This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3128559, IEEE Transactions on Vehicular Technology

System Optimization of Federated Learning Networks with A Constrained Latency

Zichao Zhao, Junjuan Xia, Lisheng Fan, Xianfu Lei, George K. Karagiannidis, and Arumugam Nallanathan

***Abstract*—This paper investigates a wireless federated learning (FL) network with limited communication bandwidth, where multiple mobile clients train their individual models with the help of one central server. We consider the practical communication scenarios, where the clients should complete the local computation and model upload within a defined latency. By jointly exploiting the dynamic characteristics of wireless channels and computational capability at the clients, we optimize the federated learning network by maximizing the number of active clients under the constraints of both latency and bandwidth. Specifically, we propose two bandwidth allocation (BA) schemes, where *scheme I* is based on the instantaneous channel state information (CSI), while *scheme II* employs the particle swarm optimization (PSO) method, based on the statistical CSI. Simulation results on the test accuracy and convergence rate are finally provided to demonstrate the advantages of the proposed optimization schemes for the considered FL network.**

***Index Terms*—Federated learning, bandwidth allocation, latency, convergence rate.**

I. Introduction

In recent years, there has been a great progress in the development of artificial intelligence (AI), which has found a lot of applications in practice [1]–[3]. The conventional AI algorithms often require centralized processing, which needs to collect the local data of all users. This is however harmful to privacy protection and causes a heavy burden on the system implementation such as the communication and computational cost. In practice, the local dataset of a single client is often insufficient to train a high-performance model. Moreover, due to the preference of clients, the local dataset is unbalanced and non-independent and identically distributed (Non-IID), which is challenging for the clients to train models with some generalization ability by using their own dataset. To alleviate these issues, the framework of federated learning (FL) has been proposed to train the model parameters without collecting data from users, in which only the model parameters of the users are collected and aggregated at the server. Moreover, in order to further reduce the communication cost and accelerate the process of federated learning, fraction clients are selected to participate in federated learning instead of all clients in each communication round. The practical scenarios of FL include the visit of website and everything the clients type on their mobile keyboards, such as password, message, and online shopping. Overall, there exists some performance loss of FL compared to the centralized, due to distributed learning, a fractional clients and unbalanced dataset.

One major challenge of FL is the convergence rate, which determines the communication rounds that FL needs to converge. The convergence rate of FL is affected by the number of active clients, who successfully upload model in each communication round, which is constrained by the limited communication resources in practice [4]. To reduce the number of rounds and speed up the convergence, the authors in [5] adopted momentum gradient descent to optimize the loss function of the local update, where the rate of gradient descent and convergence was accelerated. Moreover, the adaptive quantization and specification could be applied into the FL networks, in order to compress the local models to reduce the communication cost [6]. In further, some other wireless techniques such as mobile edge computing can be incorporated into the FL networks to reduce the communication cost, in order to accelerate the convergence [7]–[10]. Recently, the effect of latency on the FL networks has been studied, where several bandwidth allocation schemes were proposed to enhance the system performance [11], [12]. However, the channel state information is seldom incorporated into the system design of FL networks in the existing works, which motivates the work in this paper.

This paper studies a wireless FL network with limited communication bandwidth, where multiple clients complete the local computation and model upload under the constraint of latency, in order to accelerate the FL process. To improve the system performance, two optimization schemes are devised to maximize the number of active clients by performing the bandwidth allocation (BA) among clients, based on the channel state information (CSI). Specifically, *scheme I* is based on the instantaneous CSI (I-CSI) to maximize the number of active clients during the FL process, while *scheme II* is based on the statistical CSI (S-CSI) and it uses the particle swarm optimization (PSO) to maximize the expectation of the number

Z. Zhao, J. Xia and L. Fan are all with the School of Computer Science, Guangzhou University, Guangzhou, China (e-mail: {xiajunjuan,lsfan}@gzhu.edu.cn).

X. Lei is with the Provincial Key Lab of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 610031, China, and also with National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: xflei@swjtu.edu.cn).

G. K. Karagiannidis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki 54636, Greece (e-mail: geokarag@auth.gr).

A. Nallanathan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London,London, U.K (e-mail: a.nallanathan@qmul.ac.uk).

This work was supported by the NSFC (Nos. 61871139/62101145), by the Natural Science Foundation of Guangdong Province (No. 2021A1515011392), and by the research program of Guangzhou University (No. YJ2021003). The work of X. Lei was supported by the NSFC (No. 61971360), the Fundamental Research Funds for the Central Universities (No. XJ2021KJZK007), and the open research fund of National Mobile Communications Research Laboratory, Southeast University (No. 2021D05).

The corresponding author of this paper is Lisheng Fan.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3128559, IEEE Transactions on Vehicular Technology

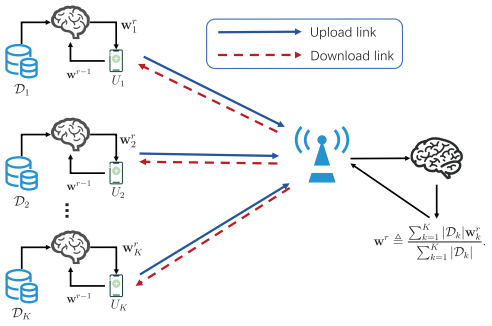


Fig. 1. System model of federated learning networks.

of active clients. Simulation results are finally provided to demonstrate the advantages of the proposed optimization schemes in the system performance for the considered FL network.

The remainder of this paper is organized as follows. After the Introduction, Section II describes the system model of the considered FL network under the constraints from the latency and communication bandwidth. Then, Section III presents two BA schemes based on CSI to optimize the system performance. In further, simulation results are provided in Section IV to demonstrate the effectiveness of the proposed optimization schemes for the considered network, followed by the conclusions made in Section V.

II. System Model

Fig. 1 shows the system model of federated learning networks, where there are K mobile clients $\{U_k|1 \leq k \leq K\}$ trying to aggregate their training models through the help of one central server $\mathcal{S}$. Client U_k has a local trainable dataset $\mathcal{D}_k$, and the number of samples in $\mathcal{D}_k$ is $|\mathcal{D}_k|$, so that the total number of samples of all clients is $|\mathcal{D}| = \sum_{k=1}^{K} |\mathcal{D}_k|$. In practice, the local dataset is limited, and it is hard for each client to obtain a fine training model of deep network, without using the datasets of other clients. However, directly using the datasets of other clients will impose a severe load on the communication and computation, and the severe issue of information leakage is also caused. To solve these problems, the FL framework is employed in Fig. 1, where each client only needs to upload the trained model parameters to the central server $\mathcal{S}$, which reduces the communication cost significantly.

The FL procedure is detailed as follows. The central server $\mathcal{S}$ firstly randomly selects N clients among K ones for efficiency and fairness at each round [4], and then it broadcasts the global model to the clients. After that, each client updates the local model by using the global model and local dataset. These N clients further upload the local models to the central server $\mathcal{S}$ for aggregating models to obtain the global model. This process is iterated R rounds, until a fine deep network model is obtained.

Specifically, at round r, the n-th client receives the global model of the previous round $\mathbf{w}^{r-1}$. In general, the downlink bandwidth is much larger than the uplink bandwidth, and the server has a larger transmit power than the mobile clients because of the connection with the power supply. Hence, for client U_n, the time to receive the global model parameter can be negligible. After μ_n times of the local epochs, client U_n updates the local mode by using the received global model and local dataset, and the local model is updated as

$$\mathbf{w}_n^r = \mathbf{w}^{r-1} - \eta \nabla F_n(\mathbf{w}^{r-1}), \tag{1}$$

where $\mathbf{w}_n^r$ denotes the model parameter of client U_n and $F_n(\bullet)$ represents the loss function of client U_n. Notations η and ∇ represent the learning rate and gradient operation on the loss function, respectively. The local training time of client U_n is t_n^e, given by

$$t_n^e = \frac{\rho \mu_n |\mathcal{D}_n|}{f_n}, \tag{2}$$

where ρ is the number of CPU cycles required to compute one sample data and μ_n is the epoch of local training of client n in a communication round. Moreover, $f_n \sim \mathcal{U}(f_{min}, f_{max})$ is the computational capacity of client U_n, where $\mathcal{U}(\bullet)$ represents a uniform distribution, in which f_{min} and f_{max} are the minimum and maximum computational capacities, respectively. After the training, the local model is uploaded to the central server, and the transmission time of client U_n is given by,

$$t_n^u = \frac{L_n}{C_n}, \tag{3}$$

where L_n is the size of the model parameter of client U_n, in which the 32-bit floating point format is often used in practice, and C_n is the transmission rate from client U_n to the server $\mathcal{S}$, given by

$$C_n = B_n \log_2 \left(1 + \frac{P_n |h_n|^2}{\sigma_n^2}\right), \tag{4}$$

where P_n is the transmit power and σ_n^2 is the variance of the additive Gaussian white noise (AWGN) at the receiver. The channel parameter h_n experiences Rayleigh flat fading with the average channel gain of ε_n, and B_n is the wireless channel bandwidth allocated to client U_n. In practice, the frequency spectrum is limited, and the channel bandwidth of the N clients should meet the following constraint,

$$\sum_{n=1}^{N} B_n \leq B_{total}, \tag{5}$$

where B_{total} is the total bandwidth of the system. The server $\mathcal{S}$ further aggregates the collected model parameters and gets the global model. We use the federated average (FedAvg), and then the global model parameter $\mathbf{w}^r$ at round r can be updated as

$$\mathbf{w}^r \triangleq \frac{\sum_{n=1}^{N} |\mathcal{D}_n| \mathbf{w}_n^r}{\sum_{n=1}^{N} |\mathcal{D}_n|}. \tag{6}$$

From the above equations, we can summarize the total latency of client U_n at each round as

$$t_n = t_n^e + t_n^u. \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3128559, IEEE Transactions on Vehicular Technology

III. System Optimization

In practice, the clients may have different computational capabilities and experience different fading channels, causing different latencies of uploading the local model to the server. This may increase the system latency of global aggregation at the server and deteriorate the aggregation performance. To speed up the global aggregation, a latency threshold γ_{th} should be set. Specifically, client U_n is able to complete the upload if its latency is below γ_{th}, i.e.,

$$t_n \leq \gamma_{th}. \tag{8}$$

By incorporating this latency constraint, we can optimize the FL networks through allocating the limited frequency spectrum among N clients, in order to minimize the global loss function,

$$\begin{aligned} \min_{\{B_1,\ldots,B_N\}} \quad & F(\mathbf{w}) \\ s.t. \quad & t_n \leq \gamma_{th}, \forall n \in \mathcal{N}, \\ & \sum_{n\in\mathcal{N}} B_n \leq B_{total}, \end{aligned} \tag{9}$$

where $\mathcal{N}$ is the set of N clients, and $F(\bullet)$ denotes the global loss function. As different tasks and datasets may use different loss functions, the optimization problem in (9) is often not general. Inspired by the fact that uploading more models successfully in each round can help improve the FL performance [13], we turn to optimize the FL networks by maximizing the number of active clients who can successfully upload the models, given by

$$\begin{aligned} \max_{\{B_1,\ldots,B_N\}} \quad & N_1 \\ s.t. \quad & t_n \leq \gamma_{th}, \forall n \in \mathcal{N}, \\ & \sum_{n\in\mathcal{N}} B_n \leq B_{total}, \end{aligned} \tag{10}$$

where N_1 is the number of active clients, who can successfully upload the models. In the following, we will present two BA schemes to solve the optimization problem in (10).

A. I-CSI based BA Scheme

When the instantaneous CSI of N users is estimated[1] and gathered at each round of federated learning, we can solve the BA optimization problem in (10) based on the instantaneous CSI. Note that the I-CSI based BA scheme can be applied to the wireless channels with static fading, such as the application scenarios of static Internet of Things (IoT) networks. To train the global model better, more clients should participate into uploading models in each communication round. Aimed by this, we propose a sorting BA scheme, where the clients with favorable channel conditions tend to be allocated an appropriate bandwidth preferentially. Specifically, we firstly sort the N clients in a descending order according to the channel condition, which forms a set $\mathcal{N}$. In $\mathcal{N}$, the former clients have better channel condition than the latter. After that, the clients in $\mathcal{N}$ are selected one by one from the first to the last, and then they are allocated an appropriate bandwidth, in order to meet the latency requirement. In particular, for client U_n, its bandwidth allocation should meet the following requirement,

$$t_n = \frac{L_n}{B_n \log_2\left(1+\frac{P_n|h_n|^2}{\sigma_n^2}\right)} + \frac{\rho\mu_n|\mathcal{D}_n|}{f_n} \leq \gamma_{th}, \tag{11}$$

which results in

$$B_n \geq \frac{L_n}{\left(\gamma_{th} - \frac{\rho\mu_n|\mathcal{D}_n|}{f_n}\right)\log_2\left(1+\frac{P_n|h_n|^2}{\sigma_n^2}\right)}. \tag{12}$$

Then, we can set B_n as

$$B_n = \frac{L_n}{\left(\gamma_{th} - \frac{\rho\mu_n|\mathcal{D}_n|}{f_n}\right)\log_2\left(1+\frac{P_n|h_n|^2}{\sigma_n^2}\right)}, \tag{13}$$

if there is enough bandwidth resource left. This process continues until the clients in $\mathcal{N}$ have been completely allocated or the bandwidth resource has been used up[2]. The whole procedure of the proposed I-CSI based BA scheme is summarized in **Algorithm** 1, where $|\mathcal{N}|$ is the cardinality of set $\mathcal{N}$.

Algorithm 1: I-CSI Based BA Scheme

```
for Each r ∈ [1, R] do
    N ← Sort the N clients according to the channel gains in Des.;
    for n = 1:|N| do
        Obtain B_n according to (13);
        Get the rest bandwidth: B_total ← B_total − B_n;
        if B_total ≤ 0 then
            Break;
        end if
        Add element U_n to N_1;
    end for
end for
```

B. S-CSI Based BA Scheme

As the above I-CSI based BA scheme requires to know the instantaneous CSI of all users at each round, a severe burden is imposed on the system implementation. To alleviate this burden, we turn to exploit the statistics CSI to solve the BA problem in (10). Note that the S-CSI has to be used to the wireless channels with fast fading, such as the application scenarios of Internet of Vehicles (IoV) networks. From (11), we firstly write the channel conditions with the given latency threshold γ_{th} as

$$|h_n|^2 \geq \left(2^{\frac{L_n}{B_n\left(\gamma_{th}-\frac{\rho\mu_n|\mathcal{D}_n|}{f_n}\right)}} - 1\right)\frac{\sigma_n^2}{P_n} = G(f_n), \tag{14}$$

[1]The clients can estimate their channel parameters of the wireless links, by using the pilot signals transmitted from the central server. Then, the central server can gather these CSIs of clients through some dedicated feedback channels.

[2]Note that if the total bandwidth is sufficient, the excess bandwidth can be allocated to the clients according to some criteria, such as uniform allocation among clients or allocating more excess bandwidth to the clients with worse channel condition. This will not affect the number of active clients, as the latency constraint has already been satisfied.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3128559, IEEE Transactions on Vehicular Technology

where $G(x)$ is

$$G(x) = \left(2^{\frac{L_n}{B_n\left(\gamma_{th} - \frac{\rho\mu_n|\mathcal{D}_n|}{x}\right)}} - 1\right)\frac{\sigma_n^2}{P_n}.$$

As the channels in the network are subject to Rayleigh fading, $|h_n|^2$ follows the exponential distribution with the average gain of ε_n. In this case, we turn to maximize the expectation of number of active clients participating into uploading models to the server. Aimed by this, we firstly calculate the probability that each client can successfully upload its model to the server, which satisfies the latency requirement. From the probability density function (PDF) of $|h_n|^2$, we can obtain the conditional expectation of the n-th client participating into uploading its model to the server as,

$$\begin{aligned}\mathbb{E}(x_n|f_n) &= \Pr\{|h_n|^2 \geq G(f_n)|f_n\} \\ &= \begin{cases} \exp\big(-\frac{G(f_n)}{\varepsilon_n}\big), & f_n > \frac{\rho\mu_n|\mathcal{D}_n|}{\gamma_{th}} \\ 0, & f_n \leq \frac{\rho\mu_n|\mathcal{D}_n|}{\gamma_{th}} \end{cases}. \end{aligned} \quad (15)$$

From $\mathbb{E}(x_n|f_n)$ and $f_n \sim \mathcal{U}(f_{min}, f_{max})$, we can write the expectation $\mathbb{E}(x_n)$ as

$$\begin{aligned}\mathbb{E}(x_n) &= \int_{f_{min}}^{f_{max}} \frac{1}{f_{max} - f_{min}} \Pr\{|h_n|^2 \geq G(f_n)|f_n\} df_n \\ &\approx \sum_{m=1}^{M} \frac{\sqrt{1-\phi_m^2}}{f_{max} - f_{min}} \exp\left(-\frac{G(\zeta_m)}{\varepsilon_n}\right), \end{aligned} \quad (16)$$

where the Gaussian-Chebyshev approximation [14] is used and M is a complexity-vs-accuracy tradeoff parameter with

$$\phi_m = \cos\left(\frac{(2m-1)\pi}{2M}\right),$$

$$\begin{aligned}\zeta_m = &\frac{f_{max} - \max\left(f_{min}, \frac{\rho\mu_n|\mathcal{D}_n|}{\gamma_{th}}\right)}{2}\phi_m \\ &+ \frac{f_{max} + \max\left(f_{min}, \frac{\rho\mu_n|\mathcal{D}_n|}{\gamma_{th}}\right)}{2}. \end{aligned}$$

Note that the Gaussian-Chebyshev approximation can be accurate with a medium value of M. From $\mathbb{E}(x_n)$, we can calculate the expectation of the number of active clients who can successfully upload model parameters, given by

$$\begin{aligned}\mathbb{E}(X) &= \sum_{n=1}^{N} \mathbb{E}(x_n), \\ &\approx \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{\sqrt{1-\phi_m^2}}{f_{max} - f_{min}} \exp\left(-\frac{G(\zeta_m)}{\varepsilon}\right). \end{aligned} \quad (17)$$

The BA scheme is then devised to maximize the expectation $\mathbb{E}(X)$,

$$\begin{aligned}&\max_{\{B_1,\ldots,B_N\}} \mathbb{E}(X) \\ &s.t. \quad \sum_{n=1}^{N} B_n \leq B_{total}. \end{aligned} \quad (18)$$

As it is difficult to directly solve the optimization problem in (18), we turn to employ the particle swarm optimization (PSO) algorithm to solve the optimization, which is an intelligent algorithm based on population. In the PSO algorithm, there are J particles in the population, and each particle includes two important attributes of position and velocity. We use $\boldsymbol{p}_j$ and $\boldsymbol{v}_j$ to represent the position and velocity of particle j, respectively, where $\boldsymbol{p}_j = \{B_1, B_2, \ldots, B_N\}$ provides a feasible solution of the bandwidth allocation problem in (18) and $\boldsymbol{v}_j = \{\Delta B_1, \Delta B_2, \ldots, \Delta B_N\}$ represents the increment of $\boldsymbol{p}_j$. Here, ΔB_n is the increment of B_n from the current iteration to the next one. Moreover, $\boldsymbol{pbest}_j$ and $\boldsymbol{gbest}$ are used to denote the best BA solutions of particle j and the global particles until the current iteration, respectively, which are measured by the fitness function. Here, the fitness function of the PSO is characterized by $\mathbb{E}(X)$. At iteration i, the velocity of the j-th particle, $\boldsymbol{v}_j^i$, is updated by

$$\begin{aligned}\boldsymbol{v}_j^i = \omega\boldsymbol{v}_j^{i-1} &+ c_1\xi_1(\boldsymbol{pbest}_j^{i-1} - \boldsymbol{p}_j^{i-1}) \\ &+ c_2\xi_2(\boldsymbol{gbest}^{i-1} - \boldsymbol{p}_j^{i-1}), \end{aligned} \quad (19)$$

where c_1 and c_2 are two acceleration constants, ξ_1 and ξ_2 are two random variables uniformly distributed in the range of [0, 1], and ω stands for the inertia weight factor. From (19), the position $\boldsymbol{p}_j^i$ is updated by

$$\boldsymbol{p}_j^i = \boldsymbol{p}_j^{i-1} + \boldsymbol{v}_j^i. \quad (20)$$

The particles require I times of iteration to update theirs velocity and position according to (19) and (20), respectively. After I iterations, the $\boldsymbol{gbest}$ will be obtained among J particles, which serves as the solution of the BA optimization problem in (18). Moreover, for J particles and I iterations in the PSO algorithm, the associated computational complexity is about $\mathcal{O}(J \times I)$, where the performance of the PSO can be improved with increased numbers of particles and iterations. The proposed S-CSI based BA scheme is summarized in **Algorithm** 2.

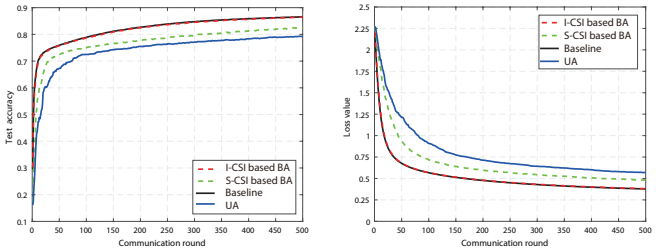
IV. Experimental Results and Discussions

In this section, we present some experimental results to validate the proposed BA schemes for the FL networks, where Python 3.6 is used and the learning framework is PyTorch 1.8.0. Specifically, the total communication round is set to 500, the total number of clients is 200, and the number of selected clients is set to 10. If not specified, the system total communication bandwidth is set to 50MHz, and the channels in the network experience Rayleigh flat fading, where the average channel gain of the k-client to the server is set to $\varepsilon_k = (k + 50)/200$, without loss of generality. Moreover, the transmit power of clients is set to 0.5W, and ρ is set to 10^4 cycle/sample. In further, the computational capability of clients is uniformly distributed in the range of $[2 \times 10^6, 3 \times 10^6]$ cycle/second. Furthermore, for the PSO method involving in the S-CSI based BA scheme, the number of particle in the population is 30, where the number of iterations is 20. Moreover, the two acceleration constants c_1 and c_2 are both set to 0.4 and $\omega = 0.5$. In addition, two typical datasets of MNIST and Fashion-MNIST are used to train the models to validate the proposed studies, detailed as follows,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3128559, IEEE Transactions on Vehicular Technology

Algorithm 2: S-CSI Based BA Scheme.

Obtain the channel condition of each client that meets the latency requirement in (14);
Obtain the conditional expectation of each client participating into uploading its model to the server by (15);
Obtain the expectation of the number of active clients by (17);
Create J particles randomly;
for Each $i \in 1:I$ **do**
  for Each $j \in 1:J$ **do**
    Update $\boldsymbol{v}_j^i$ by (19), and update $\boldsymbol{p}_j^i$ by (20);
    Evaluate $\boldsymbol{p}_j^i$ by the fitness function $\mathbb{E}(X)$;
    if the fitness evaluation of $\boldsymbol{p}_j^i$ is better than that of $\boldsymbol{pbest}_j^i$ **then**
      $\boldsymbol{pbest}_j^i$ is equal to $\boldsymbol{p}_j^i$;
    end if
    if the fitness evaluation of $\boldsymbol{p}_j^i$ is better than that of $\boldsymbol{gbest}^i$ **then**
      $\boldsymbol{gbest}^i$ is equal to $\boldsymbol{p}_j^i$;
    end if
  end for
end for



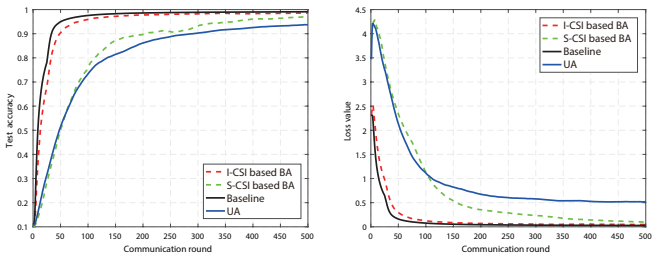
(a) Accuracy (b) Loss

Fig. 2. Performance comparison of several BA schemes with Fashion-MNIST and latency threshold $\gamma_{th} = 5$s.

1) Fashion-MNIST: In this dataset, there are 60000 samples in total. The trained model is composed of two 3×3 convolution layers with ReLU activation, two fully connected layers (600 and 120 units, respectively), a dropout layer between the first fully connected layer and the second one and an output layer with Softmax. The first convolution layer has 43 channels while the second one has 64 channels, both followed by a batch normalization layer and a 2×2 max pooling layer. Moreover, the training parameters are set as follows: the learning rate is 0.001, the batch size is 30, and the local epochs is set to 3 for the selected clients. In addition, the loss function is CrossEntropyLoss and tje optimizer is SGD. In further, we preform Non-IID operation on the dataset [4].

2) MNIST: In this dataset, there are 60000 samples in total. The deep learning model structure consists of the following components: two 5×5 convolution layers of 32 channels and 64 channels, respectively, both followed by the 2×2 max pooling, a fully connected layer with activation function ReLU and 512 units, and an output layer with Softmax. The learning rate is set to 0.065, while the other training parameters remain the same as those in the MNIST.

Figs. 2-3 demonstrate the accuracy and loss of several



(a) Accuracy (b) Loss

Fig. 3. Performance comparison of several BA schemes with MNIST and latency threshold $\gamma_{th} = 3$s.

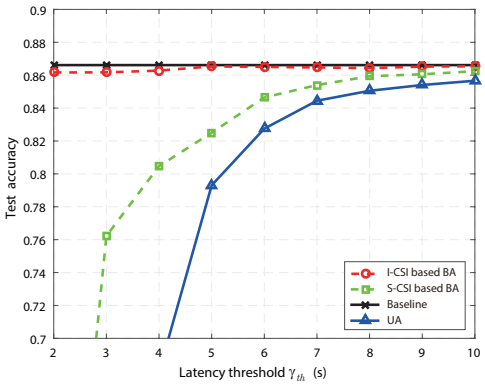


Fig. 4. Effect of latency threshold on the proposed BA schemes with Fashion-MNIST.

BA schemes versus the communication round, where Fig. 2 and Fig. 3 correspond to Fashion-MNIST with $\gamma_{th} = 5$s and MNIST with $\gamma_{th} = 3$s, respectively. For comparison, we also provide the results of 'Baseline' scheme where the latency threshold is set to infinite so that all clients can successfully upload the models to the server, and 'UA' scheme where the communication bandwidth is uniformly allocated among clients. We can observe from these two figures that the accuracy and loss of several BA schemes become convergent with the increasing number of communication round. Moreover, the proposed I-CSI and S-CSI based BA schemes outperform the UA one, since the channel information is incorporated into the BA process. In further, the I-CSI based BA scheme is superior to the S-CSI based one, and it can achieve almost the same performance as the baseline one, since the instantaneous channel state information is effectively exploited to help optimize the BA process.

Fig. 4 shows the effect of the latency threshold γ_{th} on the test accuracy of the proposed BA schemes, where the dataset Fashion-MNIST is used and γ_{th} varies from 2s to 10s. From Fig. 4, we can find that the performances of the proposed two BA schemes and UA improve with a larger γ_{th}, as the clients can successfully upload the models to the server more easier. Moreover, the proposed I-CSI based BA scheme can achieve almost the same performance as the baseline one, for a wide range of latency threshold. In further, although the S-CSI based BA scheme fails to obtain the optimal performance of the baseline scheme, it is still superior to the UA scheme. Furthermore, compared to the I-CSI based BA scheme, the S-CSI based one deteriorates much more rapidly with respect to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3128559, IEEE Transactions on Vehicular Technology

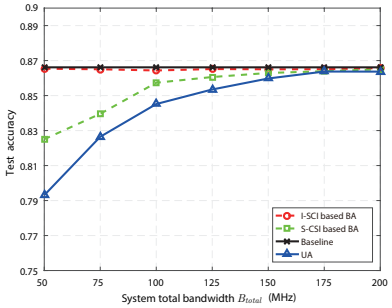
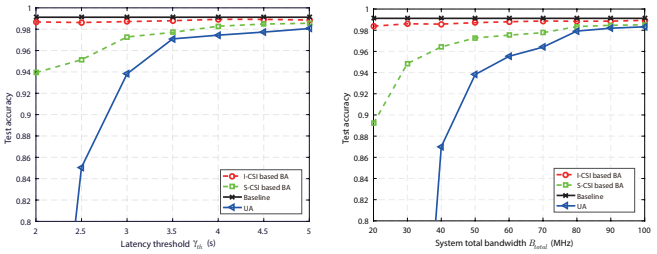


Fig. 5. Impact of total communication bandwidth on the proposed BA schemes with Fashion-MNIST.



(a) Latency threshold (b) Total bandwidth

Fig. 6. Impact of latency threshold and total communication bandwidth on the proposed BA schemes with MNIST.

the decreased latency threshold, as the probability that clients can successfully upload model is reduced severely in the low region of γ_{th}. These phenomena validate the proposed two BA schemes.

Fig. 5 depicts the impact of total communication bandwidth on the test accuracy of the proposed BA schemes, where the dataset Fashion-MNIST is used, γ_{th} is 5s, and the bandwidth B_{total} varies from 50MHz to 200MHz. From this figure, we can see that the test accuracy of the proposed two BA schemes and UA improves with the increasing value of B_{total}, as the transmission rate becomes larger and accordingly the transmission latency decreases. Moreover, the proposed I-CSI based BA scheme can achieve almost the same performance as the baseline one for a wide range of bandwidth, while the S-CSI based BA scheme is superior to the UA one, especially in the low region of bandwidth. In further, compared to the I-CSI based BA scheme, the S-CSI based one deteriorates much more rapidly with the smaller bandwidth, since it becomes more difficult for the clients to complete the model upload due to the decreased transmission data rate in the S-CSI based BA scheme. These phenomena further demonstrate the effectiveness of the proposed two BA schemes.

Fig. 6 demonstrates the impact of latency threshold and total communication bandwidth on the test accuracy of different BA schemes, where the dataset is MNIST. Specifically, Fig. 6(a) corresponds to the performance versus the latency threshold with $B_{total} = 50$MHz, while Fig. 6(b) corresponds to the performance versus the communication bandwidth with $\gamma_{th} = 3$s. From this figure, we can find that with the MNIST dataset, the proposed two schemes are still superior to the UA one, and the proposed I-CSI scheme can achieve almost the optimal performance as the baseline one, in the high region of bandwidth or latency threshold. Moreover, the performance of the two proposed schemes and UA scheme improves with a larger B_{total} or γ_{th}, as the clients have more opportunities to complete the local training and model upload. Overall, the phenomena in Fig. 6 demonstrate the effectiveness of the proposed two BA schemes furthermore.

V. Conclusions

This paper investigated the wireless federated learning network constrained by a latency, where the clients should complete the local computation and model upload under the latency constraint, in order to accelerate the federated learning process. By jointly exploiting the dynamic characteristics of wireless channels and computational capability at clients, we optimized the federated learning network by maximizing the number of active clients under the constraint of latency and system bandwidth. Two BA schemes were proposed to optimize the FL network, based on the I-CSI and S-CSI, respectively. Simulation results were finally provided to demonstrate the effectiveness of the proposed two BA schemes for the considered federated learning network. In particular, the proposed I-CSI based BA scheme can achieve almost the same performance as the baseline one for a wide range of bandwidth, while the proposed S-CSI based BA scheme outperforms the conventional UA one.

References

[1] X. Cao, J. Zhang, and H. V. Poor, "A virtual-queue-based algorithm for constrained online convex optimization with applications to data center resource allocation," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 4, pp. 703–716, 2018.

[2] X. Cao, Y. Chen, and K. J. R. Liu, "Data trading with multiple owners, collectors, and users: An iterative auction mechanism," *IEEE Trans. Signal Inf. Process. over Networks*, vol. 3, no. 2, pp. 268–281, 2017.

[3] S. Tang, "Dilated convolution based CSI feedback compression for massive MIMO systems," *IEEE Trans. Vehic. Tech.*, vol. PP, no. 99, pp. 1–5, 2021.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, vol. 54, 2017, pp. 1273–1282.

[5] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distributed Syst.*, vol. 31, no. 8, pp. 1754–1766, 2020.

[6] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2150–2167, 2021.

[7] L. Cui, X. Su, Z. Ming, Z. Chen, S. Yang, Y. Zhou, and W. Xiao, "CREAT: Blockchain-assisted compression algorithm of federated learning for content caching in edge computing," *IEEE Internet of Things Journal*, pp. 1–10, 2020.

[8] Y. Guo and S. Lai, "Distributed machine learning for multiuser mobile edge computing systems," *IEEE J. Sel. Top. Signal Process.*, vol. PP, no. 99, pp. 1–12, 2021.

[9] S. Tang and L. Chen, "Computational intelligence and deep learning for next-generation edge-enabled industrial IoT," *IEEE Trans. Network Science and Engineering*, vol. PP, no. 99, pp. 1–12, 2022.

[10] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Netw.*, vol. 35, no. 1, pp. 148–155, 2021.

[11] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 1, pp. 453–467, 2021.

[12] J. Li, X. Shen, L. Chen, and J. Chen, "Bandwidth slicing to boost federated learning over passive optical networks," *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1492–1495, 2020.

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

[14] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 10th ed. New York: Dover: Academic, 1972.