J. Liu, J. Zheng, J. Zhang, L. Xiang, D. W. K. Ng and X. Ge, "Delay-Aware Online Resource Allocation for Buffer-Aided Synchronous Federated Learning Over Wireless Networks," in *IEEE Access*, vol. 12, pp. 164862-164877, 2024, DoI: 10.1109/ACCESS.2024.3489657.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier ...

Delay-aware Online Resource Allocation for Buffer-aided Synchronous Federated Learning over Wireless Networks

JING LIU[1], JINKE ZHENG[1], Jing Zhang[1] (Member, IEEE), Lin Xiang[2] (Member, IEEE), Derrick Wing Kwan Ng[3] (Fellow, IEEE), and Xiaohu Ge[1] (Senior Member, IEEE).

[1]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: jingliu0909@hust.edu.cn; jinkezheng@hust.edu.cn; zhangjing@hust.edu.cn; xhge@hust.edu.cn)

[2]Communications Engineering Lab, Technische Universität Darmstadt, Germany (e-mail: l.xiang@nt.tu-darmstadt.de)

[3]School of Electrical Engineering, and Telecommunications, University of New South Wales, Sydney 2033, Australia (e-mail: w.k.ng@unsw.edu.au)

Corresponding author: Jing Zhang (e-mail: zhangjing@hust.edu.cn).

This work is supported by the National Natural Science Foundation of China under Grant U2001210, Hubei Provincial Science and Technology Department under Grant 2021BAA009. The work of L. Xiang has been funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center under grant LOEWE/1/12/519/03/ 05.001(0016)/72 and the BMBF project Open6GHub under grant 16KISKO14. This paper has been presented in part at the IEEE ICCC 2023 [1].

ABSTRACT Synchronous federated learning (FL) over wireless networks often suffers from the straggler effect, when the time required for training local models and uploading trained parameters varies significantly across heterogeneous wireless devices. This disparity prolongs the duration needed for model aggregation at the data center and slows down the convergence of synchronous FL, posing a significant challenge for FL over wireless networks. In this paper, we propose a novel buffer-aided FL scheme to mitigate the straggler effect. A buffer with sufficiently large storage is deployed at each wireless device to temporarily store the collected training data and adaptively outputs it during local training, according to the computational capabilities and communication data rates of the wireless devices. Consequently, all local models can be synchronously aggregated at the data center to reduce the number of rounds required for model aggregation in FL. To ensure timely information update, a staleness function is further introduced to characterize the freshness of the data used to train local models. Additionally, the entropic value-at-risk (EVaR) of the data queues is introduced to eliminate the impact of discarded data at the buffers and improve the accuracy of trained local models. We formulate a delay-aware online stochastic optimization problem to minimize the long-term average staleness of all wireless devices for buffer-aided FL. Our problem formulation simultaneously guarantees stability of data queues at the wireless devices and reduces the risk of data loss. By employing the Lyapunov optimization technique, we transform the problem into instantaneous deterministic optimization subproblems and further solve each subproblem online via utilizing its hidden convexity. Simulation results demonstrate that the proposed buffer-aided synchronous FL scheme can effectively improve the convergence rate of FL and, at the same time, ensure timely synchronization of heterogeneous wireless devices.

INDEX TERMS Federated learning, straggler effect, delay, Lyapunov optimization.

I. INTRODUCTION

FEDERATED learning (FL) has emerged as a promising distributed learning method [2]. Unlike traditional machine learning (ML) approaches, FL enables multiple wireless devices to collaboratively train a global model, together with a centralized server, while maintaining the privacy of individual data sources. Data privacy is preserved by exchanging only model parameters, rather than raw data, between wireless nodes and the server. Additionally, the bandwidth required for data transmission is significantly reduced, as the size of model parameters is much smaller than that of the local raw data. As a result, FL can greatly benefit from the massive data generated by mobile devices to enable a wide range of personalized and context-aware services, such as learning the activities of smartphone users and predicting health events via wearable devices in distributed wireless Internal-of-Things (IoT) systems [3], [4].

However, despite the advantages of FL, effectively deploying FL over wireless networks is hindered by several unsolved issues. On the one hand, wireless devices often have distinct computing capabilities and communication capacities. As such, synchronizing FL tasks among these devices, including distributing the global model to and aggregating local updates from the devices, would suffer from the straggler effect. That is, the completion time of synchronous FL jobs is limited by the slowest wireless device, defining a severe performance bottleneck in practical heterogeneous wireless networks and mobile environments. On the other hand, due to the limited communication capacity and computation power of wireless nodes, local model training may suffer from severe latency and link failures, causing significant delays in transmission of trained local models. This in turn fundamentally limit the performance of FL over wireless networks.

To address both challenges, many works have investigated novel FL methods combined with edge computing resource allocation. Here we review the existing methods in two categories, namely asynchronous/semi-asynchronous FL and buffer-aided FL.

A. ASYNCHRONOUS/ SEMI-ASYNCHRONOUS FL

Asynchronous and semi-asynchronous FL typically select only a fraction of participating devices in each synchronization round to upload their local models for aggregation [5]. For instance, the semi-asynchronous FL framework FeDiSa, introduced in [6] addressed cyberattack discrimination while considering the impact of communication latency and stragglers. In [7], a joint communication resource allocation and client selection scheme was proposed for asynchronous FL. This selection process leveraged the convergence condition of the learning model, allowing each selected client to upload its model parameters to the server immediately after training, without waiting for other clients. To minimize training latency caused by the straggler effect, [8] formulated a multi-armed bandit problem for client selection problem and solved in an online manner. To reduce the bias and variance of periodic model aggregation in FL systems, a channel-aware data importance-based scheduling policy was proposed in [9]. In [10], the authors proposed a triple-step asynchronous FL mechanism, TrisaFed, which efficiently and securely trains the intelligent core of IoT by activating clients with rich information, optimizing communication costs, and enhancing model aggregation using temporal and informative attributes. Due to only part of edge devices take part in synchronization in each round, asynchronous and semi-asynchronous FL easily achieve node synchronization, which relieve the straggler effect of FL. However, for all wireless edge nodes, only a subset of all locally trained models is used to aggregate the global model in each round and thus, the convergence rate of asynchronous and semi-asynchronous FL may be slower than traditional FL.

B. BUFFER-AIDED FL

While asynchronous and semi-asynchronous FL aim to reduce the number of edge nodes required for global model aggregation, another approach employs buffer in nodes and servers to mitigate the straggler effect in FL. For example, federated reinforcement learning (FRL) has been proposed to reduce the time consumption for training local model by effectively adjusting the content stored in the buffer. By utilizing the reward scheme of reinforcement learning, edge nodes and serve can enhance the similarity between input and stored data, thereby accelerating the timeliness of all edge nodes. In [11], a device assignment algorithm based on a dueling double deep Q-Network (D^3QN) is introduced, which uses a replay buffer to store network weight values. This method balances workloads across edge servers to minimize latency. Additionally, buffers can be deployed at nodes to store crucial information during data collection. To manage large amounts of unlabeled data collected at nodes with limited storage, the authors of [12] introduced a self-supervised on-device FL framework that automatically selects the most representative samples for storage in the replay buffer on each device. To accelerate learning in cloud-edge-terminal Internet of Things (IoT) networks, [13] proposed a collaborative policy learning framework using an edge-agnostic policy structure to aggregate local policies from different edge nodes. In [14], a three-layer end-edge-cloud FRL framework was designed to optimize client node selection and global aggregation frequency in a digital twin (DT) system.

In general, FRL can address the straggler effect over multiple rounds when rewards are appropriately designed to minimize model training delay. However, the effectiveness of FRL in overcoming this effect heavily depends on the similarity between input and stored data. If the collected data differs significantly from the stored content in unknown environment, the reward scheme of FRL may lead to slow convergence.

Unlike FRL, which relies on stored content to mitigate the straggler effect, buffer-aided synchronous FL solves the problem by adjusting the amount of data used for local model training. Data stored in the buffer can be output with amounts adjusted according to available resource at the nodes before being used for training. Therefore, the buffer-aided synchronous FL provides a novel approach to effectively solve the straggler effect of FL. However, implementing the buffering mechanism in IoT devices to minimize latency in FL tasks over wireless networks still faces two main challenges. First, a replay buffer configured in IoT nodes can introduce queueing delays, potentially prolonging the time required for training local FL models and uploading trained models, compared with the traditional FL. Another challenge with using a replay buffer is the increased complexity of resource allocation needed to ensure simultaneous model uploads from all nodes, compared to traditional FL. While the replay buffer allows for dynamic adjustment of the data used for local model training based on each node's computation and communication capabilities, the flexibility can lead to a mismatch between the amount of collected data and the data used for training. Such a mismatch can result in the discarding of raw data that could have been useful for training, potentially degrading the accuracy of trained local models.

To address these challenges, in this paper, we propose a novel mechanism that integrates buffer-aided model updating for FL in IoT applications. The collected data is stored in a replay buffer before being used to train the local model on each device for FL. The replay buffer enables to dynamically adjust the amount of data used for local model training based on each node's computation and communication capabilities. Therefore, all nodes can upload their trained local models to the data server simultaneously, preventing any single node from delaying the overall process. However, when the amount of data required for training the local model is less than the amount of raw data, part of the raw data will be discarded. To minimize the delay of data stored in the queue, the first in first-output (FIFO) policy is adopted. We introduce a staleness function to evaluate the timeliness of model updates, which characterizes the delay from data collection to transfer of trained models to the server. By minimizing the average staleness of all wireless devices in each round, we aim to reduce the overall latency of FL tasks.

To ensure the accuracy of local model training using collected data, we utilize the entropic value-at-risk (EVaR) to assess the risk of discarded data on FL. The EVaR also ensures stability in data queues, lowering the likelihood of significant queue lengths and data loss. By limiting the EVaR of discarded data, the accuracy of local models can be improved, which can reduce the require training rounds. We further formulate an optimization problem to evaluate the tradeoff between the latency for training local FL models and the required rounds for FL tasks, while considering both the risk of discarded data. An effective algorithm is developed to allocate the resources of buffer-aided edge nodes. Simulation results show that our proposed algorithm can reduce the completion latency of FL over the distributed IoT networks equipped with buffers. The contributions of this work are summarized as follows.

- We propose a delay-aware buffer-aided FL scheme, which utilizes a staleness function to evaluate the timeliness of local updates and alleviate model inconsistencies among wireless devices.
- To minimize the average delay for the FL system while guaranteeing the EVaR of the data queue, we design a resource allocation and wireless device scheduling policy for training local model in each round to mitigate the risk of data loss for local FL model training.
- Our experiments demonstrate that buffer-aided synchronous FL scheme can improve the convergence rate and reduce the completion time required by FL tasks, while also ensuring the freshness of training data.

In the remainder of this paper, the system model is defined in Section II. We formulate the delay minimization problem and propose an iterative resource allocation algorithm based on Lyapunov optimization technique in Sections III and IV, respectively. We present numerical and simulation results, along with a discussion, in Section V, followed by the conclusion in Section VI.

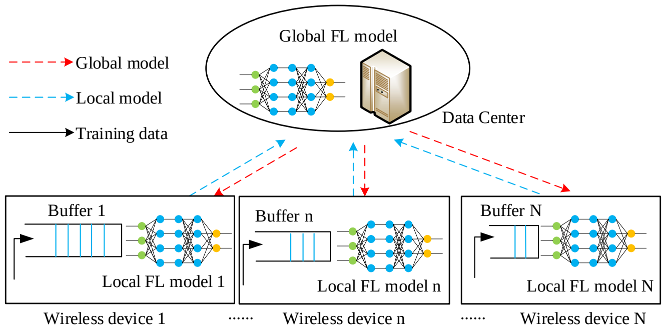


Figure 1. Illustration of a buffer-aided FL system with N wireless devices.

II. SYSTEM MODEL

A. QUEUE MODEL OF BUFFER-AIDED WIRELESS DEVICES

As illustrated in Fig. 1, we consider an FL system consisting of a data center and N IoT wireless devices that are interconnected over wireless links. The wireless devices may include smartphones, sensors, and smart meters in a smart home, which are responsible for e.g. real-time monitoring of the home environment and collecting data for collaboratively implementing ML with the data center. To guarantee privacy during data collection and processing in ML, *local* FL models are deployed at the wireless devices [15]. Each wireless device needs to train a local FL model adopting its collected data, neither sharing the data with the data center nor other wireless devices. Subsequently, the locally trained FL models are wirelessly sent to the data center, where they are aggregated into a *global* FL model. The data center then broadcasts the renewed global FL model to the wireless devices to enable another round of local training.

We consider *synchronous* FL, which has been widely adopted thanks to its fast average convergence performance of wireless devices [16]. To overcome the straggler effect of synchronous FL, we deploy a buffer at each wireless device. Exploiting the buffer, the wireless devices can store their collected data and dynamically adjust the amounts of data output from their buffers for local training, in adaption to the local computing capacity and communication resources. This enables to accelerate the local model aggregation and communication and ultimately reduce the required rounds for FL convergence. We refer to the resulting scheme as *buffer-aided FL*. We note that buffering has also been widely applied in communication applications, such as buffer-aided relaying [17], [18], to overcome channel fading while at the cost of increased communication latency. In contrast, in our paper, buffer-aided FL aims to overcome the straggler effect for synchronous FL systems by aligning the local training and communication processes at the wireless devices.

The system time is divided into multiple communication rounds indexed by set $\mathcal{I} = \{1, 2, \cdots, i, \cdots\}$. Data inputs into and outputs from the buffer at each wireless device following a FIFO policy [19]–[21]. In communication round i, wireless device n, $n \in \{1, ..., N\}$, collects a local dataset with $A_n(i)$ bytes of data samples from the environment.

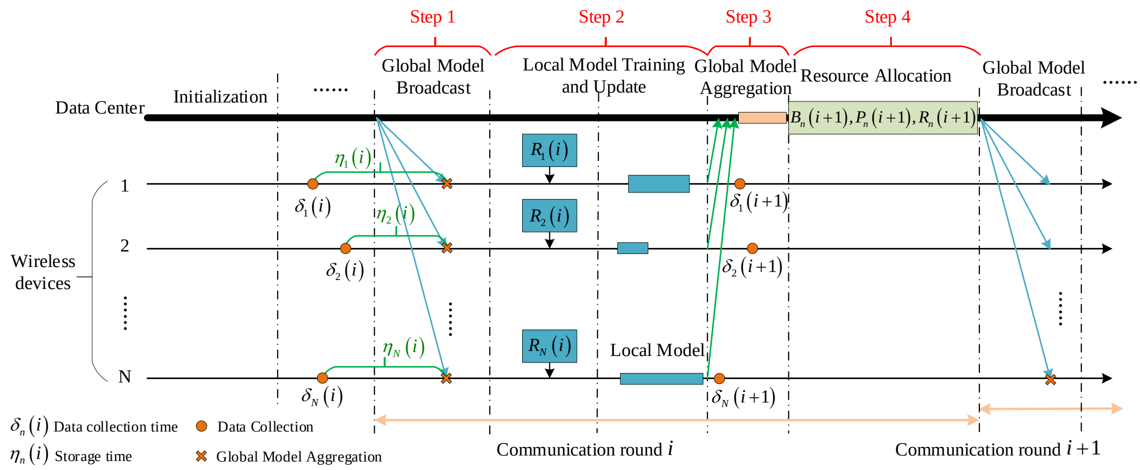


Figure 2. Illustration of a buffer-aided FL system with N wireless devices.

For convenience, we assume throughout this paper that each buffer has a sufficiently large size, such that the data collected from environment in each communication round can be stored without overflow; otherwise, for buffers with limited size, this assumption can be lifted by treating $A_n(i)$ as the effective volume of data that can be stored into the buffer at each wireless device. Moreover, we define $R_n(i)$ as the amount of data in bytes output from the buffer and adopted by wireless device n to train its local FL model in communication round i. If $A_n(i) > R_n(i)$, the remaining $D_n(i) = A_n(i) - R_n(i)$ amount of unused data will be discarded. We note that in reinforcement learning with experience replay, data can be stored in a finite replay buffer and exploited for future training [22], [23]. However, as outdated data and/or model may degrade the performance of FL, the unused training data in FL is usually instantly removed at the local computing nodes [24].

B. DATA PROCESSING FOR BUFFER-AIDED FL

As shown in Fig. 2, in communication round $i \in \mathcal{I}$, the following steps are executed by the buffer-aided FL:

Step 1 (Global model broadcast): The data center broadcasts the current global model parameter, denoted by $\mathbf{w}^*(i-1)$, to the N wireless devices.

Step 2 (Local model training and update): Wireless device n executes I_n steps of stochastic gradient descent (SGD) to train its local model, by adopting the received global model parameter $\mathbf{w}^*(i-1)$ and outputing $R_n(i)$ bytes of buffered data. The local training at wireless device n aims to minimize the loss function defined as $f_n(\mathbf{w} \mid \mathbf{w}^*(i-1), R_n(i))$, i.e., to solve the following optimization problem

$$\mathbf{w}_n^*(i) := \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} f_n\left(\mathbf{w}_n \mid \mathbf{w}^*(i-1), R_n(i)\right), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is an n-dimensional real vector parameter to be learned. The local model $\mathbf{w}_n^*(i)$ is then uploaded to the data center once it is obtained.

Step 3 (Global model aggregation): The data center aggregates all local models uploaded from all wireless devices to update the global model. As such, the global FL aims to minimize the overall loss function defined as

$$F(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} f_n\left(\mathbf{w} \mid \mathbf{w}_n(i), R_n(i)\right). \tag{2}$$

The optimal model obtained in communication round i is denoted as

$$\mathbf{w}^*(i) := \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} F(\mathbf{w}). \tag{3}$$

Step 4 (Resource allocation): The data center allocates bandwidth and power to all wireless devices and schedules training data to be output from each buffer for the next communication round.

Steps 1-4 are repeated in each communication round until convergence or termination conditions are satisfied. To ensure convergence of the global model, the loss function $f(\mathbf{w})$ should satisfy the following assumptions [25]:

- *Assumption 1 (Smoothness)*: $f(\mathbf{w})$ is L-smooth with $L > 0$, i.e., $\forall \mathbf{w}_1, \mathbf{w}_2, f(\mathbf{w}_2) - f(\mathbf{w}_1) \leq \langle \nabla f(\mathbf{w}_1), \mathbf{w}_2 - \mathbf{w}_1 \rangle + \frac{L}{2}\|\mathbf{w}_2 - \mathbf{w}_1\|^2$.
- *Assumption 2 (Strong Convexity)*: $f(\mathbf{w})$ is μ-strongly convex with $\mu \geq 0$, i.e., $\forall \mathbf{w}_1, \mathbf{w}_2, f(\mathbf{w}_2) - f(\mathbf{w}_1) \geq \langle \nabla f(\mathbf{w}_1), \mathbf{w}_2 - \mathbf{w}_1 \rangle + \frac{\mu}{2}\|\mathbf{w}_2 - \mathbf{w}_1\|^2$.

As illustrated in Fig. 2, let $\delta_n(i)$ denote the time instant of collecting data samples periodically at wireless device n from the environment in the i-th communication round. The collected data is buffered for $\eta_n(i)$ amount of time, till wireless device n starts the local FL training at time instant $\delta_n(i) + \eta_n(i)$. The queueing time $\eta_n(i)$, also referred to as the elapsed delay in this paper, can be caused by various factors, e.g. the ongoing data processing and model transmission in the previous communication round, cf. Fig. 3.

To exploit buffers for accelerated FL in the considered wireless networks, scheduling of data output from the buffer

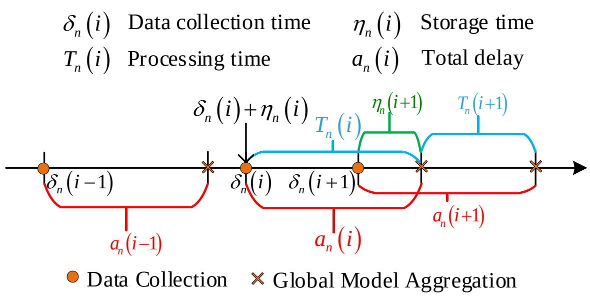


Figure 3. Illustration of time instances of data sampling and transmission.

for local training is crucial. In particular, by knowing the computing and communication times required for local FL and model uploading at all wireless devices, we can optimize the amount of data used for local model training at each wireless device such that the trained local models of all devices can be received at the data center in a timely synchronized manner, to significantly accelerate the global learning process.

C. DELAY FOR TRAINING LOCAL MODELS

Let $a_n(i)$ be the delay of collected data at wireles device n in the i-th communication round, which includes the time elapsed since the data is collected at the wireless device until the local FL training completes and the trained model is sent to the data center. Moreover, let $T_n(i)$ be the duration for computing and transmitting the local model at wireless device n. Consequently, the delay of completing local model training at wireless device n in the i-th round is given as

$$a_n(i) = \eta_n(i) + T_n(i). \quad (4)$$

Based on the definition of $\eta_n(i)$, we have

$$\eta_n(i) = [\delta_n(i-1) + a_n(i-1) - \delta_n(i)]^+, \quad (5)$$

with $[x]^+ = \max\{x, 0\}$. Note that if $\delta_n(i-1) + a_n(i-1) \leq \delta_n(i)$, i.e., the $(i-1)$-th communication round completes before collecting data from the environment in the i-th communication round, we have $\eta_n(i) = 0$. Otherwise, a positive $\eta_n(i) > 0$ is incurred. For example, we have $\eta_n(i) = 0$ and $\eta_n(i+1) > 0$ in Fig. 3. By substituting $\eta_n(i)$ in (5) into (4), we can rewrite the delay in a recursive manner as

$$a_n(i) = [\delta_n(i-1) + a_n(i-1) - \delta_n(i)]^+ + T_n(i). \quad (6)$$

In (6), it remains to analyze $T_n(i)$, which includes the required computation time $\tau_n(i)$ and transmission time $t_n(i)$. Without loss of generality, let $\varphi_n(i)$ be the frequencies of the central processing unit (CPU) at wireless devices n for processing per unit of data used in the local training tasks. The required computation time $\tau_n(i)$ at wireless device n for processing data $R_n(i)$ in communication round i is given by [26]

$$\tau_n(i) = \frac{I_n C_n R_n(i)}{\varphi_n(i)}, \quad (7)$$

where C_n is the number of CPU cycles required for processing each byte of input data sample per iteration. Meanwhile, according to the analysis in [25, Lemma 1], the number of iterations I_n required for training the local model $\mathbf{w}_n$ in wireless device n using the SGD method satisfies

$$I_n \geq v\log_2(1/\rho), \quad (8)$$

where $v \geq 0$ is a positive constant. The value of v depends on the data size and the local model training task [27], and is related to the assumptions of the adopted loss function in Sec. II-B. Moreover, $\rho > 0$ is the required accuracy.

We assume that the trained model parameter $\mathbf{w}_n(i)$ at device n has a fixed size of d_n, whose value is specified a priori according to the adopted local neural network architecture. Moreover, the multiple nodes upload their local models simultaneously adopting orthogonal frequency-division multiple access (OFDMA) [28]–[30]. Let $P_n(i)$ and $B_n(i)$ be the transmit power and allocated bandwidth for wireless device n in communication round i, respectively. The time required for transmitting the locally trained FL model $\mathbf{w}_n(i)$, can be calculated as [31]

$$t_n(i) = \frac{d_n}{B_n(i)\log_2\left(1 + \frac{h_n(i)P_n(i)}{N_0 B_n(i)}\right)}, \quad (9)$$

where $h_n(i)$ is the channel gain between wireless device n and the data center. We consider a block fading channel such that $h_n(i)$ keeps unchanged during each communication round and is known via existing channel estimation techniques [32]. But the channel condition $h_n(i)$ can change among different communication rounds. N_0 is the power spectral density of the received additive white Gaussian noise.

According to (7) and (9), the transmission time $T_n(i)$ of wireless device n in communication round i can be calculated as

$$T_n(i) = \frac{I_n C_n R_n(i)}{\varphi_n(i)} + \frac{d_n}{B_n(i)\log_2\left(1 + \frac{h_n(i)P_n(i)}{N_0 B_n(i)}\right)}. \quad (10)$$

Substituting (8) into (10), we can establish a lower bound on the required transmission time as

$$T_n(i) \geq \frac{vC_n R_n(i)\log_2(1/\rho)}{\varphi_n(i)} + \frac{d_n}{B_n(i)\log_2\left(1+\frac{h_n(i)P_n(i)}{N_0 B_n(i)}\right)}. \quad (11)$$

D. EVAR ASSOCIATED WITH FINITE BUFFER AND DATA DISCARDING

When the amount of collected data exceeds that used in local model training, the collected but unused data will be discarded, introducing risk of data loss. In this subsection, we consider the EVaR metric to quantify and mitigate the potential risk of data discarding at the buffers [33].

To apply the EVaR, let us first consider the value-at-risk (VaR), which is a well-known risk measure in finance [34]. We denote $Q_n \triangleq \frac{1}{i}\sum_{j=1}^{i} D_n(j)$ as the time average of discarded data from the beginning to the i-th communication round. Let $F_{Q_n}(\cdot)$ be a well-defined cumulative distribution function (CDF) of Q_n. The pre-determined quantity $1-\Gamma$ denotes the confidence level of this potential risk about discarded data and Γ is the risk/tail level. For given $\Gamma \in (0,1]$, the VaR quantifies the expected value of queue length in the Γ-tail of $F_{Q_n}(\cdot)$, i.e.,

$$\mathrm{VaR}_{1-\Gamma}(Q_n) = \min_{a \in \mathbb{R}} \{a : F_{Q_n}(a) \geq 1-\Gamma\}, \forall n \in \mathcal{N}, \quad (12)$$

where $\mathbb{R}$ is the field of real numbers. By maintaining the VaR below a specific threshold, the probability of data losses is constrained by Γ to ensure reliable data transmission.

However, the VaR lacks subadditivity and is not a coherent measure, for which it fails to perfectly capture the notion of risk [35]. As an alternative, the notion of EVaR has been proposed as an effective risk measurement for stochastic optimization problem due to its tractability. As such, we consider EVaR in the sequel. Specifically, EVaR resolves risk measurement by characterizing the best upper bound of VaR obtained with the Chernoff inequality [36], where $\mathrm{VaR}_{1-\Gamma}(Q_n) \leq \mathrm{EVaR}_{1-\Gamma}(Q_n), \forall n \in \mathcal{N}$. Thus, the EVaR is formally defined as

$$\mathrm{EVaR}_{1-\Gamma}(Q_n) = \min_{z>0} \left\{ z \ln \left[\frac{1}{\Gamma} \mathbb{E}_{Q_n} \left[\exp\left(\frac{Q_n}{z} \right) \right] \right] \right\}, \forall n. \quad (13)$$

Note that the expression in (13) is the perspective function of the log-sum-exp function of Q_n, which is known to be jointly convex with respect to Q_n and z. As a result, the EVaR, being the minimum of a convex function over z, is also a convex function of Q_n. In order to mitigate data loss and guarantee reliable data transmission, the following constraint should be satisfied

$$\mathrm{EVaR}_{1-\Gamma}(Q_n) \leq \epsilon, \forall n, \quad (14)$$

where ϵ is the threshold of EVaR. We will show in Sec. IV-A that the EVaR constraint in (14) can guarantee the stability of data queues at the wireless devices (but the converse may not be true) [37].

III. PROBLEM FORMULATION

A. DEFINITION OF STALENESS FUNCTION

In this section, we define a staleness function to characterize the freshness of collected/buffered data on any device during synchronization of all wireless devices. In particular, the staleness function, denoted by $g[a_n(i)]$, represents the level of dissatisfaction about data staleness characterized by the delay $a_n(i)$ of each wireless device or the need for new information updates [35]. We assume that $g[a_n(i)]$ fulfills the following properties.

- *Assumption 3:* The function $g[a_n(i)]$ is non-negative and non-decreasing [38], implying that stale data is usually less desired than fresh data [39], [40]. Also, the staleness function is monotonic with respect to delay for improving the freshness of collected data.
- *Assumption 4:* The function $g[a_n(i)]$ is convex with respect to $a_n(i)$, although the delay of collected data $a_n(i)$ can be a nonconvex function of the bandwidth allocation.

In this paper, we define the staleness function for each wireless nodes in each round as the fairness utility function

$$g[(a_n(i))] = \frac{a_n^{(1-\beta)}(i)}{1-\beta}, \forall n, i, \quad (15)$$

for $\beta \leq 0$ which is a non-negative and convex function of $a_n(i)$. Here, it can verified that (15) satisfy both *Assumption 3* and *4*. The adoption of a nonlinear convex staleness function can better capture the trend of delay variations, whereas conventional delay shows a simple linear growth trend that may be inaccurate in representing these variations. Moreover, by adjusting β, one can effectively tradeoff between efficiency and fairness for different wireless devices in (15). For example, maximum efficiency is achieved by setting $\beta = 0$ whereas proportional and max-min fairness are achieved by setting $\beta = 1$ and $\beta \to \infty$, respectively.

B. LONG-TERM STALENESS FUNCTION MINIMIZATION PROBLEM

To overcome the straggler effect in FL system, the delay of data computation and transmission for all wireless devices should be synchronized in each communication round. To this end, we formulate the following optimization problem as to minimize the average staleness function of delay for all wireless devices over a long duration:

$$\begin{aligned}
\mathrm{P1}: & \min_{P_n(i), R_n(i), B_n(i)} \lim_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} \sum_{n=1}^{N} g[a_n(i)] \\
\mathrm{s.t.}\, & C_1: 0 \leq P_n(i) \leq P_{\max}, \forall i, \forall n \in \mathcal{N}, \\
& C_2: B_n(i) \log_2 \left(1 + \frac{h_n(i) P_n(i)}{N_0 B_n(i)} \right) \geq R_{\min}, \forall n \in \mathcal{N}, \\
& C_3: T_n(i) \geq \frac{v C_n R_n(i) \log_2(1/\rho)}{\varphi_n(i)} + \frac{d_n}{B_n(i) \log_2 \left(1 + \frac{h_n(i) P_n(i)}{N_0 B_n(i)} \right)}, \\
& C_4: \lim_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} \mathbb{E}_{g[a_n(i)]} [g[a_n(i)] - g_0] \leq e_m, \forall n \in \mathcal{N}, \\
& C_5: \mathrm{EVaR}_{1-\Gamma}(Q_n) \leq \epsilon, \forall n \in \mathcal{N}, \\
& C_6: 0 \leq R_n(i) \leq A_n(i), \forall n \in \mathcal{N}, \\
& C_7: B_n(i) \geq 0, \forall i, \forall n \in \mathcal{N}, \\
& C_8: \sum_{n=1}^{N} B_n(i) \leq B, \forall i.
\end{aligned} \quad (16)$$

In problem P1, C_1 constrains the maximal transmit power of each wireless device. C_2 guarantees a minimum data rate of $R_{\min}$ for each wireless device during data transmission. C_3 requires a minimum amount of time to be allocated for computing and transmitting at each wireless device, in order to achieve the desired accuracy ρ and complete the transmission of the locally trained model. C_4 limits the staleness function by setting the thresholds g_0 and e_m on the desired staleness function value and the long-term average of deviations, respectively, in order to mitigate the straggler effect. C_5 mitigates the risk of discarding data in the buffer at the wireless devices and guarantees the collected data can be fully exploited for local training. C_6 ensures that the amount of data fetched by each wireless device for training the local model, $R_n(i)$, does not exceed the amount of data stored in its buffer, namely $A_n(i)$, in each communication round i. C_7 ensures that the bandwidth of each wireless device is nonnegative. Furthermore, C_8 ensures that the total bandwidth of all wireless device transmitting their local models synchronously does not exceed B.

Problem P1 is a nonconvex stochastic optimization problem with nonconvex constraint C_3 [41]. Besides, the resource allocation decisions over different communication rounds are coupled in the objective function and constraint C_5, posing major obstacles in online optimization. In Section IV, we will utilize the Lyapunov optimization and convex optimization techniques [35] to solve problem P1.

IV. PROPOSED PROBLEM SOLUTION

In this section, we first decompose P1 over different communication rounds by leveraging the Lyapunov optimization technique, which facilitates convenient decoupled online optimization in each communication round. Then, we tackle the remaining nonconvex objective function and constraints via problem transformation, and further solve the resulting problem by convex optimization techniques. Lastly, we analyze the complexity of the proposed iterative algorithm.

A. PROBLEM REFORMULATION USING LYAPUNOV OPTIMIZATION

1) EVaR Versus Queue Stability

Coping with the rapidly changing channel and queue conditions necessitates real-time decision-making at each time slot. However, the randomness of channel variations and data arrivals presents significant challenges for making optimal decisions while adhering to the long-term constraints. To address this, we apply the Lyapunov optimization method in this paper, which excels in online optimization by transforming a long-term averaged problem into a sequence of single time slot problems [35], [42]. Despite its effectiveness, conventional Lyapunov optimization based resource allocation approaches often impose additional stability requirements on data queues to ensure the method's applicability. In contrast, this paper departs from that tradition to maintain a more meaningful problem formulation. Nevertheless, we can show that the EVaR constraint C_5 in problem P1 inherently guarantees the stability of data queues.

Lemma 1: Let $\mathbf{X} = \{X(i), i \le T\}$ be a finite ergodic random process. The constraint on EVaR, i.e., $\mathrm{EVaR}_{1-\Gamma}(X(i)) < \epsilon < \infty$, implies that the queue with queue lengths given by $X(i)$ is both rate stable, i.e., $\lim\limits_{T\to\infty} \frac{X(i)}{T} = 0$, and mean rate stable, i.e., $\lim\limits_{T\to\infty} \frac{\mathbb{E}\{X(i)\}}{T} = 0$.

Proof: Please refer to Appendix A. ∎

Lemma 1 reveals that the EVaR constraint C_5 imposes a more stringent requirement than the conventional queue stability conditions. On the other hand, we will further show below that the EVaR constraint C_5 can also be interpreted as a stability condition for a *virtual* queue, imlpying that risk-awareness can be incorporated by applying conventional stability condition to appropriately defined queues. To this end, we reformulate C_5 as

$$\begin{aligned}
&C_5 : \mathrm{EVaR}_{1-\Gamma}(Q_n) \\
&= \min_{z>0} \left\{ z \ln \left[\frac{1}{\Gamma} \mathbb{E}_{Q_n} \left[\exp\left(\frac{1}{z}\frac{1}{i}\sum_{j=1}^{i} D_n(j) \right) \right] \right] \right\} \le \epsilon, \forall n, \\
&\overset{(a)}{\Leftrightarrow} z \ln \left[\frac{1}{\Gamma T} \sum_{i=1}^{T} \exp\left(\frac{1}{z}\frac{1}{i}\sum_{j=1}^{i} D_n(j) \right) \right] \le \epsilon, z > 0, \forall n, \\
&\Leftrightarrow \frac{1}{T}\sum_{i=1}^{T} z \exp \frac{1}{z} \left(\frac{1}{i}\sum_{j=1}^{i} D_n(j) - \epsilon \right) \le z\Gamma, z > 0, \forall n,
\end{aligned} \tag{17}$$

where (a) is due to the ergodicity of the random process $\{D_n(i), i \in T\}$ such that we have $\mathbb{E}\left[\exp\left(\frac{1}{z}\frac{1}{T}\sum_{i=1}^{T} D_n(i)\right)\right] = \frac{1}{T}\sum_{i=1}^{T}\exp\left(\frac{1}{z}\frac{1}{i}\sum_{j=1}^{i} D_n(j)\right)$. By applying Jensen's inequality, it can be shown that the following result holds for all $i > 0$:

$$\begin{aligned}
&\frac{1}{T}\sum_{i=1}^{T}\frac{1}{i}\sum_{j=1}^{i} z \exp\frac{1}{z}\left(D_n(j) - \epsilon\right) \\
&\le \frac{1}{T}\sum_{i=1}^{T} z \exp\frac{1}{z}\left(\frac{1}{i}\sum_{j=1}^{i} D_n(j) - \epsilon\right).
\end{aligned} \tag{18}$$

Therefore, we can relax C_5 as

$$\tilde{C}_5 : \frac{1}{T}\sum_{i=1}^{T}\frac{1}{i}\sum_{j=1}^{i} z \exp\frac{1}{z}\left(D_n(j) - \epsilon\right) \le z\Gamma, z > 0, \forall n. \tag{19}$$

Note that $\tilde{C}_5$ is a convex constraint.

Let us define virtual queues $\mathbf{Y}(i) = \{Y_n(i), n \in \mathcal{N}\}$ and $\mathbf{Z}(i) = \{Z_n(i), n \in \mathcal{N}\}$ with the following dynamics:

$$\begin{aligned}
Y_n(i+1) = \{Y_n(i) - z\Gamma\}^{+} + \frac{1}{i}\sum_{j=1}^{i} z \exp\frac{1}{z}\left(D_n[j] - \epsilon\right), \\
z > 0, n \in \mathcal{N},
\end{aligned} \tag{20}$$

$$Z_n(i+1) = \{Z_n(i) - e_m\}^{+} + g\left[a_n(i)\right] - g_0, n \in \mathcal{N}. \tag{21}$$

Constraints C_4 and $\tilde{C}_5$ are satisfied if $\mathbf{Z}(i)$ and $\mathbf{Y}(i)$ are mean-rate stable, i.e., $\lim_{i\to\infty}\frac{\mathbb{E}\{|\mathbf{Y}(i)|\}}{i} = 0$ and $\lim_{i\to\infty}\frac{\mathbb{E}\{|\mathbf{Z}(i)|\}}{i} = 0$ [35]. Consequently, problem P1 can be conveniently handled adopting the Lyapunov optimization technique with virtual queues $\mathbf{Y}(i)$ and $\mathbf{Z}(i)$, which is detailed in the following subsection.

2) Problem Reformulation via Lyapunov Optimization

Let $\mathbf{\Theta}(i) = \{D(i), \mathbf{Y}(i), \mathbf{Z}(i)\}$ denote the vector of all virtual and buffer queues. We consider the following quadratic Lyapunov function

$$L(i) = \frac{1}{2}\sum\nolimits_{n=1}^{N} Y_n^2(i) + \frac{1}{2}\sum\nolimits_{n=1}^{N} Z_n^2(i). \tag{22}$$

A large value of (22) implies a heavy loading in the virtual queues $\mathbf{Y}(i)$ and $\mathbf{Z}(i)$. To prevent from abrupt increase in the load, we introduce the following Lyapunov drift

$$\Delta L(i) = \mathbb{E}\left[L(i+1) - L(i)|\mathbf{\Theta}(i)\right], \tag{23}$$

where the expectation is taken with respect to the random channel states and the optimization variables that are functions of the channel states. The Lyapunov optimization aims to strike a balance between staleness function and load by minimizing the following drift-plus-penalty metric in every communication round

$$\Delta L(i) + V\mathbb{E}\left[\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N} g\left[a_n(i)\right]|\mathbf{\Theta}(i)\right], \tag{24}$$

where $V \geq 0$ is a predefined load control parameter that represents the tradeoff between the drift $\Delta L(i)$ and the penalty $\mathbb{E}\left[\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T}\sum_{n=1}^{N} g[a_n(i)]|\mathbf{\Theta}(i)\right]$. Specifically, a larger value of V is required to achieve a smaller long-term staleness in the objective value of the original optimization problem P1, whereas a smaller value of V is preferable to reduce the queue backlog and move towards a lower congestion state.

However, the Lyapunov drift-plus-penalty function in (24) is still nonlinear and intractable to optimize directly due to the expection. Instead of directly minimizing (24), based on the *opportunistic expectation minimization theory* [35, Chapter 1.8], we seek to minimize an upper bound of (24). For the virtual queue $\mathbf{Y}(i)$, we have

$$Y_n(i+1) = \{Y_n(i) - z\Gamma\}^+ + \frac{1}{i}\sum_{j=1}^{i} z\exp\frac{1}{z}\left(D_n\left[j\right] - \epsilon\right), \quad z > 0, n \in \mathcal{N}. \tag{25}$$

Note that, for any x, we have $\left(\max\{x, 0\}\right)^2 \leq x^2$. Applying the inequality to the virtual queues $\mathbf{Y}(i)$, we can obtain from (25) that

$$Y_n^2(i+1) - Y_n^2(i) \leq z^2\Gamma^2 + \frac{z^2}{i^2}\left\{\sum_{j=1}^{i}\exp\frac{1}{z}\left(D_n\left[j\right] - \epsilon\right)\right\}^2 + 2zY_n(i)\left(\frac{1}{i}\sum_{j=1}^{i}\exp\frac{1}{z}\left(D_n\left[j\right] - \epsilon\right) - \Gamma\right). \tag{26}$$

Similarly, we have

$$\begin{aligned} &Z_n^2(i+1) - Z_n^2(i) \\ &\leq e_m^2 + \left(g\left[a_n(i)\right] - g_0\right)^2 + 2Z_n(i)\left(g\left[a_n(i)\right] - g_0 - e_m\right), n \in \mathcal{N} \\ &\leq \left(g\left[a_n(i)\right]\right)^2 + 2g\left[a_n(i)\right]\left(Z_n(i) - g_0\right) + e_m^2 + g_0^2 \\ &\quad - 2Z_n(i)\left(g_0 + e_m\right), n \in \mathcal{N}. \end{aligned} \tag{27}$$

Based on (26) and (27), an upper bound of (24) can be obtained as (28) given at the top of the next page, where

$$\begin{aligned} f_{\text{total}} = \sum_{n=1}^{N} f_n = \sum_{n=1}^{N}\Bigg[&\frac{z^2}{2i^2}\left(\sum_{j=1}^{i}\exp\frac{1}{z}\left(D_n\left[j\right] - \epsilon\right)\right)^2 \\ &+ zY_n(i)\left(\frac{1}{i}\sum_{j=1}^{i}\exp\frac{1}{z}\left(D_n\left[j\right] - \epsilon\right)\right) + \frac{1}{2}\left(g\left[a_n(i)\right]\right)^2 \\ &+ g\left[a_n(i)\right]\left(Z_n(i) - g_0 + V\right) - zY_n(i)\Gamma - Z_n(i)\left(g_0 + e_m\right)\Bigg]. \end{aligned} \tag{29}$$

According to the principle of the Lyapunov optimization, we have converted the original problem P1 to minimize the upper bound (28), or (29) since the first two terms of (28) are constants, subject to the constraints in each communication round, i.e., C_1, C_2, C_3, C_6, C_7, and C_8. Then, P1 can be reformulated as

$$\begin{aligned} \text{P2}: &\min_{P_n(i), R_n(i), B_n(i)} f_{\text{total}} \\ &\text{s.t.} \quad C_1, C_2, C_3, C_6, C_7, C_8. \end{aligned} \tag{30}$$

Although constraint C_3 is still nonconvex, problem P2 is readily solvable, as elaborated below.

3) Transformation of Nonconvex Constraint C_3

To solve problem P2, we rewrite the nonconvex constraint C_3 as

$$\begin{aligned} &\frac{d_n}{B_n(i)\log_2\left(1 + \frac{h_n(i)P_n(i)}{N_0 B_n(i)}\right)} \leq T_n(i) - \frac{vC_nR_n(i)\log_2\left(1/\rho\right)}{\varphi_n(i)}, \forall n \in \mathcal{N}. \\ &\Leftrightarrow \tilde{C}_3: B_n(i)\log_2\left(1 + \frac{h_n(i)P_n(i)}{N_0 B_n(i)}\right) \\ &\quad - \frac{d_n\varphi_n(i)}{T_n(i)f_n - vC_nR_n\left(i\right)\log_2\left(1/\rho\right)} \geq 0, \forall n \in \mathcal{N}. \end{aligned} \tag{31}$$

Note that constraint $\tilde{C}_3$ is jointly convex with respect to $R_n(i)$, $B_n(i)$, and $P_n(i)$, as proved in Appendix B. Problem P2 can be reformulated as

$$\begin{aligned} \text{P3}: &\min_{P_n(i), R_n(i), B_n(i)} f_{\text{total}} \\ &\text{s.t.} \quad C_1, C_2, \tilde{C}_3, C_6, C_7, C_8. \end{aligned} \tag{32}$$

P3 is a convex optimization problem and fulfills the Slater's condition. Thus, strong duality holds for problem P3. This motivates us to solve P3 below by investigating its dual problem.

B. PROPOSED ITERATIVE ALGORITHM AND COMPUTATIONAL COMPLEXITY

Let $\widetilde{P_n(i)} = \{P_n(i)|0 \leq P_n(i) \leq P_{\max}\}$, $\widetilde{R_n(i)} = \{R_n(i)|0 \leq R_n(i) \leq A_n(i)\}$, and $\widetilde{B_n(i)} = \{B_n(i)|B_n(i) \geq 0\}$, which are independent feasible sets. Now, we define the Lagrangian associated with problem P3 as

$$\Delta L(t) + V\mathbb{E}\left[\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T}\sum_{n=1}^{N} g\left[a_n(i)\right]|\boldsymbol{\Theta}(i)\right]$$
$$= \mathbb{E}\left[\frac{1}{2}\sum\nolimits_{n=1}^{N}\left[Y_n^2(i+1) - Y_n^2(i) + Z_n^2(i+1) - Z_n^2(i)\right] + V\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T}\sum_{n=1}^{N} g\left[a_n(i)\right]|\boldsymbol{\Theta}(i)\right]$$
$$\leq \mathbb{E}\left[\frac{1}{2}\sum\nolimits_{n=1}^{N}\left[z^2\Gamma^2 + \frac{z^2}{i^2}\left\{\sum_{j=1}^{i}\exp\frac{1}{z}\left(D_n\left[j\right]-\epsilon\right)\right\}^2 + 2zY_n(i)\left(\frac{1}{i}\sum_{j=1}^{i}\exp\frac{1}{z}\left(D_n\left[j\right]-\epsilon\right)-\Gamma\right)\right]|\boldsymbol{\Theta}(i)\right]$$
$$+\mathbb{E}\left[\frac{1}{2}\sum\nolimits_{n=1}^{N}\left[\left(g\left[a_n(i)\right]\right)^2 + 2g\left[a_n(i)\right]\left(Z_n(i)-g_0\right) + e_m^2 + g_0^2 - 2Z_n(i)\left(g_0+e_m\right)\right] \mid \boldsymbol{\Theta}(i)\right] + V\mathbb{E}\left[\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T}\sum_{n=1}^{N} g\left[a_n(i)\right] \mid \boldsymbol{\Theta}(i)\right]$$
$$= \frac{1}{2}Nz^2\Gamma^2 + \frac{1}{2}N\left(e_m^2 + g_0^2\right) + \mathbb{E}\left[f_{\text{total}}|\boldsymbol{\Theta}(i)\right], \tag{28}$$

$\mathcal{L}\left(P_n(i), R_n(i), B_n(i); \mu\right)$, where μ is the dual variable or Lagrangian multiplier corresponding to constraint C_8. The Lagrange problem can be defined as follows

$$\begin{aligned}&\min_{P_n(i),R_n(i),B_n(i)} \mathcal{L}\left(P_n(i), R_n(i), B_n(i); \mu\right)\\ &= f_{\text{total}} + \mu\left(\sum_{n=1}^{N} B_n(i) - B\right)\\ &\text{s.}t. \quad C_1, C_2, \tilde{C}_3, C_6, C_7.\end{aligned} \tag{33}$$

Note that, given a fixed μ, each wireless device can calculate their own optimum considering their own local dynamics and constraints. The decomposed problem for bandwidth allocation $B_n(i)$ is

$$\begin{aligned}&\min_{P_n(i),R_n(i),B_n(i)} f_{\text{total}} + \sum\nolimits_{n=1}^{N}\mu B_n(i)\\ &\text{s.}t. \quad B_n(i) \in \widetilde{B_n(i)}.\end{aligned} \tag{34}$$

Similarly, the decomposed problem of power allocation $P_n(i)$ and data size $R_n(i)$ is denoted as

$$\begin{aligned}&\min_{P_n(i),R_n(i)} f_n\\ &\text{s.}t. \quad P_n(i) \in \widetilde{P_n(i)}, R_n(i) \in \widetilde{R_n(i)}.\end{aligned} \tag{35}$$

(34) can turn into

$$\max_{\mu} G(\mu) = f_{\text{total}}(P_n^*(i), R_n^*(i), B_n^*(i)) + \mu\left(\sum\nolimits_{n=1}^{N} B_n^*(i) - B\right). \tag{36}$$

To solve this dual problem of the original optimization P3, the system can update the global dual variable by the gradient method such as

$$\mu^{(j+1)} = \mu^{(j)} + \varsigma_{\mu,j}\left(\sum\nolimits_{n=1}^{N} B_n^*(\mu^{(j)}) - B\right), \tag{37}$$

where j is the iteration step, $\varsigma_{\mu,j}$ is the iteration step size of the dual variable $\mu^{(j)}$.

The proposed resource allocation procedure for buffer-aided FL is outlined in Algorithm 1. Therein, the buffer-aided FL system optimizes the allocated data size, transmit power, and bandwidth in each communication round for training local models. Different from centralized manner that the power and data will be centralized, the distributed manner enables to allocate the bandwidth to each wireless node firstly, and then allocate data size and transmit power according to the resource of each wireless node. The overall computational complexity of Algorithm 1 is dominated by the iterative search for the optimal transmit power and bandwidth. The computational complexity for solving problem P3 with Lagrange dual method is given by $O(N^{3.5} * \log_2(1/\epsilon_1)\log_2(1/\epsilon_2))$, with N wireless devices, where ϵ_1 and $\epsilon_2 > 0$ are the solution accuracy for transmit power and data size required in Algorithm 1. The number of iterations required for obtaining the optimal allocated data size is given by $O(1/\epsilon_0)$, where ϵ_0 specifies the required accuracy of the allocated bandwidth. Therefore, the overall computational complexity of the proposed optimal resource allocation scheme is given by $O(1/\epsilon_0 * N^{3.5} * \log_2(1/\epsilon_1)\log_2(1/\epsilon_2))$, which increases polynomially with N. The outer loop, described in Algorithm 1, has a bandwidth allocation for N nodes in I communication rounds. The computational complexity for total iteration is $O(I * 1/\epsilon_0 * N^{3.5} * \log_2(1/\epsilon_1)\log_2(1/\epsilon_2))$. It is important to note that the overall computation complexity is significantly reduced in two key aspects. First, the algorithm employs Lyapunov optimization, which approximates the optimization of a long-term average function by iteratively optimizing a sequence of instantaneous drift-plus-penalty functions. Second, the dual problem formulation is benefited to compute the optimal solution parallel and reduce the overhead of required side info among independent subproblems.

V. SIMULATION RESULTS

A. EXPERIMENTAL SETTINGS

The loss function is used to evaluate training performance of the data fetched from the buffer. We assume that $S_n(i)$ is the data size of input data $\mathbf{x}_n(i)$, the local loss function is defined

Algorithm 1 FL Based on Lyapunov Optimization

Initialization : Set the communication round $i = 1$, total communication rounds for global FL model T, the maximum tolerance $0 < \xi_0 \leq 1$. Initialize queue lengths $Y_n(1)$, and $Z_n(1)$ to be zeros.
while $i \leq T$ and Convergence = **false** **do**
  for $n = \{1, ..., N\}$ **do**
    Update $Y_n(i)$, $Z_n(i)$ according to equations (20) and (21), respectively.
    Update the local models of wireless devices with parameter $\mathbf{w}_n(i) = \mathbf{w}^*(i-1)$.
    Solve problem **P3** with with Lagrange dual method for given initialization and obtain resource allocation $\{P_n(i), R_n(i), B_n(i)\}$ according to Algorithm 2.
    return $\{P_n(i), R_n(i), B_n(i), \mathbf{Y}(i), \mathbf{Z}(i)\}$
    Perform local model training on data $R_n(i)$ with the loss function $f_n(\mathbf{w}_n \mid \mathbf{w}^*(i), R_n(i))$.
    Obtain the trained model parameters $\mathbf{w}_n(i)$ of wireless device $n \in \{1, \cdots, N\}$.
  end for
  Aggregate global model $\mathbf{w}^*(i)$ according to equation (1).
  Obtain the loss function $F(\mathbf{w}^*(i))$ of global model.
  if $|F(\mathbf{w}^*(i)) - F(\mathbf{w}^*(i-1))| < \xi_0$ **then**
    Convergence = **true**
  else
    return $\mathbf{w}^*(i)$ and Convergence = **false**
  end if
  $i = i + 1$
end while

Algorithm 2 Lagrange Dual Method for Solving Problem P3

Set the iteration step size $\varsigma_{\mu,j}$, and initialize $P_n(i), R_n(i), B_n(i)$.
repeat
  Set $j = 0$.
  repeat
    Update the Lagrange multiplier μ according to (37), and allocate $B_n(i)$ according to (34).
  until Convergence
  Update the intermediate values $P_n(i), R_n(i)$ according to (35). $j = j + 1$.
until Convergence

as follows [25]:

$$f_n(\mathbf{w}_n \mid \mathbf{w}^*(i-1), S_n(i)) = \left\| \mathbf{x}_n^T(i)\,\mathbf{w}_n - \mathbf{y}_n(i) \right\|_2, \quad (38)$$

where $\mathbf{x}_n^T(i)\,\mathbf{w}_n$ and $\mathbf{y}_n(i)$ denote the estimated value and the true value associated with the local training of wireless device n in round i, respectively. $\|\cdot\|_2$ denotes the Euclidean distance. The global loss function is defined as (2).

In this section, we evaluate the performance of the proposed algorithm via hardware-in-the-loop simulation. To this end, two different datasets are selected for training the local models. The first dataset, namely fixed total samples, obtained from [43], contains 5×10^3 measurement data samples about the received signal strength indicator (RSSI) of user equipments (UEs) in different locations in a multi-floor indoor environment. The RSSI is measured by access points (APs) deployed on different floors. The second dataset, non-fixed total samples, collects 5×10^5 data samples of bit error rate (BER) from optical modules, including temperature, voltage, bias current, input power, and output power of the optical modules from a measured testbed. The hardware testbed for measurement is shown in Fig. 4, including a BER meter and an optical module. Each dataset is divided into a training set and a test set, and the proportions of training and test data will be set during simulation.

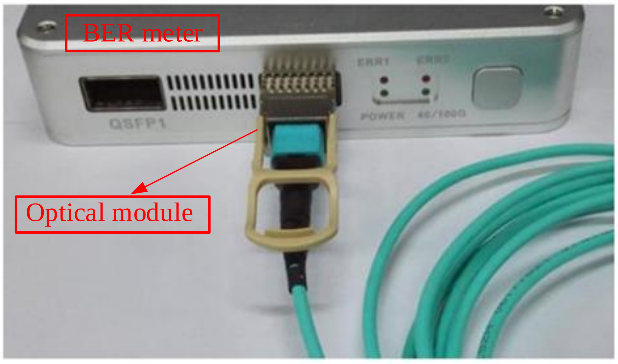


Figure 4. Data acquisition equipment.

For the hardware-in-the-loop simulation, we adopt nine laptops installed with developed machine learning software to simulate the wireless FL, where $N = 8$ laptops with different computation capacities emulate wireless devices and the ninth emulates the data center, respectively. The machine learning software includes projects of input data, queuing data in the buffer, processing data by machine learning, and transmitting data to other laptops. The data is collected to the laptops and each laptop trains the local model. When all the local models have been trained, they will be uploaded to a separate laptop serving as the data center for aggregation in FL. Then the data center returns the global model to the laptops simulating wireless devices. All time durations are recorded to calculate the delay. It should be noted that the hardware-in-the-loop FL system can receive the collected dataset from other hardware testbeds. The CPU frequency of each laptop is fixed in the simulation. Moreover, the CPU frequency is fixed as $\varphi_n(i) \equiv \varphi_n^*, \forall i$ in each local training, where φ_n^* is the CPU frequency the laptop simulating wireless device n.

For comparison, we also evaluate the performance of several baseline schemes.

- *Baseline Scheme 1:* Baseline scheme 1 employs synchronous FL with no buffers deployed at the wireless

Table 1. Simulation parameter settings. [44]

Parameter	Description	Default Value
ρ	Accuracy of local training	1/2
ξ_0	Convergence threshold of the global loss function	0.005
$R_{\min}$	Minimum data rate for each wireless device	0.5 Mbits
$P_{\max}$	Maximum transmit power of wireless device	1 W
Γ	Risk level	0.05
ϵ	Threshold of EVaR	3
I_n	Number of iterations	200
N_0	Noise power spectral density	-170 dBm/Hz
e_m	Threshold of delay exceedance	10^{-4}
C_n	Number of CPU cycles	$[1\times10^4, 3\times10^4]$ cycle/sample
φ_n	Computation capacity of wireless devices	$[1\times10^9, 4\times10^9]$ cycle/s

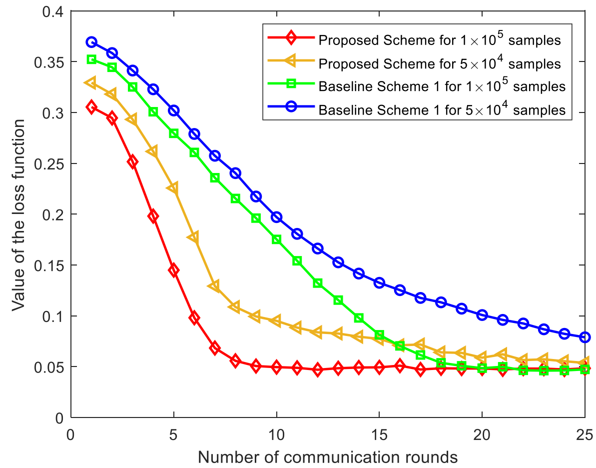


Figure 5. Value of the global loss function vs. the number of communication rounds.

devices [45]. As such, unlike the proposed scheme, the available data are equally allocated to different wireless devices. And all available data is utilized to train the local models, ignoring the heterogeneity of data. After all local models have been trained and uploaded, the data center aggregates the global model synchronously.

- *Baseline Scheme 2:* Baseline scheme 2 employs the asynchronous FL scheme in [46] with a buffer available at each wireless device. The available data are allocated to the wireless devices according to their computing and communication capabilities. Unlike synchronous FL, as long as two wireless devices have finished training their local FL models, they can upload their local models to the data center for aggregation. Upon receiving the partially aggregated local FL model, data center will use it to update the global FL model.
- *Baseline Scheme 3:* Baseline scheme 3 employs the synchronous FL scheme in [47], in which the edge node is equipped with a buffer. Unlike our proposed method, all available data is firstly stored in the buffer before chosen for training model with probability $p_n(i) = \frac{R_n(i)}{A_n(i)}$.

Unless otherwise specified, the simulation parameters are set according to Table I. The computation and communication processes repeat in each communication round until either the number of communication rounds exceeds a preset threshold or the global model converges. The latter means that the values of the loss function in consecutive iterations differ by no more than a specified threshold ξ_0.

The performance metrics to be evaluated include the loss function for FL model, cf. (38), the staleness function of proposed buffer-aided FL, cf. (15), and the computation time of training local models.

Based on (7), the average time of wireless device n required for training its local FL model can be calculated as

$$\tau = \frac{1}{N}\sum_{i=1}^{T}\sum_{n=1}^{N}\frac{I_n C_n R_n(i)}{\varphi_n(i)}. \tag{39}$$

B. COMPARISON OF LOSS FUNCTION VALUES

Fig. 5 shows the value of the global loss function of proposed scheme and baseline scheme 1 over multiple communication rounds for different sets of data samples with and without buffer. It can be seen from Fig. 5 that the buffer-aided synchronous FL model can achieve the same loss function threshold with fewer number of communication rounds than the synchronous FL model without buffer. This is because all wireless devices can adapt data sizes fetched from their buffers to accelerate the aggregation of the global FL model and therefore improve the convergence rate of buffer-aided synchronous FL. Fig. 5 also shows that with more data samples available, the loss function can approach the threshold much faster. This result implies that exploiting more data samples, the global FL model can achieve convergence rapidly.

In Fig. 6, we evaluate the accuracy of the resulting FL model, namely the accuracy of the trained model on the test dataset for different amounts of training time. The second dataset from 4 optical modules is used to train global model. The trained global model is used to evaluate the sample data of test set from different optical modules. The accuracy is the ratio of correct sample data to the total sample data. In this simulation, the input data for the FL model system is assumed to be of unlimited size. We observe that the accuracy of FL increases with the total training time. Meanwhile, a higher accuracy can be achieved as the number of wireless devices increases from $N = 6$ to $N = 10$, for a given total training time. This is because more wireless devices can compute more data and upload more trained parameters of local models. Consequently, the global FL model can obtain less loss function by aggregating more number of local FL models. On the other hand, the training time is also influenced by the number of epochs in each communication round. In FL, an "epoch" refers to one complete pass through the entire

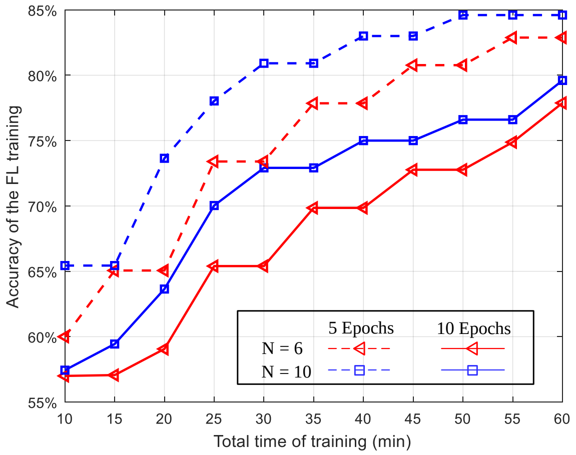


Figure 6. **The accuracy of the FL training vs. total time of training using proposed scheme.**

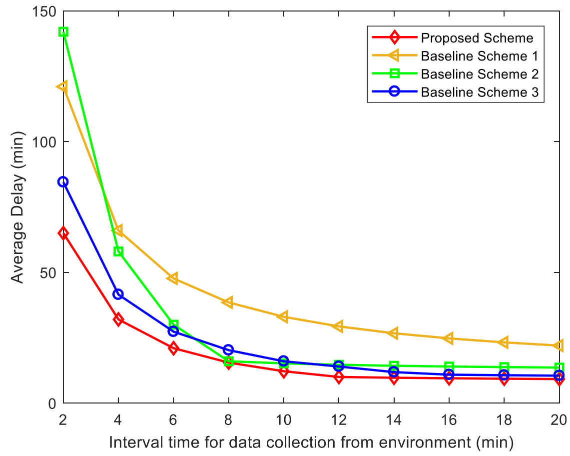


Figure 7. **Average delay vs. interval time of data collection.**

training dataset during the training process of a local model. When the number of epochs to training local FL model is set as 5 and 10, the accuracy of the FL training for different total training time is also evaluated. We observe that when the number of epochs is smaller, the accuracy of the FL training is higher due to more number of communication rounds within the same total amount of training time. This result shows that to achieve high evaluation accuracy of the FL model, we can appropriately reduce the number of epochs for the local FL model training and increase the number of communication rounds for aggregating global FL model.

C. COMPARISON OF AVERAGE DELAYS

Fig. 7 shows how the average delay of all wireless devices varies with the interval time of wireless devices in collecting data from environment, where the total data size is identical for all wireless devices. Note that the computation time of the FL also depends on the number of epochs for the local FL training in each communication round. To facilitate the comparison, in this subsection, the number of epoch is fixed as 5 for the training local FL models without further specified.For baseline scheme 1, all the availble data is equally allocated to each wireless device. While in the proposed scheme, the amount of data is allocated based on the computation capacity of wireless devices. Unlike proposed scheme, the asynchronous FL in baseline scheme 2 aggregates the global FL model if two local models are uploaded. It can explain that asynchronous FL scheme needs to aggregate the local model many communication round while the buffer-aided synchronous FL scheme aggregates the local FL model per communication round. As shown in Fig. 7, the computation time of the asynchronous FL scheme is longer than that of synchronous FL scheme because the asynchronous FL scheme needs more communication rounds and number of aggregations to ensure given performance of the FL model.

Fig. 7 reveals that the delay depends tightly on the time interval between consecutive data collections from the environment. Due to the completion time for each communication round is almost equal, the waiting time for data in the buffer significantly increases when the interval time is small, which greatly affects the average delay. In all schemes, the average delay decreases with the interval time for data collection. We also observe that due to the straggler effect, the synchronous FL model without buffer is larger than two other methods. Therefore, the delay for synchronous FL without buffer is much longer than other schemes. This result also shows the advantage of computation efficiency of the buffer-aided synchronous FL scheme. We observe from Fig. 7 that as the interval time of data collection is small, baseline scheme 3 results in larger average delays than our proposed scheme. When the interval time of data collection increases, the gap of average delay between the two schemes reduces. Unlike our proposed scheme, baseline scheme 3 probabilistically choose data for training. When the interval time for data collection is small, the time for data chosen increases the average delay of baseline scheme 3.

Fig. 8 shows under different gaps of the loss function in the adjacent communication round, the required training time for different accuracies of the FL training. We evaluate the computation time to alter the number of wireless devices, where the total data size is identical for all wireless devices. Since the accuracy is for a range of training time, we represent in Fig. 8 the intermediate values for the time periods. We can observe that to achieve the required FL training accuracy, more wireless devices require less training time. This result is reasonable that more wireless devices can obtain more accurate global FL model due to more local FL models are aggregated at the data center. It can also be observed that if the gap of the loss function in the adjacent communication round is smaller, the FL system require more training time. It is interesting that even the consumed training time with gap

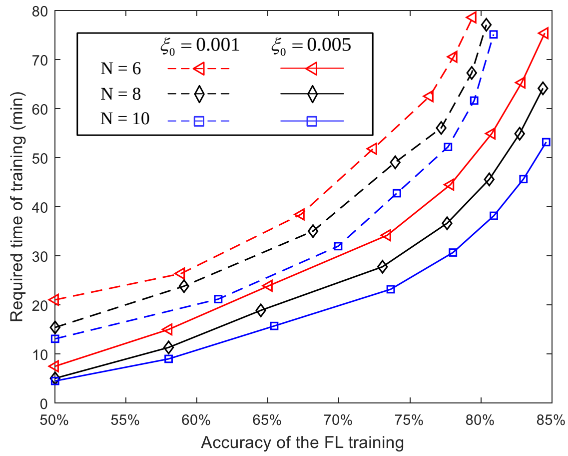


Figure 8. **The required global training time vs. accuracy of the FL training.**

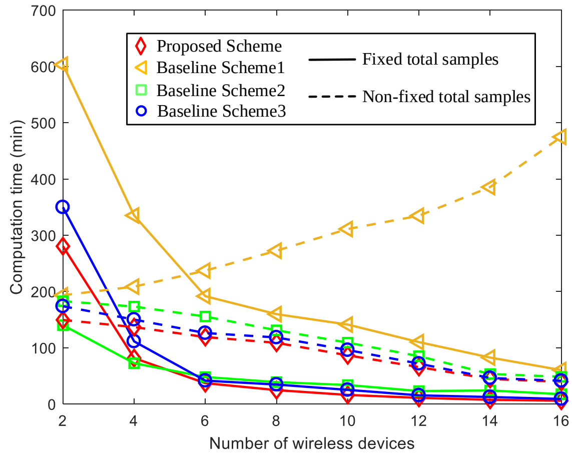


Figure 9. **Computation time vs. number of wireless devices.**

of the loss function $\xi_0 = 0.001$ is more than with $\xi_0 = 0.005$, the accuracy of the FL training with $\xi_0 = 0.001$ is not good than that with $\xi_0 = 0.005$. This result means that the setting of the gap of the loss function impacts on the training time but not necessarily affect accuracy. However, the accuracy more depends on the result calculated by the trained model.

Fig. 9 shows the total computation time changes with the number of wireless devices under 4 FL schemes, our proposed buffer-aided synchronous model, synchronous model without buffer, and asynchronous model. We consider the loss function for two adjacent rounds of communication to be less than $\xi_0 = 0.005$ as converged, and discuss two cases of limited dataset and unlimited dataset. It can be observed that the computation time of our proposed buffer-aided synchronous model use shorter computation time than synchronous without buffer under all number of wireless devices, but it is less than asynchronous method only the number of wireless devices is more than 6. The reason is that the asynchronous model should use more time to aggregate global model when the number of wireless devices is larger. However, the time consumption of buffer-aided synchronous model for aggregating global FL model is independent of the number of wireless devices. Due to the straggler effect, the time consumption of synchronous model without buffer is more than other two FL schemes. Additionally, the computation time of baseline 3 is larger than that of our proposed scheme when the number of wireless devices is small. This result is because the computation time in each round is short when the number of wireless devices is small. The time consumption for choosing data with probability in baseline 3 occupies a very large part of the computation time in each round, which leads to the computation time being larger than our proposed scheme. On the other hand, the computation time for the unlimited data set is more than limited data set. This result is due to unlimited data set introduce more randomness and data size increases with the number of wireless devices.We can find

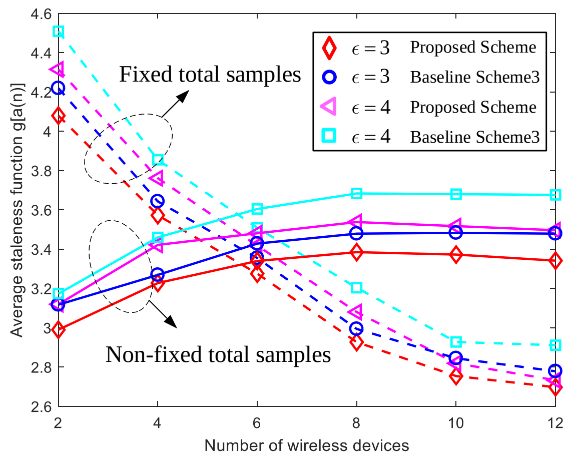


Figure 10. **Average staleness function vs. number of wireless devices.**

that the time consumption of the synchronous model without buffer increases with the number of wireless devices. This is due to the wireless devices have almost same data size for unlimited data set, the total data size increase with the number of wireless devices. The increase of the data size introduces more randomness for training FL model and hence, the total consumption time increases.

D. COMPARISON OF STALENESS FUNCTION VALUES

Fig. 10 illustrates the effect of the number of wireless devices on long-term average $g[a(n)]$ for different EVaR upper bound ϵ for limited and unlimited data set. For limited data set, we can find that $g[a(n)]$ decreases as the number of wireless devices increases due to the data is distributed to more wireless devices. For unlimited data set, with the number of wireless devices increasing, the amount of data will increase

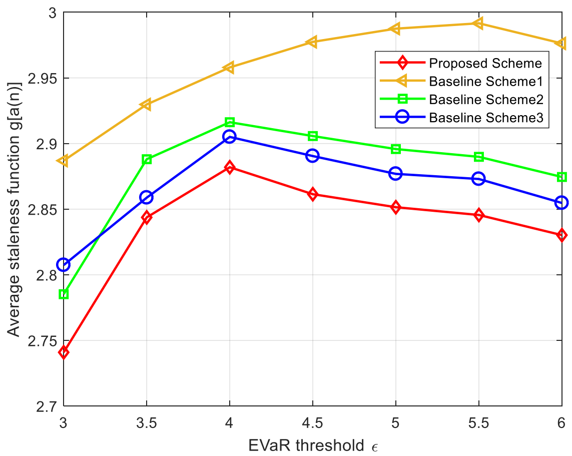


Figure 11. Average staleness function $g[a(n)]$ vs. EVaR threshold ϵ.

exponentially, which will increase the amount of data on the wireless devices. However, with buffer on the wireless devices, data can be allocated to the wireless devices with large computational capability, which may lead to $g[a(n)]$ decreases. The average staleness function of baseline scheme 3 is larger than that of our proposed algorithm. In baseline 3, the data used to train the local model must wait for the edge node to discard previously collected data with probability and hence, the average delay of data is larger than our proposed method. According to the fairness function described in (15), the staleness function of baseline 3 is larger than that of our proposed method.

In Fig. 11, we evaluate the relationship between the staleness function and the EVaR threshold ϵ for the limited data set. Our findings indicate that the EVaR threshold of the data collection queue significantly impacts the average staleness function in the FL system. Notably, as ϵ increases, the average staleness function initially rises and then decreases across all four FL schemes. This behavior occurs because the average staleness function is influenced by both the buffer size and the amount of data required for training the local model. When ϵ is small, the buffer can only store a limited amount of collected data, resulting in a low average staleness function due to the discarding of some data. Conversely, with a large ϵ, more data is used for local FL model training, leading to increased queue delays. Consequently, the average staleness function first increases with ϵ and then decreases. It is important to note that there is an optimal EVaR threshold where the average staleness is maximized. This threshold balances the amount of data discarded with the data used to train the local FL model at a fixed learning rate. Among the four transmission schemes, the buffer-aided synchronous FL model achieves the lowest average staleness function, indicating that data experiences minimal waiting time in the buffer.

VI. CONCLUSION

This paper proposed a novel buffer-aided synchronous FL framework for improving the convergence rate and overall loss function. To facilitate the performance improvement of the FL system deployed in data collection applications, we introduced the delay to represent the freshness of collected data and utilized the EVaR of the queue length to guarantee the risk of data loss at the buffer. We formulated a nonconvex optimization problem for minimizing the average staleness function of global FL system within a given tolerable risk of data loss at the buffer equipped at the wireless device for local FL model. We transformed the non-convex constraints of the EVaR limitation and time consumption for data processing to the convex counterpart. Meanwhile, exploiting the Lyapunov optimization technique, we decomposed the non-convex problem into convenient online optimizations in each frame, which were further solved by convex optimization. Simulation results unveiled that our proposed algorithm can effectively mitigate the risk of data loss in queues and minimizing the staleness function for the FL system, compared to the counterpart synchronous FL framework without a buffer. These results demonstrated the huge potential of employing buffer in wireless devices to enhance the performance of synchronous FL systems.

APPENDIX A
PROOF OF LEMMA 1

Due to the ergodicity of random process $\mathbf{X}$, we have $\mathbb{E}[\exp(X(i)/z)] = \lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T}\exp(X(i)/z)$. According to the definition in (13), the EVaR of $\mathbf{X}$ is given as

$$\mathrm{EVaR}_{1-\Gamma}(X(i)) = \min_{z>0}\left\{z\ln\left[\frac{1}{\Gamma T}\sum_{t=1}^{T}\exp\left(\frac{X(i)}{z}\right)\right]\right\}. \tag{40}$$

Meanwhile, following the properties of the log-sum-exp function, we have

$$\begin{aligned}&\max\{X(0),...,X(T)\}\\&\leq z\ln(\Gamma T)+z\ln\left[\frac{1}{\Gamma T}\sum_{i=1}^{T}\exp\left(\frac{X(i)}{z}\right)\right], z>0,\\&\leq z\ln(\Gamma T)+\epsilon, z>0.\end{aligned} \tag{41}$$

Since for any i, we have $X(i) \leq \max\{X(i),...,X(T)\}$ and $\mathbb{E}\{X(i)\} \leq \max\{X(0),...,X(T)\}$. Meanwhile, for any finite $z>0$, we have $\lim_{T\to\infty}\frac{z\ln(\Gamma T)+\epsilon}{T} = 0$. Therefore, the limits $\lim_{T\to\infty}\frac{X(i)}{T}$ and $\lim_{T\to\infty}\mathbb{E}\frac{\{X(i)\}}{T}$ exist and satisfy $\lim_{T\to\infty}\frac{X(i)}{T} = \lim_{T\to\infty}\mathbb{E}\frac{\{X(i)\}}{T} = 0$, which completes the proof.

APPENDIX B
PROOF OF CONVEXITY OF C_3

We first show that the first term in (31) is a jointly concave function of $B_n(i)$ and $P_n(i)$. Define $x \triangleq B_n(i)$, $y \triangleq \frac{h_n(i)P_n(i)}{N_0}$, and $f_1(x,y) = -x\log_2\left(1+\frac{y}{x}\right) \triangleq$

$-B_n(i)\log_2\left(1+\frac{h_n(i)P_n(i)}{N_0B_n(i)}\right)$, then the first term in (31) is equal to $-f_1(x,y)$. The Hessian matrix of $f_1(x,y)$ is

$$\mathbf{H}=\begin{bmatrix}\frac{y^2/x}{(x+y)^2} & -\frac{y}{(x+y)^2}\\ -\frac{y}{(x+y)^2} & \frac{x}{(x+y)^2}\end{bmatrix}, \tag{42}$$

which is positive semidefine with nonnegative eigenvalues

$$\begin{aligned}\sigma_1&=0,\\ \sigma_2&=\frac{x^2+y^2}{x^3+2x^2y+xy^2}\geq 0.\end{aligned} \tag{43}$$

Therefore, $-f_1(x,y)$ is a concave function.

Now define the second term in (31) as $f_2(R_n(i)) = -\frac{d_n\varphi_n(i)}{T_n(i)f_n-vC_nR_n(i)\log_2(1/\rho)}$. We have

$$\frac{\partial^2 f_2(R_n(i))}{\partial^2 R_n(i)}=-\frac{2d_n\varphi_n(i)[vC_n\log_2(1/\rho)]^2}{[T_n(i)f_n-vC_nR_n(i)\log_2(1/\rho)]^3}<0. \tag{44}$$

Thus, $f_2(R_n(i))$ is a concave function of $R_n(i)$.

Therefore, $\tilde{C}_3$ can be transformed into $-f_1(x,y)+f_2(R_n(i))\geq 0$, which is jointly convex with respect to $R_n(i)$, $B_n(i)$, and $P_n(i)$. This ends the proof.

References

[1] J. Zhang, J. Zheng, L. Xiang, and X. Ge, "Risk-aware online optimization of cross-Layer resource allocation in next-generation WLANs," in *Proc. CIC/ICCC*, 2023, pp. 1–6.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[3] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[4] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2238–2251, 2021.

[5] H. Zhou, M. Li, P. Sun, B. Guo, and Z. Yu, "Accelerating federated learning via parameter selection and pre-synchronization in mobile edge-cloud networks," *IEEE Trans. Mobile Comput.*, pp. 1–15, 2024.

[6] M. A. Husnoo, A. Anwar, H. T. Reda, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss, "Fedisa: A semi-asynchronous federated learning framework for power system fault and cyberattack discrimination," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[7] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2021.

[8] H. Zhu, Y. Zhou, H. Qian, Y. Shi, X. Chen, and Y. Yang, "Online client selection for asynchronous federated learning with fairness consideration," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2493–2506, 2023.

[9] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and aggregation design for asynchronous federated learning over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 874–886, 2023.

[10] L. You, S. Liu, Y. Chang, and C. Yuen, "A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24 199–24 211, 2022.

[11] T. Zhang, K.-Y. Lam, and J. Zhao, "Device scheduling and assignment in hierarchical federated learning for internet of things," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 18 449–18 462, 2024.

[12] J. Shi, Y. Wu, D. Zeng, J. Tao, J. Hu, and Y. Shi, "Self-supervised on-device federated learning from unlabeled streams," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4871–4882, 2023.

[13] D.-Y. Kim, D.-E. Lee, J.-W. Kim, and H.-S. Lee, "Collaborative policy learning for dynamic scheduling tasks in cloud–edge–terminal IoT networks using federated reinforcement learning," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10 133–10 149, 2024.

[14] X. Zhou, X. Zheng, X. Cui, J. Shi, W. Liang, Z. Yan, L. T. Yang, S. Shimizu, and K. I.-K. Wang, "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3191–3211, 2023.

[15] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 2021.

[16] C. You, K. Guo, H. H. Yang, and T. Q. S. Quek, "Hierarchical personalized federated learning over massive mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8141–8157, 2023.

[17] Y. Wang, L. Xiang, and A. Klein, "Completion time minimization for uav-based communications with a finite buffer," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 6404–6410.

[18] M. Mohammadkhani Razlighi and N. Zlatanov, "Buffer-aided relaying for the two-hop full-duplex relay channel with self-interference," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 477–491, 2018.

[19] G. Stamatakis, N. Pappas, and A. Traganitis, "Optimal policies for status update generation in an iot device with heterogeneous traffic," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5315–5328, 2020.

[20] D. Han and W. Chen, "QoS differential scheduling of urllc under fifo service discipline: A cross-layer approach," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1370–1373, 2020.

[21] P. Söderman, K.-J. Grinnemo, M. Hidell, and P. Sjödin, "A comparative analysis of buffer management algorithms for delay tolerant wireless sensor networks," *IEEE Syst. J.*, vol. 21, no. 7, pp. 9612–9619, 2021.

[22] H. Huang, C. Zeng, Y. Zhao, G. Min, Y. Zhu, W. Miao, and J. Hu, "Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2558–2571, 2021.

[23] Q. Wei, H. Ma, C. Chen, and D. Dong, "Deep reinforcement learning with quantum-inspired experience replay," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9326–9338, 2021.

[24] C. Wang, Q. Wu, Q. Ma, and X. Chen, "A buffered semi-asynchronous mechanism with MAB for efficientfederated learning," in *2022 International Conference on High Performance Big Data and Intelligent Systems (HDIS)*. IEEE, 2022, pp. 180–184.

[25] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2020.

[26] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless quantized federated learning: A joint computation and communication design," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2756–2770, 2023.

[27] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optim. Method Softw.*, vol. 32, no. 5, pp. 993–1005, 2017.

[28] A. Mahmoudi, H. S. Ghadikolaei, J. M. Barros Da Silva Júnior, and C. Fischione, "Fedcau: A proactive stop policy for communication and computation efficient federated learning," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2024.

[29] H. Li, R. Wang, W. Zhang, and J. Wu, "One bit aggregation for federated edge learning with reconfigurable intelligent surface: Analysis and optimization," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 872–888, 2023.

[30] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency wlans," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 197–216, 2019.

[31] J. Tang, "The shannon channel capacity of dispersion-free nonlinear optical fiber transmission," *J. Lightw. Technol*, vol. 19, no. 8, p. 1104, 2001.

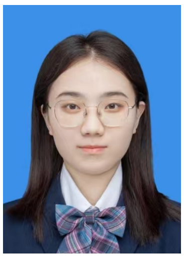
[32] H. Yuan and P.-Y. Kam, "Soft-decision-aided, smoothness-constrained channel estimation over time-varying fading channels with no channel model information," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 73–86, 2017.

[33] Y.-L. Hsu, C.-F. Liu, S. Samarakoon, H.-Y. Wei, and M. Bennis, "Age-Optimal Power Allocation in Industrial IoT: A Risk-Sensitive Federated Learning Approach," in *Proc. IEEE Personal, Indoor and Mobile Radio Commun. Sympos.*, 2021, pp. 1323–1328.

[34] H. Föllmer and A. Schied, *Stochastic finance: an introduction in discrete time*. Walter de Gruyter, 2011.

[35] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Springer Nature, 2022.

[36] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *The Annals of Mathematical Statistics*, pp. 493–507, 1952.
[37] A. Ahmadi-Javid, "Entropic value-at-risk: A new coherent risk measure," *J. Optim. Theory Appl.*, vol. 155, pp. 1105–1123, 2012.
[38] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *J. Commun. Netw.*, vol. 21, no. 3, pp. 204–219, 2019.
[39] C. Shapiro and H. R. Varian, *Information rules: A strategic guide to the network economy*. Harvard Business Press, 1999.
[40] S. Razniewski, "Optimizing update frequencies for decaying information," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1191–1200.
[41] C. Ye and Y. Cui, "Stochastic successive convex approximation for general stochastic optimization problems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 6, pp. 755–759, 2020.
[42] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, 2020.
[43] Z. Tóth and J. Tamás, "Miskolc IIS hybrid IPS: Dataset for hybrid indoor positioning," in *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2016, pp. 408–412.
[44] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2021.
[45] Z. Jiang, W. Wang, B. Li, and Q. Yang, "Towards efficient synchronous federated training: A survey on system optimization strategies," *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 437–454, 2022.
[46] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *Computer Science Review*, vol. 50, p. 100595, 2023.
[47] Y. Kang and B. Li, "Polaris: Accelerating asynchronous federated learning with client selection," *IEEE Trans. Cloud Comput.*, vol. 12, no. 2, pp. 446–458, 2024.

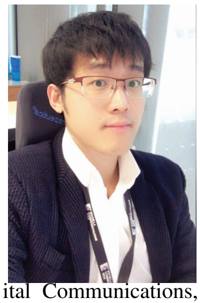


JING LIU received her undergraduate degree from Wuhan University of Technology and is currently pursuing a master's degree at Huazhong University of Science and Technology. Her current research interests include age-of-information and federated learning.

JINKE ZHENG received her undergraduate degree from Huazhong University of Science and Technology, where she is currently pursuing a master's degree. Her current research interests include green communication, resource allocation, and convex optimization.

JING ZHANG received the Ph.D. degree from Huazhong University of Science and Technology (HUST) in 2002 and 2010, respectively. He is currently an associate professor with HUST. His current research interests include unmanned aerial vehicle communications, green communications, millimeter-wave communications, reconfigurable intelligent surfaces, integration of sensing and communication, and optical and wireless fusion networks.

LIN XIANG (S'14, M'18) received the PhD degree from the Friedrich-Alexander University of Erlangen-Nuremberg (FAU), Germany. He is currently a junior research group leader at the Communications Engineering Lab, Technical University of Darmstadt, Germany. His research interests include physical layer design, modeling and optimization of wireless communication systems, with applications to wireless caching, unmanned aerial vehicles (UAVs) and autonomous vehicles, as well as joint sensing, communication, and computing.



DERRICK WING KWAN NG (S'06-M'12-SM'17-F'21) received his bachelor's degree (with first-class Honors) and the Master of Philosophy degree in electronic engineering from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2006 and 2008, respectively, and his Ph.D. degree from The University of British Columbia, Vancouver, BC, Canada, in November 2012. Following his Ph.D., he was a senior postdoctoral fellow at the Institute for Digital Communications, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Germany. He is currently a Scientia Associate Professor with the University of New South Wales, Sydney, NSW, Australia. His research interests include global optimization, integrated sensing and communication (ISAC), physical layer security, IRS-assisted communication, UAV-assisted communication, wireless information and power transfer, and green (energy-efficient) wireless communications.

He has been recognized as a Highly Cited Researcher by Clarivate Analytics (Web of Science) since 2018. He was the recipient of the Australian Research Council (ARC) Discovery Early Career Researcher Award 2017, IEEE Communications Society Leonard G. Abraham Prize 2023, IEEE Communications Society Stephen O. Rice Prize 2022, Best Paper Awards at the WCSP 2020, 2021, IEEE TCGCC Best Journal Paper Award 2018, INISCOM 2018, IEEE International Conference on Communications (ICC) 2018, 2021, 2023, 2024, IEEE International Conference on Computing, Networking and Communications (ICNC) 2016, IEEE Wireless Communications and Networking Conference (WCNC) 2012, IEEE Global Telecommunication Conference (Globecom) 2011, 2021, 2023 and IEEE Third International Conference on Communications and Networking in China 2008. From January 2012 to December 2019, he served as an Editorial Assistant to the Editor-in-Chief of the IEEE Transactions on Communications. He is also the Editor of the IEEE Transactions on Communications and an Associate Editor-in-Chief for the IEEE Open Journal of the Communications Society.