Saddle Point Approximation Based Delay Analysis for Wireless Federated Learning

Longwei Yang*, Lintao Li*, Xin Guo‡, Yuanming Shi†,Haiming Wang‡, and Wei Chen*
* Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, 100084, China
† School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
‡ Lenovo Research, Beijing, 100094, China
Email: ylw18@mails.tsinghua.edu.cn, llt20@mails.tsinghua.edu.cn, guoxin9@lenovo.com,
shiym@shanghaitech.edu.cn, wanghm14@lenovo.com, wchen@tsinghua.edu.cn

***Abstract*—Wireless federated learning (FL) holds the potential of preserving data privacy and reducing network traffic congestion, thereby attracting much recent attention. Due to the fading nature of wireless channels, wireless FL suffers from the random delay in each uplink and downlink transmission. As a result, how to analyze the overall random delay of a FL task over wireless fading channels remains open. To solve this challenging problem, we present a saddle point approximation based approach to obtain the distribution of the delay caused by communication in wireless FL systems. In particular, we obtain the uplink delay distribution and the downlink delay distribution by Lugannani-Rice formula. The overall delay distribution is then obtained through the convolution of those two distributions and the generating function. Simulation results demonstrate that the theoretical results provide accurate characterizations for the empirical results, which corroborates the validity of the analysis in this paper.**

I. Introduction

Tremendous developments of communication technology and machine learning has generated a huge amount of data on user devices. With the development of Artificial Intelligence (AI) and the wide scale deployment of edge devices, edge AI [1] has emerged as a promising alternative in the Six-Generation (6G) communication [2]. In order to let edge nodes make full use of those data, the concept of "bringing the code to the data, instead of the data to the code." is introduced by Federated Learning (FL) to preserve data privacy and reduce network traffic congestion. Federated Learning, proposed by Google [3] in 2016, is a distributed machine learning method which enables training on a fleet of participating distributed user devices such as mobile phones or vehicles.

In recent years, several research efforts have been conducted on FL. Previous works can be divided into three categories: first, improving the prediction performance of the FL model. Second, reducing the transmission load. Third, analysing the convergence rate. Some of them focus on analog transmission and aggregation built on the appealing idea of over-the-air computation [4]–[6]. It is achieved by exploring the superposition property of a wireless multiple-access channel to compute the desired function (i.e., the weighted average function) of distributed locally computed updates via concurrent transmission. In [4], joint device selection and beamforming design are developed to improve the learning performance. Some of them consider reducing the communication overhead via quantization [7], [8], sparsification, or coding [9]. In [8], they propose a hierarchical stochastic gradient quantization framework and efficiently quantize gradients using Grassmann manifold and hinge vector under the criterion of minimum distortion. The convergence rate of FL has also been investigated in previous works. In [10], a probabilistic scheduling framework is introduced considering update importance, which is measured by the difference between the scaled local gradient and the ground-truth global gradient. In [11], the upper bound of global iteration is derived under the assumption that the loss function is L-smooth and γ-strongly convex. They propose to minimize the weighted completion time and energy consumption under bandwidth and power constrains. Although many researches on FL has been done, few of they provide the theoretical analysis of transmission delay. In [12], the authors focused on the connection between the performance of FL algorithms and wireless networks, while they derived the optimal transmit power allocation and uplink resource block allocation scheme for each user. However, there is not any literature focusing on the delay distribution analysis in FL systems.

In this paper, we shall propose to analyse the transmission delay and the convergence time by means of saddle point approximation. The contributions of this paper are summarized as follow

- We present a FL transmission model over wireless fading channels to analyse the delay distribution caused by communication. We derive the distribution of the one iteration delay by means of saddle point approximation and Lugannani-Rice formula.
- Experiments show that the simulation results are in good agreement with the theoretical results in terms of the distribution of the one iteration delay. We also obtain the empirical distribution of the number of overall iterations, through which we can derive the theoretical convergence time of the training process.

The rest of this paper is organized as follows. Section II

This research is supported by Lenovo Research, the National Natural Science Foundation of China under Grant No. 61671269, and the National Key Research and Development Program 2018YFA0701601.

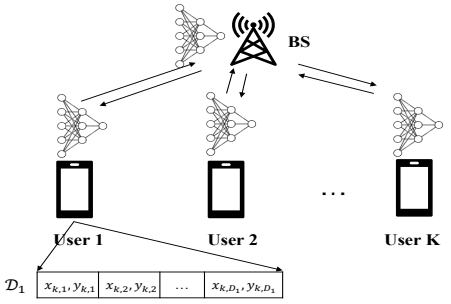

Fig. 1. The considered FL model.

Algorithm 1 Federated Learning Algorithm

1: Initialize global regression vector w^0 and iteration number $n = 0$.
2: **repeat**
3: Each user k computes $\nabla F_k(\boldsymbol{w}^{(n)})$ and sends it to the BS.
4: The BS computes

$$\nabla F(\boldsymbol{w}^{(n)}) = \frac{1}{K}\sum_{k=1}^{K} \nabla F_k(\boldsymbol{w}^{(n)}), \tag{1}$$

and broadcast it to all involved users.
5: **for** user $k \in \mathcal{K}$ **do**
6: Solve local problem (5) to get the solution $\boldsymbol{h}_k^{(n)}$.
7: Each user sends $\boldsymbol{h}_k^{(n)}$ to the BS.
8: **end for**
9: The BS computes

$$\boldsymbol{w}^{(n+1)} = \boldsymbol{w}^{(n)} + \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{h}_k^{(n)}, \tag{2}$$

and broadcast it to all involved users.
10: Set $n = n + 1$.
11: **until** the accuracy ϵ_0 of problem (4) is obtained.

presents the system model, which contains the FL model and the transmission model. Based on the FL model and transmission model, delay and convergence time analysis are given in Section III. In Section IV, experiment setting and simulation results are described. The conclusion is given in Section V.

II. System Model

Consider a wireless network that consists of a set $\mathcal{K}$ of K users and one base station (BS) both with single antenna, as shown in Fig. 1. Each user k has a local dataset $\mathcal{D}_k$ with D_k samples, where $\mathcal{D}_k = \{\boldsymbol{x}_{k,i}, y_{k,i}\}_{i=1}^{D_k}$, $\boldsymbol{x}_{k,i} \in \mathbb{R}^d$ is the input vector and $y_{k,i} \in \mathbb{R}$ is the corresponding output. In this section, the FL model and transmission model are introduced respectively.

A. Federated Learning Model

In this subsection, the FL model used in the wireless network is introduced, followed by some assumptions on the learning model. The model trained by each device's dataset is called the *local FL model*, while the shared model generated by the BS is called the *global FL model*.

The parameter of the global model is denoted by $\boldsymbol{w}$. The loss function used for performance measurement is given by $f(\boldsymbol{w}, \boldsymbol{x}_{k,i}, y_{k,i})$. The loss function can be different for different learning tasks, such as $f(\boldsymbol{w}, \boldsymbol{x}_{k,i}, y_{k,i}) = \frac{1}{2}(\boldsymbol{x}_{k,i}^T\boldsymbol{w} - y_{k,i})^2$ for linear regression and hinge loss $f(\boldsymbol{w}, \boldsymbol{x}_{k,i}, y_{k,i}) = \max(0, 1 - y_{k,i}\boldsymbol{x}_{k,i}^T\boldsymbol{w})$ for linear Support Vector Machine (SVM). Thus, the total loss function of user k is given by

$$F_k(\boldsymbol{w}, \boldsymbol{x}_{k,1}, y_{k,1}, ..., \boldsymbol{x}_{k,D_k}, y_{k,D_k}) = \frac{1}{D_k}\sum_{i=1}^{D_k} f(\boldsymbol{w}, \boldsymbol{x}_{k,i}, y_{k,i}), \tag{3}$$

where $F_k(\boldsymbol{w}, \boldsymbol{x}_{k,1}, y_{k,1}, ..., \boldsymbol{x}_{k,D_k}, y_{k,D_k})$ is simplified by $F_k(\boldsymbol{w})$. The goal of training process is to find a global FL model that minimizes the weighted sum of every involved user's loss function without sharing any local dataset. The FL training problem can be formulated as

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) \triangleq \sum_{k=1}^{K} \frac{D_k}{D} F_k(\boldsymbol{w}) = \frac{1}{D}\sum_{k=1}^{K}\sum_{i=1}^{D_k} f(\boldsymbol{w}, \boldsymbol{x}_{k,i}, y_{k,i}), \tag{4}$$

where $D = \sum_{k=1}^{K} D_k$. Following [3] and [12], we adopt the FL algorithm summarized in Algorithm 1. The global FL parameter at iteration n is denoted by $\boldsymbol{w}^{(n)}$. Each user computes the local FL problem:

$$\min_{\boldsymbol{h}_k} G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k) \triangleq F_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k) - \big(\nabla F_k(\boldsymbol{w}^{(n)}) - \xi \nabla F(\boldsymbol{w}^{(n)})\big)^T \boldsymbol{h}_k, \tag{5}$$

where ξ is a step size parameter and $\boldsymbol{h}_k$ is the difference between the global FL parameter and the local FL parameter for user k. Here we obtain a feasible solution of (5) with a given accuracy η instead of the optimal solution due to the difficulty to find the optimal solution [12]. The solution $\boldsymbol{h}_k^{(n)}$ with accuracy η at iteration n meets the requirement

$$G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n)}) - G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n)^*}) \leq \eta\big(G_k(\boldsymbol{w}^{(n)}, \boldsymbol{0}) - G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n)^*})\big), \tag{6}$$

where $\boldsymbol{h}_k^{(n)^*}$ is the optimal solution. Similarly, we obtain a feasible solution $\boldsymbol{w}^{(n)}$ of (4) with a given accuracy ϵ_0, which meets the requirement

$$F(\boldsymbol{w}^{(n)}) - F(\boldsymbol{w}^*) \leq \epsilon_0\big(F(\boldsymbol{w}^{(0)}) - F(\boldsymbol{w}^*)\big), \tag{7}$$

where $\boldsymbol{w}^*$ is the optimal solution of Problem (4).

B. Computation and Transmission Model

In this subsection, the channel model and transmission model are described. The delay of one iteration, which consists of the uplink delay and the downlink delay, will be provided. We consider a block fading channel model, in which the channel between the user and the BS remains unchanged within a time slot of length T_0 but changes independently from one time slot to another. We assume that each BS-user pair transmits independently via frequency domain multiple access (FDMA) [13] in the uplink. In the downlink, the BS makes full use of the bandwidth and broadcasts the parameters to all the involved users. We assume that $\boldsymbol{w}$ is a d_v-dimension

vector and each dimension of it is quantified by q nats. The achievable uplink data rate of user k can be given by[1]

$$r_k = W \ln\left(1 + \frac{P_k |h_k|^2}{\sigma^2}\right), \tag{8}$$

where h_k is the channel coefficient between user k and BS, P_k is the transmission power of user k, σ^2 is the noise power, and W is the bandwidth.

We consider a scenario where the uplink transmission time is much longer than the time slot length T_0. This scenario is practical because the bandwidth and the transmission power are very limited in the Internet of Things (IoT) applications or Industrial Internet applications. Also nowadays the number of machine learning model parameters is increasing incredibly. Thus, the transmission time is long enough and can be uniformly quantized with a relative small error. The time slots that are needed to transmit all those $S = d_v q$ nats at iteration n are given by

$$d_k^{(n)} = \min\left\{ d : \sum_{t=1}^{d} r_k^{(n)(t)} T_0 \geq S, d \in \mathbb{N}_+ \right\}, \tag{9}$$

where $r_k^{(n)(t)}$ is the achievable data rate of user k at tth time slot of iteration n. The remaining parts of this paper will follow with such notation.

Since the BS has to wait for all the involved users to finish their computation and transmission, the total time needed for an uplink transmission (referred as *uplink time*) is given by [2]

$$T_{up}^{(n)} = \max\{t_k : t_k = C_k D_k + d_k^{(n)} T_0, k \in \mathcal{K}\}, \tag{10}$$

where $C_k D_k$ is the computation time that is proportional to the dataset size D_k and C_k is a constant value.

In the downlink, we assume that the BS has to adjust its downlink transmission rate to adapt to the worst one of the users' channels. So the achievable downlink transmission rate is given as

$$r_{down} = W_d \ln\left(1 + \frac{P_{down} h_{down}^2}{\sigma^2}\right), \tag{11}$$

where $h_{down} = \min\{|h_k| : k \in \mathcal{K}\}$, P_{down} is the downlink transmission power, and W_d is the bandwidth for downlink transmission.

Similar to the assumptions in the uplink, the time needed for a downlink transmission (referred as *downlink time*) is given as

$$T_{down}^{(n)} = T_0 \min\left\{ d : \sum_{t=1}^{d} r_{down}^{(n)(t)} T_0 \geq S, d \in \mathbb{N}_+ \right\}. \tag{12}$$

Thus the time needed for one iteration is given as the sum of *uplink time* and *downlink time*

$$T_{ite}^{(n)} = T_{up}^{(n)} + T_{down}^{(n)} \tag{13}$$

Based on the above computation and transmission model, delay and convergence time analysis for this scenario will be provided in Section III.

[1]For the convenience of derivation, we use nats rather than bits as the unit of data.

[2]Actually, in Algorithm 1, each iteration contains two rounds of computation and transmission, one round for $\nabla F_k(\boldsymbol{w}^{(n)})$ and $\nabla F(\boldsymbol{w}^{(n)})$, another round for $\boldsymbol{h}_k^{(n)}$ and $\boldsymbol{w}^{(n+1)}$. Without loss of generality, we consider only one round of computation and transmission for the convenience of analysis.

III. Delay and Convergence Time Analysis

A. Number of Iterations

According to [12], we have Lemma 1, which gives the upper bound of the number of global iteration.

Lemma 1. Assume that $F_k(\boldsymbol{w})$ is L-smooth and γ-strongly convex:

$$\gamma \boldsymbol{I} \preceq \nabla^2 F_k(\boldsymbol{w}) \preceq L \boldsymbol{I}. \tag{14}$$

If we run Algorithm 1 with $0 < \xi \leq \frac{\gamma}{L}$ for

$$n \geq \frac{a}{1-\eta} \triangleq I_0 \tag{15}$$

iterations with $a = \frac{2L^2}{\gamma^2 \eta} \ln \frac{1}{\epsilon_0}$, we have $F(\boldsymbol{w}^{(n)}) - F(\boldsymbol{w}^*) \leq \epsilon_0 (F(\boldsymbol{w}^{(0)}) - F(\boldsymbol{w}^*))$.

Proof: See [12] Appendix A. ∎

However, many machine learning models don't meet the assumptions mentioned in Lemma 1. Actually it is hard for most machine learning models to find the optimal solution. By assuming that n is a random variable, We can instead use empirical distribution to characterize it. It is reasonable to assume that the learning process has been executed in other networks or base stations. The distribution of the number of iterations can be approximated by the empirical distribution from other networks.

B. Transmission Delay of One Iteration

Theorem 1. Given the uplink delay distribution and the downlink delay distribution, the one iteration delay is given as

$$\begin{aligned} &\Pr\{T_{ite}^{(n)} = dT_0\} \\ &= \sum_{i=1}^{d-1} \Pr\{T_{up}^{(n)} = iT_0\} \Pr\{T_{down}^{(n)} = (d-i)T_0\}, \end{aligned} \tag{16}$$

where the uplink delay distribution and the downlink delay distribution are given in Lemma 2 and Lemma 3.

Since the computation time in local update is a deterministic constant for user k, it has no influence on the random distribution of the uplink and downlink delay. Moreover, the computational power of the edge devices is growing rapidly. Therefore, for simplicity, we ignore the computation time in the following discussions.

Lemma 2. The uplink delay distribution is given as

$$\Pr\{T_{up}^{(n)} = dT_0\} = \prod_{k=1}^{K} \Pr\{d_k \leq d\} - \prod_{k=1}^{K} \Pr\{d_k \leq d-1\}, \tag{17}$$

where $\Pr\{d_k = d\} = \Pr\{Z_{d-1} > 0\} - \Pr\{Z_d > 0\}$ and $Z_d = \frac{S}{WT_0} - \sum_{t=1}^{d} \frac{r_k}{W}$. $\Pr\{Z_d > 0\}$ can be approximated through Lugannani-Rice formula:

$$\Pr\{Z_d > 0\} \approx 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\omega} e^{-\frac{u^2}{2}} du + \frac{e^{-\frac{\omega^2}{2}}}{\sqrt{2\pi}} (\frac{1}{\psi} - \frac{1}{\omega}), \quad (18)$$

where $\omega = \text{sign}(s^*)\sqrt{-2K_d(s^*)}$, $\psi = s^*\sqrt{K_d''(s^*)}$, $K_d'(s^*) = 0$, and s^* can be found through binary search. $K_d(s)$ is the cumulant generating function (CGF) of Z_d.

Proof: As shown in (8), r_k is a random variable. We assume h_k follows Rayleigh distribution. In order to analyze the transmission delay $d_k^{(n)}$, firstly we have to obtain the distribution of the delay. According to (9), we have

$$d_k^{(n)} = \left\{ d : \sum_{t=1}^{d} r_k^{(n)(t)} \geq \frac{S}{T_0}, \sum_{t=1}^{d-1} r_k^{(n)(t)} < \frac{S}{T_0}, d \in \mathbb{N}_+ \right\}, \quad (19)$$

$$\Pr\{d_k^{(n)} > d\} = \Pr\left\{ \sum_{t=1}^{d} r_k^{(n)(t)} < \frac{S}{T_0} \right\}. \quad (20)$$

However, it is hard to obtain the expression for the distribution of $\sum_{t=1}^{d} r_k^{(n)(t)}$ by the way of convolution, which makes further analysis difficult. Thus, we apply saddle point approximation [14] [15] to approximate the distribution of $\sum_{t=1}^{d} r_k^{(n)(t)}$. Because $r_k^{(n)(t)}$ is $i.i.d.$ over n and t, we omit n and t, and let $Z_d = \frac{S}{WT_0} - \sum_{t=1}^{d} \frac{r_k}{W}$. Equation (19) and (20) can be rewritten as

$$d_k = \{d : Z_d \leq 0, Z_{d-1} > 0, d \in \mathbb{N}_+\}. \quad (21)$$

$$\Pr\{d_k > d\} = \Pr\{Z_d > 0\}. \quad (22)$$

Then we have

$$\begin{aligned} \Pr\{d_k = d\} &= \Pr\{d_k > d-1\} - P\{d_k > d\} \\ &= \Pr\{Z_{d-1} > 0\} - \Pr\{Z_d > 0\}. \end{aligned} \quad (23)$$

We obtain the CGF of Z_d as

$$\begin{aligned} K_d(s) &= \ln \mathbb{E}\{e^{sZ_d}\} \\ &= \frac{S}{WT_0} s + d \ln \mathbb{E}\{e^{-\frac{sr_k}{W}}\}. \end{aligned} \quad (24)$$

Given that h_k follows Rayleigh distribution, we have the distribution of $\frac{r_k}{W}$ as

$$f(r) = \frac{1}{2\lambda} e^{\frac{1}{2\lambda}} e^{r - \frac{e^r}{2\lambda}} \mathbb{I}\{x > 0\}, \quad (25)$$

where $\lambda = \frac{P_k}{\sigma^2}$ is the Signal to Noise Ratio (SNR), and $\mathbb{I}\{\cdot\}$ is the indicator function. According to the defination of Moment Generating Function (MGF), the MGF of $\frac{r_k}{W}$ is given as

$$\begin{aligned} M(s) &= \int_{-\infty}^{+\infty} e^{sr} f(r) dr \\ &= e^{\frac{1}{2\lambda}} (2\lambda)^s \int_{\frac{1}{2\lambda}}^{+\infty} t^s e^{-t} dt \\ &= e^{\frac{1}{2\lambda}} (2\lambda)^s \Gamma(s+1, \frac{1}{2\lambda}), \end{aligned} \quad (26)$$

where $\Gamma(s,x) \triangleq \int_x^{+\infty} t^{s-1} e^{-t} dt$ is the Incomplete Gamma Function. Thus, (24) can be rewritten as

$$\begin{aligned} K_d(s) &= \frac{S}{WT_0} s + d \ln M(-s) \\ &= \frac{S}{WT_0} s + d \ln \left[e^{\frac{1}{2\lambda}} (2\lambda)^{-s} \Gamma(-s+1, \frac{1}{2\lambda}) \right] \end{aligned} \quad (27)$$

Then we can obtain the first and second derivative of $K_d(s)$

$$K_d'(s) = \frac{S}{WT_0} - d \frac{M'(-s)}{M(-s)}, \quad (28)$$

$$K_d''(s) = d \frac{M''(-s)}{M(-s)} - d \frac{(M'(-s))^2}{(M(-s))^2}, \quad (29)$$

$$\begin{aligned} M'(-s) =& e^{\frac{1}{2\lambda}} (2\lambda)^{-s} \Big[\ln(2\lambda) \Gamma\Big(-s+1, \frac{1}{2\lambda}\Big) \\ &+ \Gamma'\Big(-s+1, \frac{1}{2\lambda}\Big) \Big], \end{aligned} \quad (30)$$

$$\begin{aligned} M''(-s) =& e^{\frac{1}{2\lambda}} (2\lambda)^{-s} \Big[\ln^2(2\lambda) \Gamma\Big(-s+1, \frac{1}{2\lambda}\Big) \\ &+ 2\ln(2\lambda) \Gamma'\Big(-s+1, \frac{1}{2\lambda}\Big) + \Gamma''\Big(-s+1, \frac{1}{2\lambda}\Big) \Big], \end{aligned} \quad (31)$$

$$\begin{aligned} \Gamma'\Big(-s+1, \frac{1}{2\lambda}\Big) =& \Gamma\Big(-s+1, \frac{1}{2\lambda}\Big) \ln\Big(\frac{1}{2\lambda}\Big) \\ &+ \frac{1}{2\lambda} G_{2,3}^{3,0}\left(\begin{matrix} 0, 0 \\ -s, -1, -1 \end{matrix} \middle| \frac{1}{2\lambda} \right), \end{aligned} \quad (32)$$

$$\begin{aligned} \Gamma''\Big(-s+1, \frac{1}{2\lambda}\Big) =& \Gamma\Big(-s+1, \frac{1}{2\lambda}\Big) \ln^2\Big(\frac{1}{2\lambda}\Big) \\ &+ \frac{1}{\lambda} \Bigg[G_{3,4}^{4,0}\left(\begin{matrix} 0, 0, 0 \\ -s, -1, -1, -1 \end{matrix} \middle| \frac{1}{2\lambda} \right) \\ &+ \ln\Big(\frac{1}{2\lambda}\Big) G_{2,3}^{3,0}\left(\begin{matrix} 0, 0 \\ -s, -1, -1 \end{matrix} \middle| \frac{1}{2\lambda} \right) \Bigg]. \end{aligned} \quad (33)$$

$$\begin{aligned} & G_{\rho_3,\rho_4}^{\rho_1,\rho_2}\left(\begin{matrix} a_1, a_2, \ldots, a_{\rho_3} \\ b_1, b_2, \ldots, b_{\rho_4} \end{matrix} \middle| z \right) \\ &= \frac{1}{2\pi i} \oint_{\mathcal{L}} \frac{\prod_{j=1}^{\rho_1} \Gamma(b_j - h) \prod_{j=2}^{\rho_2} \Gamma(1 - a_j + h)}{\prod_{j=\rho_1+1}^{\rho_4} \Gamma(1 - b_j + h) \prod_{j=\rho_2+1}^{\rho_3} \Gamma(a_j - h)} z^h dh \end{aligned} \quad (34)$$

where $G_{\rho_3,\rho_4}^{\rho_1,\rho_2}\left(\begin{matrix} a_1, a_2, \ldots, a_{\rho_3} \\ b_1, b_2, \ldots, b_{\rho_4} \end{matrix} \middle| z \right)$ is the Meijier G-function. Once we get the s^*, according to the Lugannani-Rice formula [15], we can approximate the probability $\Pr\{Z_d > 0\}$ by Equation (18).

Similarly, we can obtain $\Pr\{Z_{d-1} > 0\}$ by the same way. Then we can obtain $\Pr\{d_k = d\}$. As we assume that the computation time can be ignored, the distribution of $T_{up}^{(n)}$ is given by Lemma 1 and can be calculated numerically. ■

Lemma 3. The downlink delay distribution is given as

$$\Pr\{T_{down}^{(n)} = dT_0\} \tag{35}$$
$$= \Pr\{\sum_{t=1}^{d} r_{down}^{(n)(t)} T_0 \geq S\} - \Pr\{\sum_{t=1}^{d-1} r_{down}^{(n)(t)} T_0 \geq S\},$$

where $\Pr\{\sum_{t=1}^{d} r_{down}^{(n)(t)} T_0 \geq S\}$ and $\Pr\{\sum_{t=1}^{d-1} r_{down}^{(n)(t)} T_0 \geq S\}$ can be obtained in a similar way as Lemma 1.

Proof: Similarly, the distribution of $T_{down}^{(n)}$ can be obtained. More specifically, the difference lies in that the distribution of h_{down} follows $f_{down}(h) = Khe^{-\frac{Kh^2}{2}}\mathbb{I}\{x > 0\}$ since $h_{down} = \min\{|h_k| : k \in \mathcal{K}\}$. The distribution of $\frac{r_{down}}{W_d}$ is given as follow

$$f_{down}(r) = \frac{K}{2\lambda_d} e^{\frac{K}{2\lambda_d}} e^{r - \frac{Ke^r}{2\lambda_d}}, \tag{36}$$

where $\lambda_d = \frac{P_{down}}{\sigma^2}$ is the downlink SNR. The MGF of $\frac{r_{down}}{W_d}$ is given as follow

$$M_{down}(s) = e^{\frac{K}{2\lambda_d}} (\frac{2\lambda_d}{K})^s \Gamma(s+1, \frac{K}{2\lambda_d}). \tag{37}$$

Further results are similar to the results of uplink. Thus, they are omitted due to the limitation of layout. ■

C. Overall Delay

Corollary 1. The time needed for the convergence of the training process is less than the sum of I_0 iteration delays in the stochastic order

$$T_c \leq_{st} \sum_{i=1}^{I_0} T_{ite}^{(i)}, \tag{38}$$

where T_c is a random variable representing the time needed for the convergence of the training process, and $\sum_{i=1}^{I_0} T_{ite}^{(i)}$ is a random variable representing the sum of I_0 iteration delays. The distribution of $\sum_{i=1}^{I_0} T_{ite}^{(i)}$ can be calculated through generating function $G_c(z) = G_{ite}^{I_0}(z)$.

Proof: Since $T_c = \sum_{i=1}^{n} T_{ite}^{(i)}$ and $n \leq I_0$, we have $\Pr\{T_c > t\} \leq \Pr\{\sum_{i=1}^{I_0} T_{ite}^{(i)} > t\}$ for all $t > 0$. ■

Besides, as mentioned in Section III-A, if we can get the empirical distribution of the number of global iteration, the generating function is given as $G_c(z) = H(G_{ite}(z))$, where $H(z)$ is the generating function of the empirical distribution of the number of global iteration n. We can calculate the distribution of convergence time through its generating function. For a discrete random variable X, its corresponding generating function is defined as

$$G(z) = E(z^X) = \sum_{x=0}^{\infty} p(x) z^x, \tag{39}$$

where $p(x)$ is the probability distribution function of X. Given the generating function, the probability distribution function can be obtained as

$$p(k) = \Pr\{X = k\} = \frac{G^{(k)}(0)}{k!}, \tag{40}$$

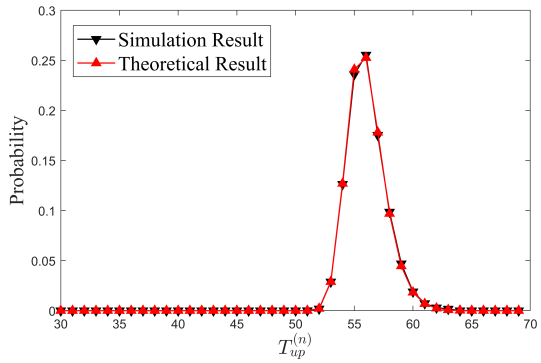

Fig. 2. The simulation and theoretical results of the uplink delay distribution.

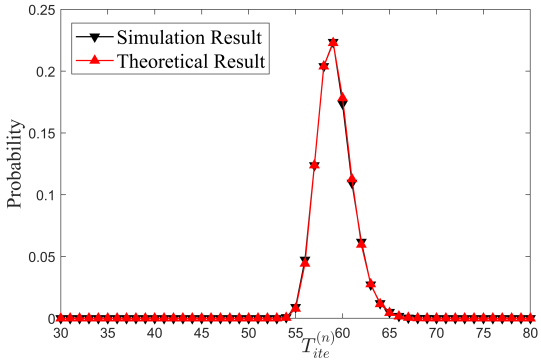

Fig. 3. The simulation and theoretical results of the one iteration delay distribution.

where $G^{(k)}(z)$ is the k-th derivative of $G(z)$.

IV. Experiment Results

A. Experiment Setting

For our simulations, we use Support Vector Machine (SVM) as the classification model, and handwritten digit database (MNIST) as the dataset. Model parameters are quantized into bit sequence with 8-bit per parameter for transmission. The quantized size of the corresponding model parameters are 32080 bits. The uplink bandwidth W is 100 KHz, and the downlink bandwidth W_d is 3 MHz. The coherence time T_0 is 2.5 ms. The number of users is given as $K = 30$. The SNR λ and λ_d are set to 10 dB and 20 dB. The channel coefficient following Rayleigh distribution with an expectation of $\sqrt{\frac{\pi}{2}}$. The dataset D_k of each user k follows the same distribution and has the same size, which is 2000 samples.

B. Experiment Results

Firstly, we run simulations on the uplink delay mentioned in Theorem 1. As shown in Fig. 2, almost all of the simulation results lie above the theoretical expected value although they are very close. This phenomenon is reasonable because theoretically the transmission delay can be infinite, but we neglect the probability that the transmission delay goes too big since the probability is almost zero, for the convenience of computation. Further, we compare the empirical distribution of one iteration delay from our simulation with the theoretical

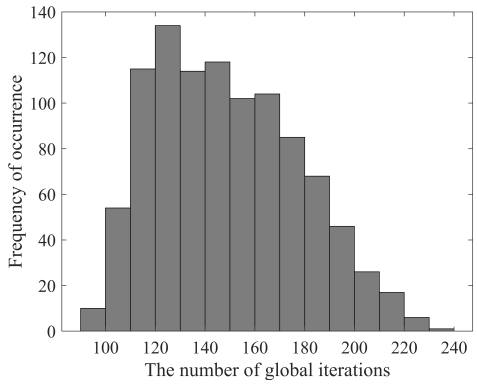

Fig. 4. The empirical distribution of the number of global iteration.

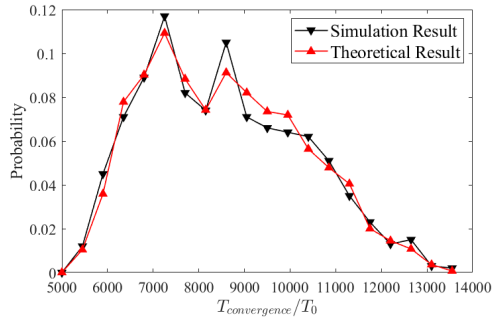

Fig. 5. The simulation result and theoretical result of the distribution of convergence time.

distribution in Fig. 3. As shown in Fig. 3, the empirical distribution and the theoretical distribution are almost identical. The above results prove that the proposed analysis of one iteration delay is correct.

We run 1000 times of SVM learning process to obtain the empirical distribution of the number of global iteration. As shown in Fig. 4, almost all the numbers are less than 240 and the distribution is unimodal, which make it practical to calculate the corresponding generating function. Some of the recognition results are shown in Fig. 6, where the number below the figure picture is the prediction result with red standing for wrong prediction. This accuracy of FL is almost the same with the accuracy of centralized training. Since the dataset of each user follows a independently identically distribution and has the same dataset size, it is reasonable to have similar results for FL and centralized learning. Furthermore, we verify the theoretical analysis about the convergence time with more simulation results. As shown in Fig. 5, although the difference between the theoretical result and the simulation result is slightly more evident compared with Fig. 2 and Fig. 3, they are still in good agreement with each other.

V. Conclusion

FL becomes a promising solution to enable fast and accurate intelligence distillation at edge with data privacy guarantees. Most of those existing researches focus on analysing the convergence rate, reducing the transmission overheads, or improving the prediction performance of the federated

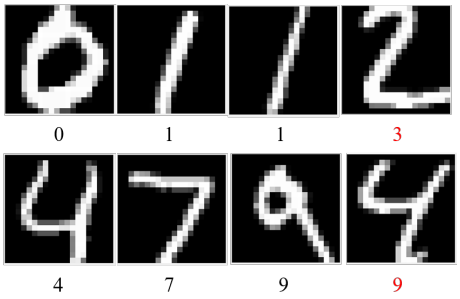

Fig. 6. Several recognition results.

learning model. In this paper, we presented a FL transmission model to analyse the delay distribution in wireless FL systems. We derived the theoretical results for uplink transmission delay and the downlink transmission delay of one iteration by means of saddle point approximation and Lugannani-Rice formula. The distributions of those delays can be calculated numerically. Experiments showed that the simulation results are in good agreement with the theoretical results in terms of the mean value and the distribution of the iteration delay, which verifies the accuracy of our theoretical analysis.

References

[1] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2167–2191, 4th Quart., 2020.

[2] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

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

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

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

[6] K. Yang, Y. Shi, Y. Zhou, Z. Yang, L. Fu, and W. Chen, "Federated machine learning for intelligent iot via reconfigurable intelligent surface," *IEEE Netw.*, vol. 34, no. 5, pp. 16–22, Oct. 2020.

[7] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, 2021.

[8] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 2128–2142, Mar. 2020.

[9] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1135-1143.

[10] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[11] 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.*, pp. 1–1, Nov. 2020.

[12] 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, Jan. 2021.

[13] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier FDMA for uplink wireless transmission," *IEEE Veh. Technol. Mag.*, vol. 1, no. 3, pp. 30–38, Sept. 2006.

[14] R. W. Butler, *Saddlepoint approximations with applications*. Cambridge University Press, 2007, vol. 22.

[15] R. Lugannani and S. Rice, "Saddle point approximation for the distribution of the sum of independent random variables," *Advances in applied probability*, vol. 12, no. 2, pp. 475–490, 1980.