A General Solution for Straggler Effect and Unreliable Communication in Federated Learning

Tianming Zang[1,2], Ce Zheng[3*], Shiyao Ma[4], Chen Sun[3], and Wei Chen[1,2]
[1]Department of Electronic Engineering, Tsinghua University, Beijing, 100084, CHINA
[2]Beijing National Research Center for Information Science and Technology (BNRist)
[3]Research and Development Center, SONY(China) Ltd, Beijing, 100084, CHINA
[4]Department of Information and Communication Engineering, Dalian Minzu University, Dalian, 116600, CHINA
Email: zangtm20@mails.tsinghua.edu.cn, ce.zheng@sony.com,
shiyaoma@gmail.com, chen.sun@sony.com, wchen@tsinghua.edu.cn

***Abstract*—The *straggler effect* is the main bottleneck for Federated Learning (FL), where the performance of training is degraded by the slowest member. Another significant problem is *unreliable communication*, which somehow has been neglected in previous studies. That is, the transmission of local models is not successful every time. In this paper, we find that the problems of straggler effect and unreliable communication are implicitly caused by *time divergence* of User Equipments (UEs) in each training round. Based on this, we propose our solutions for these two problems and show that our solutions can be merged into a general one: the problem of the straggler effect and unreliable communication can be solved with a simple UE selection method. This method consists of two steps: First, we cluster UEs into several groups based on UEs' physical parameters or performance metrics; Second, in each training round, only UEs from the same group are chosen for FL operation. Full explanations are given why the *time divergence* is statistically reduced, and therefore it can mitigate the aforementioned two problems. Our solutions are further illustrated with some examples and validated by simulations.**

***Index Terms*—federated learning, straggler effect, unreliable communication, time divergence, re-transmission**

I. Introduction

Federated Learning (FL) was first proposed by McMahan in 2016 [1], where models are trained based on the local data at User Equipments (UEs). In this way, privacy is protected and computation burden at the server or base station (BS) is released. However, it faces two challenges:

*1) **Straggler effect**:* Due to the heterogeneity of computation and communication, some UEs may fail to complete their local training and upload their models in time, which become "stragglers" and hinder the whole FL training process.

To mitigate this problem, various solutions are put forward. Some works focus on *sampling* and try to avoid selecting stragglers for FL participation [2]–[4]. However, this leads to the global model biased towards the model of "fast" UEs. Other methods propose computation offloading. That is, offload the data on stragglers to other UEs [5], [6] or edge servers [7], [8] to reduce the computation time. But this requires a trusting environment and may have privacy issues.

*2) **Unreliable communication**:* Most works assume reliable communication where models are perfectly received at BS. However, transmission errors may happen. Local models will be abandoned from the global aggregation if they are not successfully received. As a result, the global model will be biased towards UE models with good channel quality.

Corresponding author: Ce Zheng[3*] (ce.zheng@sony.com)

To address this problem, [9] first studies the impact of packet error and derive a closed-form expression. Then an FL algorithm suitable for unreliable and resource-constrained wireless systems is proposed in [10]. In [11], [12], FL convergence is analyzed under transmission outage and quantization errors. The simplest way dealing with unreliable communication is the re-transmission strategy: [10] considers re-transmission for each device, and [12] employs re-transmission when all devices encounter outage.

In this paper, we aim to propose a general solution for the problems of straggler effect and unreliable communication. Inspired by [13], we find out the essence of straggler effect is the problem of time divergence. Straggler effect can be mitigated as long as the time divergence is reduced in each training round. Based on this observation, we propose our first solution. The idea is essentially to put UEs with the similar upload time or communication time in the same training round, which will reduce the time divergence statistically. As for unreliable communication, the re-transmission strategy is considered. To the best of our knowledge, no one ever studies the impact of prolonged time on FL performance due to re-transmission. We give a clear analysis that unreliable communication with re-transmission can be treated as a straggler problem. On that account, similar solutions are given. Moreover, we do some simulations to validate our solutions. In the end, we merge the solutions for straggler effect and unreliable communication into a general one, and discuss its extensions and potential impact on the 3rd Generation Partnership Project (3GPP) standards.

The rest of this paper is organized as follows. Section II gives the system model. In Section III and Section IV, solutions for straggler effect and unreliable communication are provided along with the analysis, respectively. Simulation results are presented to validate our solutions in section V. In Section VI, a general solution is given, and its impact on 3GPP standards is further discussed. Conclusions are drawn in Section VII.

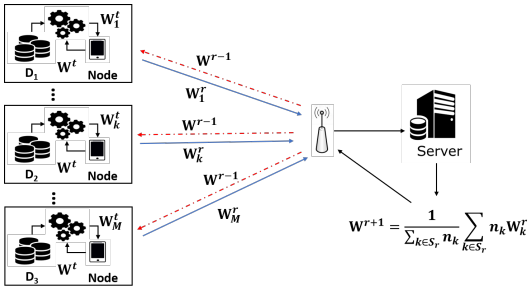


Fig. 1. Federated averaging learning process.

II. System Model

A. Federated learning model

In this work, we consider a federated learning process where BS or server tries to solve the following distributed optimization problem:

$$\min \quad F(w) = \sum_{k=1}^{N} u_k F_k(w), \tag{1}$$

where w is the model parameters to be learned. N is the number of UEs. u_k is the weight of UE#k, $u_k \geq 0$ and $\sum_{k=1}^{N} u_k = 1$. $F_k(w)$ is the local loss function. Let $\mathcal{D}_k$ denotes the local dataset on UE#k, and we have

$$F_k(w) = \frac{1}{n_k} \sum_{j=1}^{n_k} \ell(w; x_k^j, y_k^j), \quad k = 1, 2, \cdots, N, \tag{2}$$

where $n_k = |\mathcal{D}_k|$ is the number of samples in $\mathcal{D}_k$. (x_k^j, y_k^j) is the j-th sample of UE#k. $\ell(w; x_k^j, y_k^j)$ is the loss function on (x_k^j, y_k^j).

We employ the *FedAvg* algorithm [1]. In the r-th FL training round, *FedAvg* executes the following steps (see Fig. 1):

*1) **UE selection and broadcasting**:* BS first selects a candidate set S_r out of M UEs, where $|S_r| = N$[1] and $N \leq M$. Each element of S_r represents the index of selected UE. It then broadcasts the global model w^{r-1} to UEs in S_r.

*2) **Local model updating**:* Each UE#$k \in S_r$ updates the local model as follows:

$$\begin{aligned} \bar{w}_k^{0,r} &= w^{r-1} \\ \bar{w}_k^{j,r} &= \bar{w}_k^{j-1,r} - \eta \bigtriangledown F_k(\bar{w}_k^{j-1,r}), \ j = 1, ..., \tau, \\ w_k^r &= \bar{w}_k^{E,r}, \end{aligned} \tag{3}$$

where $\eta > 0$ is the learning rate. τ is the number of iterations. w^r is the updated global model in r-th training round, and w^0 represents the initialized model at the beginning. $\bar{w}_k^{j,r}$ is the updated model parameters in the j-th iteration of r-th round at UE#k. $\bigtriangledown F_k(\bar{w}_k^{j-1,r})$ is the gradient of UE#k at $\bar{w}_k^{j-1,r}$. w_k^r is the model parameter to be uploaded in r-th round.

*3) **Aggregation**:* The selected UEs upload their local models w_k^r to BS, and BS aggregates all received local models to generate a new global model based on *FedAvg* [1]:

$$w^r = \frac{1}{\sum_{k \in S_r} n_k} \sum_{k \in S_r} n_k w_k^r. \tag{4}$$

[1] N is usually set as the number of resource blocks (RBs) for communication in the system (one RB for one UE at most), i.e., the maximum number of UEs that BS can associate with.

B. Communication and computation time

We characterize the communication and computation time as follows:

*1) **Uplink transmission time**:* Let $T_{k,r}^{up}$ denote the time of UE#k transmitting its local model in the uplink at r-th training round, and

$$T_{k,r}^{up} = \frac{S_{model}}{R_{k,r}}, \tag{5}$$

where S_{model} is the size of ML model parameters, and $R_{k,r}$ is the uplink transmission rate.

*2) **Communication time**:* The total communication time is given by

$$T_{k,r}^{comm} = L_k \cdot T_{k,r}^{up}, \tag{6}$$

where L_k is the number of transmissions.

We neglect the downlink transmission time for the following reasons: 1). BS transmit power is sufficiently large; 2). Global model is broadcast and could occupy the whole bandwidth.

*3) **Computation time**:* To capture the randomness of computation time, we employ the shifted exponential distribution [3], [14]:

$$P[t_{k,r}^{comp} < t] = \begin{cases} 1 - \exp\left(-\frac{\mu_k}{\tau u_k}(t - a_k \tau u_k)\right), & t > a_k \tau u_k \\ 0, & otherwise \end{cases}, \tag{7}$$

where $a_i > 0$ is the maximum computation capability and $\mu_i > 0$ is the fluctuation of the computation capability. And we assume these parameters remain constant during the training process.

The computation time at BS is not considered as the BS has high computational power and only performs low-complexity model aggregation.

*4) **Upload time**:* The upload time is defined as the sum of computation time and communication time:

$$T_{k,r}^{upload} = T_{k,r}^{comp} + T_{k,r}^{comm}. \tag{8}$$

C. Communication model

In uplink, the channel capacity of UE#k is given by

$$C_k = b_k \log_2 \left(1 + \frac{P_k |g_k|^2}{b_k N_0}\right), \tag{9}$$

where b_k denotes the bandwidth allocated for UE#k with $\sum_{k=1}^{N} b_k = B$, and B is the total bandwidth. P_k is the transmit power at UE#k. g_k is the channel coefficient. N_0 is the noise power spectral density.

We further assume an equal bandwidth allocation within N UEs. Each UE has the same transmit power, and

$$|g_k|^2 = |h_k|^2 d_k^{-\alpha}, \tag{10}$$

where $|h_k|^2 \sim Exp(1)$ is the Rayleigh fading. d_k is the distance between UE and BS, and α is the path-loss coefficient.

Then we have

$$C_k = b \log_2 \left(1 + \frac{P_k |h_k|^2 d_k^{-\alpha}}{b N_0}\right), \tag{11}$$

where $b = B/N$.

According to the channel coding theorem [15], given the target rate R_k, when the channel capacity C_k is lower than the target rate R_k, the outage occurs. That is, BS can't decode

the received local model correctly. Therefore, the outage probability is denoted as

$$q_k = Pr(C_k \leq R_k). \tag{12}$$

Then, the success probability of a single transmission is

$$p_k = 1 - q_k. \tag{13}$$

To increase reliability, re-transmission schemes may be considered, which will be further discussed in Section IV.

III. Straggler Effect & Our Solution

In this section, we offer our solutions for straggler effect and give our analysis of how it works.

A. Straggler effect: intuition on time divergence

Observe that if all UEs take the same time, there will be no stragglers. Mathematically, it is that the time divergence between all UEs equals zero. In intuition, the straggler does less harm to the "slow" UEs as they have less "waiting time" compared with "fast" ones. Therefore, if we put the straggler and slow UEs in the same training round, the negative effect of straggler becomes less severe. By doing so, we implicitly reduce the time divergence between UEs.

Essentially, we can mitigate straggler effect by reducing the time divergence between UEs. And we give our solution on how to do this in the next subsection.

B. Solution: clustering based on upload time or communication time

Assume all UEs' transmissions are successful, and we have

***Solution#1(Clustering based on upload time)**:*
*1). We cluster the UEs into K groups based on the **upload time** where UEs with same or similar **upload time** are put into the same group; 2). In each training round, only UEs from the same group are selected for FL operation.*

An example is given in Fig. 2. There are 6 UEs, and only 3 UEs can be chosen in each training round. In random UE selection scheme, UE#3 becomes the straggler in the first round, and so do UE#2 and UE#6 in the second round. To alleviate straggler effect, we employ Solution#1 by putting the fast ones (UE#1, UE#4, UE#5) in the first round and the slow ones (UE#2, UE#3, UE#6) in the second round. It is evident that our solution outperforms the random selection method in terms of execution time ($t'_1 + t'_2 < t_1 + t_2$). Since all UEs are chosen only one time for training in the two rounds, fairness and accuracy are also guaranteed.

The philosophy of Solution#1 is that the divergence of upload time is, as much as possible, minimized in each training round. It is based on the implicit condition that the upload time $T_{k,r}^{upload}$ stays the same for all training rounds. However, such conditions can be easily released.

In most of the time, the time divergence is a random variable depending on the selected UEs in each training round. If the divergence of upload time is reduced in statistics, the performance will also be improved over a sufficient number of training rounds. Therefore, we could exploit the statistical value of upload time for clustering, e.g., the mean value.

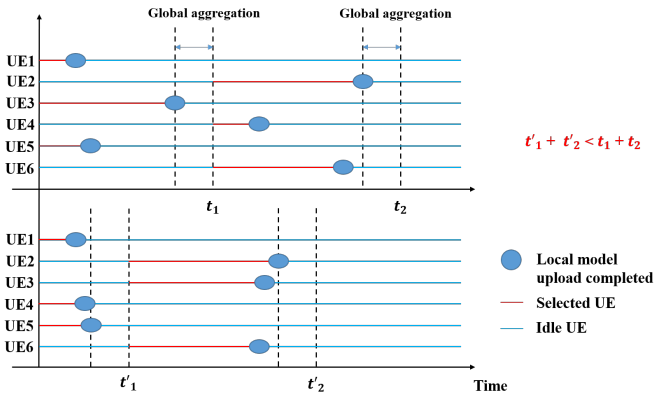


Fig. 2. Comparison of random UE selection and Solution#1.

***Solution#2(Clustering based on communication time)**:*
*Replace **upload time** in Solution#1 with **communication time**, we get this solution.*

According to (8), upload time is the summation of computation time and communication time. In general, the computation time is decided by UEs' computation capability, data size, etc. On the other hand, the communication time depends on channel quality, transmit power, number of antennas, etc. Hence, one can regard them as two independent variables. And the divergence of upload time becomes the summation of the divergence of computation time and the divergence of communication time. As a result, we can reduce the divergence of upload time by minimizing the divergence of communication time.

The communication time may vary in each round due to the uncertainty of channel condition and UE mobility. Similar to Solution#1, Solution#2 may be adapted by exploiting the mean value of communication time or maybe uplink transmission rate, etc.

C. Implementation in practice

In practice, computation time and communication time are unknown but can be estimated beforehand: computation time can be estimated with UE's CPU, CPU occupancy, data size, etc. Communication time can be estimated with the UE's location, transmit power, etc. In addition, the experience or empirical results could be exploited as well. That is, before the FL starts, we may let UEs run some computation or transmit a few packets to BS for estimation.

The number of groups K is usually given as

$$K = \left\lceil \frac{M}{N} \right\rceil, \tag{14}$$

where $\lceil \cdot \rceil$ is the ceiling function.

IV. Unreliable Communication

In this section, we offer our solutions for unreliable communication and give our analysis of how it works.

A. Unreliable communication: re-transmission strategy

In practice, local models will be abandoned from the global aggregation if they are not successfully received. Thus, (4) is rephrased as

$$w^r = \frac{1}{\sum_{k\in S_r} a_k n_k} \sum_{k\in S_r} a_k n_k w_k^r. \tag{15}$$

where $a_k = 1$ if the local model of UE#k is successfully received, otherwise $a_k = 0$. And $a_k \sim B(1, p_k)$, where p_k is given in (12).

For UE#k with a bad channel quality, p_k is small, and a_k is more likely to be zero. As a consequence, the aggregated global model will be biased towards the models of UEs with good channels. To tackle this issue, we employ the *continuous re-transmission strategy*:

***Continuous re-transmission strategy**: When transmission outage occurs, BS requires UEs that fail to upload their model to re-transmit till it is received successfully, or the maximum number of transmissions $L_{\max}$ is reached.*

We assume transmission rate is same and remains a constant for all UEs, denoted as R. And thus, so is the uplink transmission time for one transmission T^{up}. Then we have

$$T_{k,r}^{upload} = T_{k,r}^{comp} + L_k \cdot T^{up} \tag{16}$$

To the fullest extent, this strategy ensures that BS successfully receives the local model of all selected UEs for global aggregation. However, UEs with poor channel quality tend to have more re-transmissions, which prolongs the whole communication time, and thus the upload time in (16).

An example is given in Fig. 3, where we assume 8 UEs with Group#1={UE#1, UE#2, UE#3, UE#4} of good channel quality and Group#2={UE#5, UE#6, UE#7, UE#8} of bad channel quality. In most times, UEs from Group#2 have more transmissions, prolonging the communication time and upload time in each round. These "bad" UEs are more inclined to become stragglers due to the increased duration of re-transmission. What is more, the prolonged upload time of "bad" UEs increases the time divergence between them and "good" ones.

Accordingly, we could solve this problem with UE selection by clustering in the same manner as dealing with the straggler problem.

B. Solution: clustering based on NR measurements (RSRP, RSRQ, SINR, etc)

***Solution#3(Clustering based on NR measurements)**: Replace **upload time** in Solution#1 with **NR measurements** (e.g. SNR, RSRP, RSRQ, etc.), we get this solution.*

Given p_k in (13), the probability that local model of UE#k is successfully received at l-th transmission is:

$$P(L_k = l) = p_k(1-p_k)^{l-1}. \tag{17}$$

Thus, the average number of transmissions for UE#k is

$$\begin{aligned}\bar{L}_k &= \sum_{l=1}^{L_{\max}} l \cdot P(L_k = l) \\ &= p_k \left[\frac{1-(1-p_k)^{L_{\max}-1}}{p_k^2} - L_{\max} \frac{(1-p_k)^{L_{\max}}}{p_k} \right].\end{aligned} \tag{18}$$

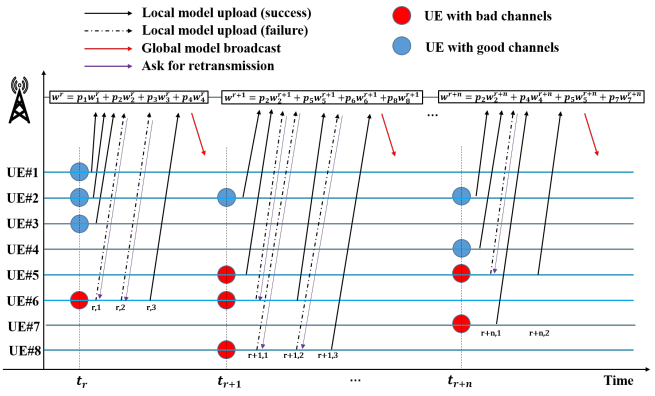


Fig. 3. Random UE selection under the continuous re-transmission strategy.

When $L_{\max}$ is large enough, i.e., $L_{\max} \to \infty$, we have

$$\bar{L}_k = 1/p_k. \tag{19}$$

And the average communication time of UE#k is

$$\bar{T}_k^{comm} = \frac{T^{uplink}}{p_k} = \frac{S_{model}}{R} \cdot \frac{1}{p_k}, \tag{20}$$

where p_k is different among UEs due to the variety of channel conditions.

As the average communication time depends on p_k, we can solve this "implicit straggler problem" by clustering based on p_k or q_k. But the estimation of p_k or q_k requires multiple transmissions before FL process.

According to 3GPP [16], The 5G NR measurements are good metrics that represent the channel condition or outage probability, e.g., Signal-to-Interference-plus-Noise Ratio (SINR), Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ), etc.

The channel capacity is

$$C_k = b_k log_2(1 + SINR). \tag{21}$$

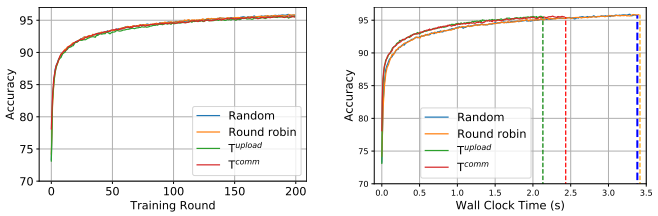
Therefore, the outage probability $P(C_k < R)$ is equivalent to $SINR = \frac{S}{I+N} < \gamma$, where S is the received signal power, and S can be measured with the RSRP or RSRQ. For this reason, we can also consider clustering based on NR measurements.

C. Implementation in practice

The benefits of Solution#3 are that 5G NR measurements have been standardized as SS-RSRP, CSI-RSRP, SS-RSRQ, CSI-RSRQ, SS-SINR, CSI-SINR, etc [16], which do not need further estimation or execution process.

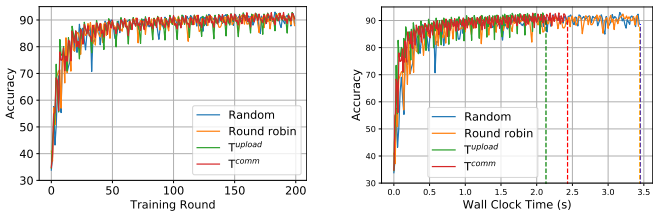
V. Simulation Results

We refer to and use most of the setup in [17]. Assume we have $M = 100$ UEs uniformly distributed over the cell with radius $D_{\max} = 600$ m. In each round, only $N = 10$ UEs can be chosen. The bandwidth is $B = 20$ MHz. With equal allocation, the bandwidth for each UE is $b = \frac{B}{N} = 2$ MHz. (11) is employed. Furthermore, we assume the channel capacity follows (11). The path-loss exponent and noise power spectrum are set $\alpha = 3.76$ and $N_0 = -114$ dBm, respectively. $|h_k|^2 \sim \exp(1)$ is the Rayleigh fading. Noise-limited scenarios are considered, i.e., the interference can be ignored.



(a) Training round (b) Wall-clolck time

Fig. 4. Straggler effect: comparison of four UE selection methods on IID dataset — random selection, round robin, clustering based on upload time and clustering based on communication time.



(a) Training round (b) Wall-clolck time

Fig. 5. Straggler effect: comparison of four UE selection methods on non-IID dataset — random selection, round robin, clustering based on upload time and clustering based on communication time.

The multilayer perceptron (MLP) with a single hidden layer of 64 nodes is chosen as the training model, and ReLU activation is used. Hence the model size is $S_{model} = 0.2$ MB. Let $a = 0.5$ ms/sample and $\mu = \frac{1}{a}$ in (7).

As for data partitioning, we employ the method from [1]:

- **IID**: The MNIST 60,000 training images are partitioned into 100 datasets with each of 600 samples. Then each UE is allocated with one dataset.
- **Non-IID**: The 60,000 training images are sorted by digit label and divided into 200 shards of size 300. Then we assign each of 100 clients 2 shards.

A. Simulation for straggler effect

We assume perfect CSI, and UE can adapt its rate to achieve zero-error transmission. UEs' location is assumed to be fixed. The transmit power for each UE is set the same: $P = 10$ dBm. Then, the transmission rate of UE#k at r-th round is

$$R_{k,r} = b \log_2 \left(1 + \frac{P|h_{k,r}|^2 d_k^{-\alpha}}{bN_0}\right), \tag{22}$$

where $|h_{k,r}|^2$ is the value of small-scale fading at r-th round.

Then communication time $T_{k,r}^{up}$ can be computed with (5). The computation time is sampled following (7).

Fig. 4 gives the simulation results under IID data, where four methods are considered:

- Random: In each round, 10 UEs are randomly selected;
- Round robin [18]: The M UEs are randomly divided into N groups, each with $K = \frac{M}{N}$ UEs. Each group joins FL consecutively. This process reinitializes every K training rounds. That is, UEs will be regrouped after all of them have been chosen one time.
- T^{upload}: Clustering based on upload time;
- T^{comm}: Clustering based on communication time.

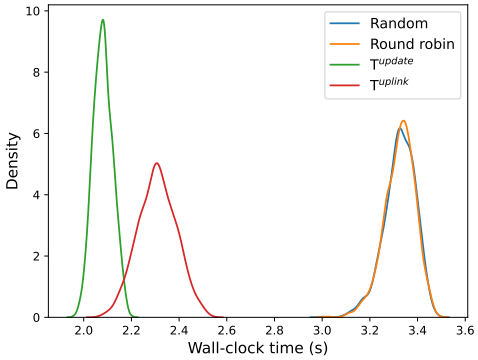
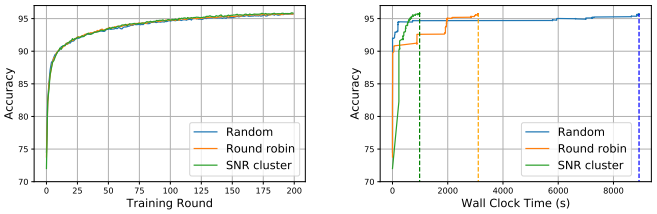


Fig. 6. Straggler effect: PDF of wall-clock time for performing 200 training rounds under four UE selection methods: random selection, round robin, clustering based on upload time and clustering based on communication time.



(a) Training round (b) Wall-clock time

Fig. 7. Unreliable communication: comparison of three UE selection methods on IID dataset — random selection, round robin, clustering based on SNR.

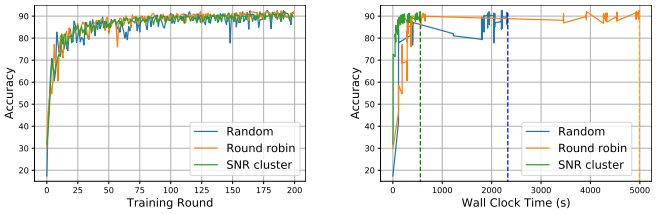
As is shown in Fig. 4(a), there is little difference on accuracy in terms of training round. However, as for wall-clock time, the method clustering based on upload time takes 2.13s and the method on communication time takes 2.46s to finish 200 training rounds, which are less than the other two (approximately 3.4s). Similar results are also shown in Fig. 5 with the non-IID data.

To further illustrate the performance of our methods in statistics, Fig. 6 gives the probability density function (PDF) of wall-clock time for performing 200 training rounds, which is estimated from 500 Monte Carlo simulations. It is obvious that our methods are better than the other two, which is consistent with Fig. 4 and Fig. 5. In addition, the method clustering based on upload time is better than that clustering based on communication time because the computation time is also accounted for reducing the time divergence.

B. Simulation for unreliable communication

In this subsection, we assume all UEs transmit at a fixed target rate $R = 15$ MB/s, and transmission error may occur. The transmit power of each UE may be different. For ease of analysis, we uniformly choose a value from $[7, 10, 13, 16, 19]$ dBm for each UE as its transmit power. According to (11) and (12), the SNR and outage probability will vary between UEs.

Similar to Section V-A, three methods are considered: random, round robin, and clustering based on SNR (denoted as SNR cluster). As is shown in Fig. 7, Fig. 8 and Fig. 9, our method of clustering outperforms the other two in terms of wall-clock time.



(a) Training round (b) Wall-clock time

Fig. 8. Unreliable communication: comparison of three UE selection methods on non-IID dataset — random selection, round robin and clustering based on SNR.

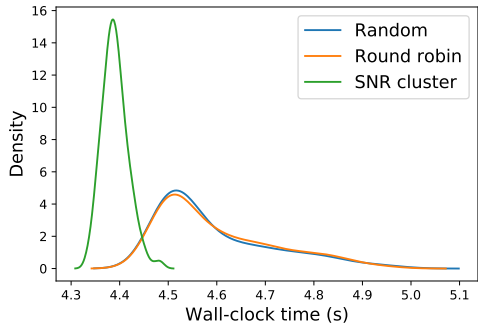


Fig. 9. Unreliable communication: PDF of wall-clock time for performing 200 training rounds under random selection, round robin and clustering based on SNR.

VI. A General Solution and Its Impact on 3GPP

The solutions in Section III and Section IV can be merged into a general one:

***General Solution(Clustering based on performance metrics or physical parameters)**:*
1). We cluster the UEs into K groups based on the $\mathbf{Q}$ where UEs with same or similar $\mathbf{Q}$ are put into the same group; 2). In each training round, only UEs from the same group are selected for FL operation.

$\mathbf{Q}$ could be performance metrics introduced in Section III and Section IV, i.e., computation time, communication time, transmission rate, and NR measurement. It could also be extended to include other physical parameters that impact the time divergence directly or indirectly. For example, the distance to BS for each UE: the larger distance, the lower transmission rate and smaller communication time, or larger number of retransmissions. These existing or future performance metrics or physical parameters could be included in 3GPP for FL UE selection.

VII. Conclusion

In this paper, we provide our solutions for two problems: straggler effect and unreliable communication, and validate them with illustration, analysis, and simulations. For straggler effect, our clustering method aims to reduce the time divergence in each training round. For unreliable communication, it can be converted into a "straggler" problem and solved with clustering, because prolonged time caused by retransmission leads to an increase of time divergence, as well. These solutions are further merged into a general one which could have a further impact on 3GPP standards in FL.

Acknowledge

This paper is based on the work of SONY-Tsinghua Coresearch Project fully funded by the Sony (China) Ltd. Tianming ZANG and Ce ZHENG (Corresponding author) contribute most to paper (ideas, writing, etc). Shiyao MA does the simulations. Wei CHEN and Chen SUN are in charge of the project from Tsinghua and Sony, respectively.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[3] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

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

[5] J. X. Xiaoran CAI, Xiaopeng MO, "D2d computation task offloading for efficient federated learning," *Chinese Journal on Internet of Things*, pp. 82–90, 2019.

[6] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[7] Z. Ji, L. Chen, N. Zhao, Y. Chen, G. Wei, and F. R. Yu, "Computation offloading for edge-assisted federated learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9330–9344, 2021.

[8] S. Dong, D. Zeng, L. Gu, and S. Guo, "Offloading federated learning task to edge computing with trust execution environment," in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2020, pp. 491–496.

[9] 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 Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[10] M. Salehi and E. Hossain, "Federated learning in unreliable and resource-constrained cellular wireless networks," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5136–5151, 2021.

[11] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 323–341, 2021.

[12] ——, "Robust federated learning in wireless channels with transmission outage and quantization errors," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2021, pp. 586–590.

[13] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, 2020, pp. 125–136.

[14] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Coded computation over heterogeneous clusters," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4227–4242, 2019.

[15] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[16] "NR; Physical layer measurements," *3GPP TS 38.215*.

[17] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, 2020.

[18] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, 2019.