obtains the locally trained model parameters as follows:

$$w^{(t+L)} = w^{(t)} + m_i(t+1) \tag{14}$$

After the local learning is finished, the server sends locally trained model parameters, denoted as $w_i^{(t+L)}$. The parameter server updates the global model as follows:

$$w^{(t+1)} = \sum_{i \in S_t} w_i^{(t+L)} \tag{15}$$

In this equation, S_t represents the set of users who have successfully participated in the learning process. The algorithm is presented in Algorithm 1.

4.2. Federated learning with stochastic energy arrivals

The system architecture of the stochastic energy arrivals is the same as the deterministic energy arrivals. However, there is no need to determine J in scheduling, since the nature of the energy arrival is random, and it provides the desired unbiasedness. Within the scope of stochastic energy arrival, the case of binary energy arrival has been examined. Different from the scheduling method for deterministic energy arrival, the battery status will be necessary for scheduling. The scheduling method is explained as follows: If there is an energy arrival, the user is directly scheduled. If there is no energy arrival but energy available at the user's battery, the channel status is checked. If the channel is available, the user is scheduled. The user is scheduled for the next round if the channel is unavailable. This approach is aimed to avoid the waste of energy. The algorithm is presented in Algorithm 2.

4.3. Convergence analysis

Convergence analysis for the proposed scheduling method for deterministic energy arrivals and IID data, without AoI and momentum factors, is performed. To show that such a method does not violate the convergence guarantees, a few assumptions must be revisited:

Algorithm 1: Age-involved federated learning with momentum for deterministic energy arrivals.

Require: Total number of global rounds T, number of users K, channel status for user i Q_i, channel error probability of user i q_i, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
Initialize $K_i^t = 0$ for t $\in$ [T]
for Global round $t = 0, ..., T-1$ **do**
  for User i in K **do**
    if $E_i^t = 1$ **then**
      Determine J using (9) and (10)
      Schedule $K_i^{t+J} = 1$
      Determine the momentum attenuation factor using (13)
    end if
    if $K_i^t = 1$ **then**
      if $Q_i^t = 1$ **then**
        for Local iteration m in L **do**
          Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^{(t)})$
        end for
        Send the locally trained model parameters to the parameter server
        $\Delta_i^{t+1} = 1$
      else if $Q_i^t = 0$ **then**
        Schedule $K_i^{t+J} = 0$ and $K_i^{t+J+1} = 1$
        $\Delta_i^{t+1} = \Delta_i^t + 1$
      end if
    end if
  end for
  Parameter Server:
  Update the global model
  Send model parameters $w^{(t+1)}$ to the users
end for

Assumption *(Variance Bound)* The variance of the stochastic gradients from (3) are bounded:

$$E_{\xi_i^{(t)}}[||g_i(w^{(t)}, \xi_i^{(t)}) - \nabla F_i(w^{(t)})||^2] \leq \sigma^2, i \in [K] \tag{16}$$

Assumption *(Second Moment Bound)* The expected square norm of the stochastic gradients from (3) are bounded:

$$E_{\xi_i^{(t)}}[||g_i(w^{(t)}, \xi_i^{(t)})||^2] \leq G^2, i \in [K], \tag{17}$$

where $G > 0$ is a finite real constant.

Assumption *(μ-Strong Convexivity)* The local loss functions of the participating users and the global loss function are μ-strongly convex: For all $\boldsymbol{v}$ and $\boldsymbol{w}$,

$$F_i(\boldsymbol{v}) \geq F_i(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_i(\boldsymbol{w}) + \frac{\mu}{2}||\boldsymbol{v} - \boldsymbol{w}||_2^2 \tag{18}$$

Assumption *(L-Smoothness)* The local loss functions of the participating users and the global loss function are L-smooth: For all $\boldsymbol{v}$ and $\boldsymbol{w}$,

$$F_i(\boldsymbol{v}) \leq F_i(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_i(\boldsymbol{w}) + \frac{L}{2}||\boldsymbol{v} - \boldsymbol{w}||_2^2 \tag{19}$$

Let the scaling coefficient of the local gradients for $J = 0$ be $\chi_i = \frac{1}{P(J=0)}$ and for $0 < J \leq T_i - 1$ be $\varphi_i = \frac{1}{P(0<J\leq T_i-1)}$. Using these parameters, the following lemma can be defined:

Algorithm 2: Age-involved federated learning with momentum for stochastic energy arrivals

Require: Total number of global rounds T, number of users K, channel status for user i Q_i, channel error probability of user i q_i, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
Initialize $K_i^t = 0$ and $B_i^t = 0$ for t $\in$ [T]
for Global round $t = 0, ..., T-1$ **do**
  for User i in K **do**
    if $E_i^t = 1$ **then**
      Schedule $K_i^t = 1$
      Determine the momentum attenuation factor using (13)
      Battery level $B_i^t = 1$
    else
      if $B_i^t = 1$ **then**
        if $Q_i^t = 1$ **then**
          Schedule $K_i^t = 1$
          Determine the momentum attenuation factor using (13)
        else
          Schedule $K_i^{t+1} = 1$ and $K_i^t = 0$
        end if
      end if
    end if
    if $K_i^t = 1$ **then**
      if $Q_i^t = 1$ **then**
        for Local iteration m in L **do**
          Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^{(t)})$
        end for
        Send the locally trained model parameters to the parameter server
        $\Delta_i^{t+1} = 1$
      else if $Q_i^t = 0$ **then**
        Schedule $K_i^t = 0$ and $K_i^{t+1} = 1$
        $\Delta_i^{t+1} = \Delta_i^t + 1$
        Battery level $B_i^{t+1} = 1$
      end if
    end if
  end for
  Parameter Server:
  Update the global model
  Send model parameters $w^{(t+1)}$ to the users
end for

Lemma 1 *(Unbiasedness) For distributed SGD with deterministic energy arrivals,*

$$\mathbb{E}_{S_t}\left[\sum_{i\in S_t} p_i \chi_i g_i(w^{(t)}, \xi_i^{(t)})\right] = \sum_{i=1}^{N} p_i g_i(w^{(t)}, \xi_i^{(t)}) \tag{20}$$

$$\mathbb{E}_{S_t}\left[\sum_{i\in S_t} p_i \varphi_i g_i(w^{(t)}, \xi_i^{(t)})\right] = \sum_{i=1}^{N} p_i g_i(w^{(t)}, \xi_i^{(t)}) \tag{21}$$

for $J = 0$ *and* $0 < J \leq T_i - 1$, *respectively.*

For heterogeneous user set, Lemma 1 analytically shows unbiasedness in a way that even for different participation frequencies of the users, for which the participation depends on the channel and energy conditions, the expected sum of scaled local gradients is equivalent to unbiased local gradient summation, where each user participates equally.

Theorem 1 *For training a machine learning model (1) with deterministic energy arrivals and a learning rate* $\eta \leq min\{\frac{1}{2\mu}, \frac{1}{L}\}$*, the global loss function can be upper bounded as follows:*

$$\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2 \leq (1-\eta\mu)\mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 + \eta^2(\sum_{i=1}^{K} p_i^2(\alpha_{i,max} - 1) + \sum_{i=1}^{K}\sum_{j=1}^{K} p_i p_j)G^2 \tag{22}$$

in T iterations, where $\boldsymbol{w^*}$ *denotes the optimal parameters that minimize the global loss function.*

The proof of this theorem is provided in Appendix S2.

5. Performance evaluation

Experiments were performed as an image classification task with 10 classes of 40 users, for 1000 global rounds and 5 local training rounds, using both the CIFAR-10 [22] and MNIST [23] datasets. The CIFAR-10 dataset was distributed as 50,000 training and 10,000 test samples, with a batch size of 64. For the CIFAR-10 dataset, images were preprocessed before the training to train the model more accurately, including horizontal and vertical flips, color jittering, resizing, and normalization. As the optimizer, SGD is used. The learning rate is set to 0.01. As the architecture, the convolutional neural network (CNN) is used, which includes three 3x3 convolutional layers (with 32, 64, and 64 channels, respectively, the first two with 2x2 pooling layers), a 0.25 dropout layer, a 64-unit fully connected layer, and an output layer. MNIST is an introductory yet useful dataset, which includes handwritten digits from 0 to 9 and has 60,000 samples for training and 10,000 samples for testing, with a batch size of 64. The digits are in normal size and centered in a fixed-size image, and the images are relatively clean and easy to recognize and learn. To show the effect of momentum more clearly, both IID and non-IID training datasets are used in the simulations. IID is applied by shuffling the dataset and splitting it between the clients. Non-IID is applied by sorting the dataset by the digit label, dividing the data into shards, and assigning two shards to each client.

In addition to Algorithm 1 and Algorithm 2 proposed in this paper, the performances of two benchmark policies are demonstrated in the experiments. For the Conservative Algorithm, the learning process takes place only when all users have enough energy and the gradients are not multiplied by a coefficient. As a baseline and reference for many FL algorithms in the literature, *FederatedAveraging* [5] algorithm is simulated. Note that this algorithm represents the performance in perfect conditions, i.e. no energy or channel constraint. To show the effect of nonhomogeneous energy arrivals and channel reliabilities, we divide users into four equal groups and assign different energy arrival parameters and channel error probabilities to each group. In addition, the channel models were generated in a way that the channel error probabilities were randomly assigned, independent of the energy harvesting. The performance evaluation is conducted by calculating the test accuracy for each image class and taking the average.

Figure 2 shows the convergence of Algorithm 1 for deterministic energy arrivals for IID MNIST and CIFAR datasets. Algorithm 1 reaches 100% of accuracy. Similarly, for CIFAR-10, accuracy reaches approximately 70%. It can be observed that the performance of the algorithms with the MNIST dataset significantly outperforms the performance of the algorithms with the CIFAR-10 dataset. This is because images in the MNIST dataset are relatively clean and easy to recognize and learn. Additionally, it can be seen that Algorithm 1, in which the channel state is known, provides high test accuracy. It is important to point out that the reference algorithm, *FederatedAveraging*, does not have any channel, processing power, or time constraints, and still, it provided 75%

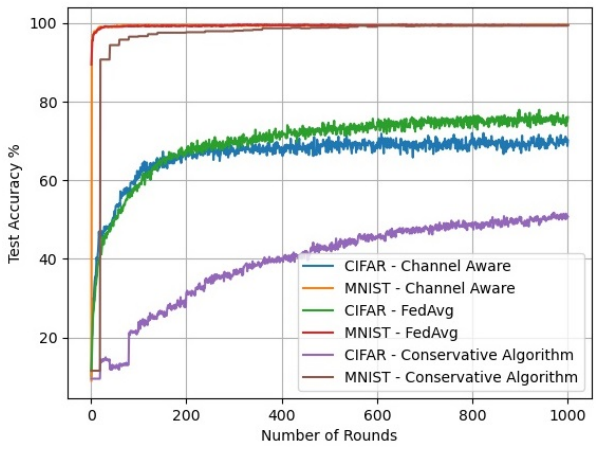

Figure 2. Test accuracy of Algorithm 1 (the method for deterministic energy arrival) for IID MNIST and CIFAR-10 datasets.

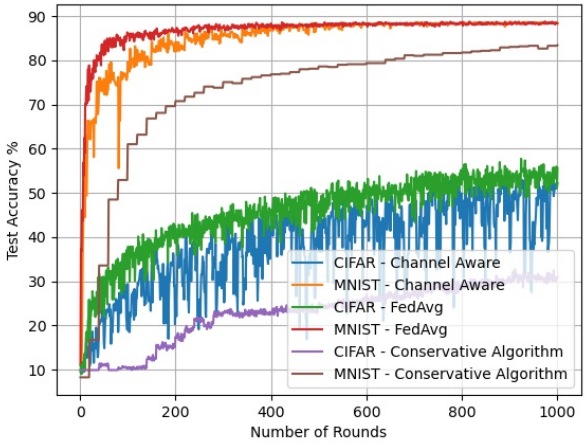

Figure 3. Test accuracy of Algorithm 1 (the method for deterministic energy arrival) for non-IID MNIST and CIFAR-10 datasets.

test accuracy, which is close to the accuracy of the Algorithm 1. Note that the model used in these simulations is basic, and the proposed scheduling methods can be applied to more complicated and high-efficient models. The main goal of these experiments is to show that the proposed algorithms do not violate convergence guarantees and do not achieve higher test accuracy. Figure 3 shows the convergence of Algorithm 1 for deterministic energy arrivals, comparing with *FedAvg*, for non-IID MNIST and CIFAR-10 datasets. For the MNIST dataset, Algorithm 1 reaches approximately 88% of accuracy and for CIFAR-10, accuracy reaches approximately 50%. An observation can be stated as the oscillation of the test accuracy for the non-IID datasets is related to the number and variation of participating users. As an example, the test accuracy of the Conservative Algorithm is much more consistent than the others because all users participate every 20 rounds, so the scheduling is very certain and predictable, resulting in consistent, settled test accuracy. Algorithm 1, Algorithm 2, and *FederatedAveraging* algorithms have oscillation because the scheduling in these algorithms is random. Note that in *FederatedAveraging* algorithm, in every global round, users are randomly selected; and in Algorithm 1 and Algorithm 2, users are scheduled according to the energy arrivals and the value of integer J, which is determined by a probability value that is dependent to a randomly available channel. Compared to the numerical results of the experiments with non-IID data, it can be observed that the numerical results with synthetic datasets are more settled because of the difference in the dataset characteristics. Note that CIFAR-10 is a dataset produced by real-life events, and even the *FederatedAveraging* algorithm without any communicative constraints does not have a settled test accuracy. Similar oscillation can be observed in the numerical results in [24] and [25]. To reduce the oscillation, a possible approach can be eliminating several image classes in the training process.

Figure 4 shows the convergence of Algorithm 2 for stochastic energy arrivals for IID MNIST and CIFAR datasets. Algorithm 2 reaches 100% of accuracy. For CIFAR-10, accuracy reaches approximately 73%, which is very close to the accuracy of *FedAvg*. Figure 5 shows the convergence of Algorithm 2 for stochastic energy arrivals, comparing with *FedAvg*, for non-IID MNIST and CIFAR-10 datasets. For the MNIST dataset, Algorithm 2 reaches approximately 88% accuracy. Additionally, for CIFAR-10, accuracy reaches approximately 53%. It can be seen from the figures that the performance of both proposed algorithms increased compared

to the deterministic arrival case. For Algorithm 1 and Algorithm 2, aiming not to waste substantial energy, the scheduling process includes checking both the channel and battery status when there is no energy arrival. Additionally, there is no J included in the stochastic energy arrival case, because both the scheduling process and scaling parameter provide unbiasedness among users. These factors lead to a better convergence result because the participation of the user is much more guaranteed. Last, but not least, it can be observed from the experimental results that adding AoI as a gradient scaling factor and a metric for momentum attenuation factor has a positive effect on both the test accuracy and convergence rate for non-IID datasets. Additionally, for IID datasets, the model converged significantly faster. As a result, these experimental results verify the claim that AoI-aware momentum improves the model's accuracy for non-IID data and decreases the convergence time for IID data. Note that in the worst case scenario where all of the users participate in the training, the time complexity of both Algorithm 1 and Algorithm 2 is $\mathcal{O}(T*(KL+K)) = \mathcal{O}(TKL)$, where T is the total number of global rounds, K is the total number of users and L is the total number of local training rounds.

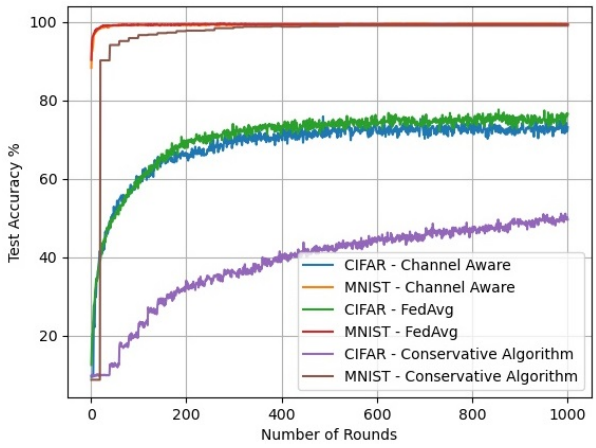

Figure 4. Test accuracy of Algorithm 2 (the method for stochastic energy arrival) for IID MNIST and CIFAR-10 datasets.

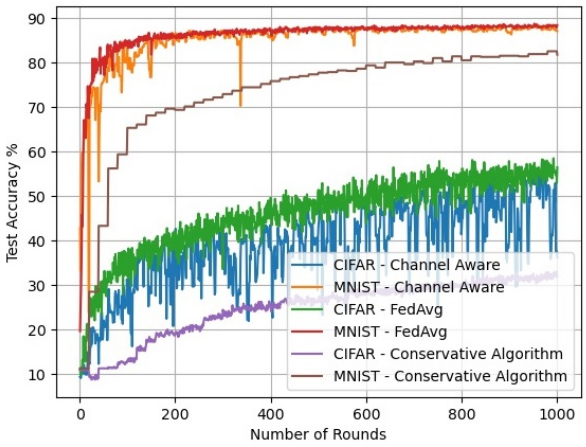

Figure 5. Test accuracy of Algorithm 2 (the method for stochastic energy arrival) for non-IID MNIST and CIFAR-10 datasets.

6. Conclusions and future work

This paper focused on developing FL algorithms that are extended by constraints such as channel availability, energy harvesting, and data freshness, and provide the same guarantee of convergence with the algorithms that have no constraints. A study focused on the effect of AoI with momentum for the scheduling algorithm sensitive to channel and energy state for an FL system prone to energy harvesting and channel errors is presented. As a result, it is shown that with AoI-aware momentum, the accuracy of the model for non-IID data increases, and the convergence time for IID data decreases. In future work, network pruning, defined as the elimination of a specified proportion (referred to as the pruning rate) of weights with the least absolute values until the required model size is achieved, can be studied. Model pruning can be adapted to the constraints such as energy harvesting, data freshness, etc., to achieve more accurate model parameters.

Acknowledgment

This work has been supported in part by BAP Grant AGEP-301-2022-10974.

References

[1] Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT et al. Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492, 2016.

[2] Kaul S, Gruteser M, Rai V, Kenney J. Minimizing age of information in vehicular networks. In: Sensor, Mesh and Ad-Hoc Communications and Networks (SECON), 8th Annual IEEE Communications Society Conference 2011; 350-358.

[3] Kaul S, Yates RD, Gruteser M. Real-time status: How often should one update?. In: IEEE Conference on Computer Communications (INFOCOM) 2012; 2731–2735.

[4] Konečnỳ J, McMahan B, Ramage D. Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575; 2015.

[5] McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. PMLR Artificial intelligence and statistics 2017; 1273-1282.

[6] Li X, Huang K, Yang W, Wang S, Zhang Z. On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189, 2019.

[7] Xu J, Wang S, Wang L, Yao AC. Fedcm: Federated learning with client-level momentum. arXiv preprint arXiv:2106.10874, 2021.

[8] Kim G, Kim J, Han B. Communication-Efficient Federated Learning with Acceleration of Global Momentum. arXiv preprint arXiv:2201.03172, 2022.

[9] Liu W, Chen L, Chen Y, Zhang W. Accelerating federated learning via momentum gradient descent. IEEE Transactions on Parallel and Distributed Systems 2011; 31 (8): 1754-1766.

[10] Shinohara N. Development of Rectenna with Wireless Communication System. In: Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP) 2011; 3970-3973.

[11] Popovic Z. Cut the cord: Low-power far-field wireless powering. IEEE Microwave Magazine 2013; 55-62.

[12] Vyas RJ, Cook BB, Kawahara Y, Tentzeris MM. E-WEHP: A batteryless embedded sensor platform wirelessly powered from ambient digital-tv signal. IEEE Transactions on Microwave Theory and Techniques 2013; 61 (6): 2491-2505.

[13] Güler B, Yener A. Energy-Harvesting Distributed Machine Learning. In: IEEE Int. Symp. on Information Theory (ISIT) 2021; 320-325.

[14] Güler B, Yener A. Sustainable federated learning. arXiv preprint arXiv:2102.11274; 2021.

[15] Gündüz D, Kurka DB, Jankowski M, Amiri MM, Ozfatura E et al. Communicate to learn at the edge. IEEE Communications Magazine 2020; 58 (12): 14-19.

[16] Ozfatura E, Gunduz D, Poor HV. Collaborative learning over wireless networks: An introductory overview. arXiv preprint arXiv:2112.05559 2021.

[17] Çakir Z, Ceran Arslan ET. Federated Learning with Channel and Energy Aware Scheduling. In: IEEE 30th Signal Processing and Communications Applications Conference; Karabük, Turkey 2022; 1-4. doi: 10.1109/SIU55565.2022.9864979

[18] Aygün O, Kazemi M, Gündüz D, Duman TM. Over-the-Air Federated Learning with Energy Harvesting Devices. arXiv preprint arXiv:2205.12869, 2022.

[19] Yang HH, Arafa A, Quek TQS, Poor HV. Age-based scheduling policy for federated learning in mobile edge networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020; 8743-8747.

[20] Buyukates B, Ulukus S. Timely communication in federated learning. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) 2021; 1-6.

[21] Liu X, Qin X, Chen H, Liu Y, Liu B et al. Age-aware Communication Strategy in Federated Learning with Energy Harvesting Devices. In: IEEE/CIC International Conference on Communications in China (ICCC) 2021; 358-363.

[22] Krizhevsky A, Hinton G et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada; 2009.

[23] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE 1998; 86 (11): 2278-2324.

[24] Zhao Y, Li M, Lai L, Suda N, Civin D et al. Federated learning with non-iid data. arXiv preprint arXiv:1806.00582, 2018.

[25] Wang H, Kaplan Z, Niu D, Li B. Optimizing federated learning on non-iid data with reinforcement learning. In: IEEE Conference on Computer Communications (INFOCOM) 2020; 1698-1707.

Appendix

S1. Derivation of the probability of the scheduling parameter

Let $\alpha_t = P_J(j)$ and the channel error probability of user i is q_i. To ensure fairness among all participant users, it is assumed that $\alpha_0 = \alpha_1 = \alpha_2 = ... = \alpha_{T_i-1}$. The probabilities of different J values can be defined as follows:

$$\alpha_t(0) = (1-q_i)P_J(0)$$

$$\alpha_t(1) = (1-q_i)P_J(1) + q_i(1-q_i)P_J(0)$$

$$...$$

$$\alpha_t(T_i-1) = (1-q_i)P_J(T_i-2) + q_i(1-q_i)P_J(T_i-3) + ... + q_i^{(T_i-2)}(1-q_i)P_J(0)$$

Because of the assumption:

$(1-q_i)P_J(0) = (1-q_i)P_J(1) + q_i(1-q_i)P_J(0)$, implies that $(1-q_i)P_J(0) = P_J(1)$,

$(1-q_i)P_J(1) + q_i(1-q_i)P_J(0) = (1-q_i)P_J(2) + q_i(1-q_i)P_J(1) + q_i^2(1-q_i)P_J(0)$, implies that $P_J(2) = P_J(1)$,

$P_J(3) = (1-q_i)P_J(2) + q_i(1-q_i)P_J(1) + q_i^2(1-q_i)P_J(0) = P_J(2) = P_J(1)$, implies that $P_J(T_i-1) = ... = P_J(2) = P_J(1)$.

It is known that $\sum_0^{T_i-1} P_J(j) = 1$. This leads to:

$$P_J(0) = \frac{1}{T_i - T_i q_i + q_i}, \text{ and } P(0 < J \leq T_i - 1) = \frac{1-q_i}{T_i - T_i q_i + q_i}$$

This completes the derivation.

S2. Proof of the Theorem 1

By letting $g_i^t \triangleq g_i(w^{(t)}, \xi_t)$, $\boldsymbol{w^*} \triangleq argmin_{\boldsymbol{w}} F(\boldsymbol{w})$, $\xi_t = (\xi_1^{(t)}, \xi_2^{(t)}, ..., \xi_K^{(t)})$, from (15), and $\alpha_i^t = \chi_i$ for J = 0 and $\alpha_i^t = \varphi_i$ otherwise, we find that:

$$\begin{aligned} \mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2] &= \mathbb{E}_{S_t,\xi_t}[||w^{(t)} - \eta \sum_{i\in S_t} p_i(\alpha_i^t g_i(w^{(t)}, \xi_i^{(t)})) - w^*||^2] \\ = \mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 &- 2\eta \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i\in S_t} p_i(\alpha_i^t g_i^t)\rangle] + \eta^2 \mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i(\alpha_i^t g_i^t)||^2]] \end{aligned} \tag{23}$$

The second term in (23) can be expanded as in the following:

$$\begin{aligned} \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i\in S_t} p_i(\alpha_i^t g_i^t)\rangle] &= \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i\in S_t} p_i(\alpha_i^t g_i^t) - \sum_{i=1}^K p_i \nabla F_i(w^{(t)}) + \sum_{i=1}^K p_i \nabla F_i(w^{(t)})\rangle] \\ = \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i\in S_t} p_i(\alpha_i^t g_i^t) - \sum_{i=1}^K p_i \nabla F_i(w^{(t)})\rangle] &+ \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i=1}^K p_i \nabla F_i(w^{(t)})\rangle] \end{aligned} \tag{24}$$

Because of Lemma 1, the first term in (24) vanishes, and by using Assumption 4.3 and (5), we obtain:

$$\mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i=1}^K p_i \nabla F_i(w^{(t)})\rangle] = \langle w^{(t)} - w^*, \nabla F(w^{(t)})\rangle \geq F(w^{(t)}) - F(w^*) + \frac{\mu}{2}||w^* - w^{(t)}||^2 \tag{25}$$

The third term in (23) can be expanded as in the following:

$$\begin{aligned}\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t||^2]] &= \mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t + \sum_{i=1}^{K} p_i g_i^t||^2]] \\ &= \mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] - 2\mathbb{E}_{S_t,\xi_t}[\langle\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t, \sum_{i=1}^{K} p_i g_i^t\rangle] + \sum_{i=1}^{K}||p_i g_i^t||^2\end{aligned} \tag{26}$$

The second term in (26) vanishes, and the equation becomes:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t||^2]] = \mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] + \mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K}||p_i g_i^t||^2] \tag{27}$$

By combining (23) and (27), and using Assumption 4.3, we get that:

$$\begin{aligned}\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2] &= \mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 - 2\eta\langle w^{(t)} - w^*, \nabla F(w^{(t)})\rangle \\ &\quad + \eta^2(\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] + \mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K}||p_i g_i^t||^2])\end{aligned} \tag{28}$$

$$\leq (1-\eta\mu)\mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2] - 2\eta(F(w^{(t)}) - F(w^*)) + \eta^2\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] + \eta^2\mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K}||p_i g_i^t||^2] \tag{29}$$

Let $U_i^t = \begin{cases} 1, & \text{if the user participates at time } t \\ 0, & \text{otherwise} \end{cases}$ and $P(U_i^t = 1) = \alpha_i$. Under this definition:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2] = \mathbb{E}_{U_t,\xi_t}[||\sum_{i=1}^{K} p_i(\alpha_i^t g_i^t - g_i^t)||^2] \tag{30}$$

$$= \sum_{i=1}^{K} p_i^2\mathbb{E}_{U_t,\xi_t}[||\alpha_i^t g_i^t - g_i^t||^2] + \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\mathbb{E}_{U_t,\xi_t}[\langle p_i(\alpha_i^t g_i^t - g_i^t), p_j(\alpha_j^t g_j^t - g_j^t)\rangle] \tag{31}$$

Because of independence, the second term in (31) vanishes:

$$\sum_{i=1}^{K} p_i^2\mathbb{E}_{U_t,\xi_t}[||\alpha_i^t g_i^t - g_i^t||^2] = \sum_{i=1}^{K} p_i^2(U_i^2)\mathbb{E}_{\xi_t}[\mathbb{E}_{U_t|\xi_t}[(U_i^t - \frac{1}{\alpha_i^t})^2 - ||g_i^t||^2|\xi_t] \tag{32}$$

By using Assumption 4.3, it can be stated that:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2] \leq \sum_{i=1}^{K} p_i^2(\alpha_{i,max} - 1)G^2 \tag{33}$$

From Cauchy-Schwarz inequality, the last term in (29) can be expressed as in the following:

$$\eta^2\mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K}||p_i g_i^t||^2] \leq \sum_{i=1}^{K} p_i^2\mathbb{E}_{\xi_t}[||g_i^t||^2] + \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K} p_i p_j\mathbb{E}_{\xi_t}[||g_i^t||||g_j^t||] \tag{34}$$

$$\leq \sum_{i=1}^{K} p_i^2 \mathbb{E}_{\xi_t}[||g_i^t||^2] + \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \frac{p_i p_j}{2} \mathbb{E}_{\xi_t}[||g_i^t||^2 + ||g_j^t||^2] \tag{35}$$

$$\leq \sum_{i=1}^{K} \sum_{j=1}^{K} p_i p_j G^2 \tag{36}$$

Equation (35) holds by using AM-GM Inequality, and Equation (36) is stated by using Assumption 4.3, where $G > 0$ is a finite real constant. Finally, by combining (33) and (36) and noting that $-2\eta(F(w^{(t)}) - F(w^*)) \leq 0$, it can be stated that:

$$\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2 \leq (1 - \eta\mu)\mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 + \eta^2(\sum_{i=1}^{K} p_i^2(\alpha_{i,max} - 1) + \sum_{i=1}^{K}\sum_{j=1}^{K} p_i p_j)G^2 \tag{37}$$

This completes the proof.