Hierarchical Federated Learning in Delay Sensitive Communication Networks

Abdulmoneam Ali Ahmed Arafa
Department of Electrical and Computer Engineering
University of North Carolina at Charlotte, NC 28223
aali28@uncc.edu *aarafa@uncc.edu*

***Abstract*—The impact of local averaging on the performance of federated learning (FL) systems is studied in the presence of *communication delay* between the clients and the parameter server. We focus on a *hierarchical* FL (HFL) setting where clients are assigned into different groups, each having its own *local* parameter server (LPS). The number of local and global communication rounds in our work is randomly determined by the (different) delays experienced by each group of clients. Specifically, the number of local averaging rounds are tied to a wall-clock time period coined the sync time S, after which the LPSs synchronize their models by sharing them with a *global* parameter server (GPS). Such sync time S is then reapplied until a global wall-clock time is exhausted. First, an upper bound on the deviation between the updated model at each LPS with respect to that available at the GPS is derived. This is then used to derive the convergence bounds of our proposed HFL setting, at each LPS and at the GPS. The bounds showcase the effects of the whole system's parameters, including the number of groups, the number of clients per group, and the value of S. Our results show that optimizing S is especially crucial in heterogeneous systems in which the number of clients per group and their delay statistics are different. In addition to showing the necessary need of collaboration for under-performing groups, optimizing the value of S promotes fairness among groups, and allows one to deal with delay-sensitive FL applications in which the training time is restricted.**

I. Introduction

Federated Learning (FL) is a distributed machine learning training system in which edge devices (clients) collaboratively train a model of interest based on their locally stored datasets. A central node (parameter server) orchestrates the learning process by collecting the clients' parameters for aggregation [1]. Due to its data privacy preserving and bandwidth saving nature, FL has attracted a lot of attention and has been used in diverse applications including healthcare and mobile services.

Challenges. In order to successfully deploy FL in communication networks, lots of challenges should be addressed. These include: the computing capabilities of the clients; the communication overhead between the clients and the parameter server; and the system heterogeneity, whether in the clients' communication channels or their data statistics.

Among all challenges, communication remains to be the bottleneck issue, and various solutions have been proposed in the literature to mitigate it. One of these solutions is to introduce intermediate parameter servers, denoted local parameter servers (LPSs), between the clients and the (now) global parameter server (GPS). Such setting of FL is known in the literature as the *hierarchical* FL (HFL) setting [2]. The main advantage of having LPSs close to the clients is to reduce the latency and required energy to communicate with the GPS [3]. In [4], a joint resource allocation and client association problem is formulated in an HFL setting, and then solved by an iterative algorithm. Reference [5] shows that HFL settings can also enhance data privacy.

Contributions. To cope with the very low latency service requirements in 6G (and beyond), in this paper we focus on HFL for *delay-sensitive* communication networks. We study FL settings that have an additional requirement of conducting training within a predefined deadline. Such scenario is relevant for, e.g., energy-limited clients whose availability for long times is not always guaranteed. To enforce the system to abide by this constraint, the number of local training updates will be determined by a wall-clock time. Specifically, we define a *sync time* S within which the LPSs are allowed to aggregate the parameters they receive from their groups' clients. Each local iteration consumes a random group-specific *delay*, and hence the total number of local updates within S will also be random, and could possibly be *different* across groups. This dissimilarity in the delay statistics is introduced to capture, e.g., the effects of wireless channels and different computational resources among different group clients. Following the deadline S, the LPSs forward their models to the GPS.

We set another time constraint at the GPS and denote it the *total system time* T. This is the total allowed time for the overall HFL system to perform the training and get its final model parameter. Different values of S and T will lead to a different number of local and global updates. Thus, by controlling S, we also control how many times the clients will communicate with the GPS, i.e., more local iterations would lead to less global ones. This is is different from the existing works that assume that the global communication rounds are constant and unaffected by the number of local updates.

We present a thorough theoretical convergence analysis for the proposed HFL setting for non-convex loss functions. Our results show how the different system parameters affect the accuracy, namely, the wall-clock times S, T, the number of groups, and the number of clients per group. Various experiments are then performed to show how to optimize the sync time S based on the other system parameters.

This work was supported by the U.S. National Science Foundation under Grants CNS 21-14537 and ECCS 21-46099.

II. System Model

We consider an HFL system with a global PS (GPS) and a set of local PSs (LPSs), $\mathcal{N}_g$, that serve a number of clients. Clients are distributed across different LPSs to form clusters (groups), in which a client can only belong to one group, and may only communicate with one specific LPS. Denoting by $\mathcal{N}_i$ the set of clients in group i, the total number of clients in the system is $\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|$, with $|\cdot|$ denoting cardinality. Each client has its own dataset, and the data is independently and identically distributed (i.i.d.) among clients. The empirical loss function at the LPS of group $i\in\mathcal{N}_g$ is defined as follows:

$$f_i(x)\triangleq\frac{1}{|\mathcal{N}_i|}\sum_{k\in\mathcal{N}_i}F_{i,k}(x),\quad i\in\mathcal{N}_g,\tag{1}$$

where $F_{i,k}(x)$ is the loss function at client k in group i. The goal of the HFL system is to minimize a global loss function:

$$\begin{aligned}f(x)&\triangleq\frac{1}{\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|}\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|f_i(x)\\&=\frac{1}{\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|}\sum_{i\in\mathcal{N}_g}\sum_{k\in\mathcal{N}_i}F_{i,k}(x).\end{aligned}\tag{2}$$

The global loss function is minimized over a number of *global communication rounds* between the GPS and the LPSs. At the beginning of the uth global round, the GPS broadcasts the global model, $x^u\in\mathbb{R}^d$, with d representing the model dimension, to the LPSs. The LPSs then forward x^u to their associated clients, which is used to run a number of SGD steps based on their own local datasets. After each SGD step, the clients share their models with their LPS, which aggregates them and broadcasts them back locally to its clients. We call this local round trip a *local iteration*. We further illustrate how the global rounds and local iterations interact as follows. Let $x_i^{u,l}$ denote the model available at LPS i after local iteration l during global round u, and let $x_{i,k}^{u,l}$ denote the corresponding local model of client k of group i. We now have the following equations that build up the models:

$$x_i^{u,0}=x^u,\quad\forall i\in\mathcal{N}_g,\tag{3}$$

$$x_{i,k}^{u,0}=x_i^{u,0},\ x_{i,k}^{u,l}=x_i^{u,l-1}-\alpha\,\tilde{g}_{i,k}\left(x_i^{u,l-1}\right),\quad\forall k\in\mathcal{N}_i,\tag{4}$$

where α is the learning rate, and $\tilde{g}_{i,k}$ is an unbiased stochastic gradient evaluated at $x_i^{u,l-1}$. After the lth SGD step, LPS i collects $\left\{x_{i,k}^{u,l}\right\}$ from its associated clients and aggregates them to get the lth local model,

$$x_i^{u,l}=\frac{1}{|\mathcal{N}_i|}\sum_{k\in\mathcal{N}_i}x_{i,k}^{u,l},\tag{5}$$

which is shared with its clients to initialize SGD step $l+1$.

Each local iteration takes a *random* time to be completed. This includes the time for broadcasting the local model by the LPS to its clients, the SGD computation time, and the aggregation time. Let $\tau_{i,l}^u$ denote the wall-clock time elapsed during local iteration l for group i in global round u. We assume that $\tau_{i,l}^u$'s are i.i.d. across local iterations l and global rounds u, but may not be identical across groups i. This is motivated by the different channel delay statistics that each group may experience when communicating with its LPS. In addition to that, each group may have clients with heterogeneous computational capabilities. These two factors together hinder one group to (statistically) do an identical number of local updates like other groups. We define a *sync time,* S, that represents the allowed local training time for *all* groups. After the sync time, the LPSs need to report their local models to the GPS, and thereby ending the current global round. During global round u, and within the sync time S, group i will therefore conduct a random number of local iterations given by

$$t_i^u\triangleq\min\left\{n:\ \sum_{l=1}^{n}\tau_{i,l}^u\geq S\right\},\quad i\in\mathcal{N}_g.\tag{6}$$

Observe that the statistics of t_i^u's are not identical across groups, see Fig. 1 for an example sample path during global round u. After the t_i^u local iterations are finished, and using (3)–(5), LPS i will have acquired the following model:

$$x_i^{u,t_i^u}=x_i^{u,0}-\frac{\alpha}{|\mathcal{N}_i|}\sum_{l=0}^{t_i^u-1}\sum_{k\in\mathcal{N}_i}\tilde{g}_{i,k}\left(x_i^{u,l}\right).\tag{7}$$

We consider a synchronous setting in which the GPS waits for all the LPSs to finish their local iterations before a global aggregation. Since LPSs incur different wall-clock times to collect their models, some of them may need to stay idle waiting for others to finish. The GPS therefore starts aggregating the models after

$$\max_{i\in\mathcal{N}_g}\left\{\sum_{l=1}^{n}\tau_{i,l}^u\right\}\tag{8}$$

time units from the start of the local iterations in global round u. We denote this period the *syncing period* (see Fig. 1). When updating the GPS, LPS i sends the difference between its final and initial models, *divided by the number of its local iterations performed,* i.e., it sends

$$\frac{1}{t_i^u}\left(x_i^{u,t_i^u}-x_i^{u,0}\right)=-\frac{\alpha}{|\mathcal{N}_i|}\frac{1}{t_i^u}\sum_{l=0}^{t_i^u-1}\sum_{k\in\mathcal{N}_i}\tilde{g}_{i,k}\left(x_i^{u,l}\right),\ i\in\mathcal{N}_g.\tag{9}$$

We note that the purpose of diving by t_i^u is to avoid *biasing* the global model. To see this, observe that (cf. Assumption 2)

$$\begin{aligned}\mathbb{E}_{|\boldsymbol{t}_i^u}\frac{1}{t_i^u}\left(x_i^{u,t_i^u}-x_i^{u,0}\right)&=-\frac{\alpha}{|\mathcal{N}_i|}\frac{1}{t_i^u}\sum_{l=0}^{t_i^u-1}\sum_{k\in\mathcal{N}_i}\nabla F_{i,k}\left(x_i^{u,l}\right)\\&=-\frac{\alpha}{t_i^u}\sum_{l=0}^{t_i^u-1}\nabla f_i\left(x_i^{u,l}\right),\end{aligned}\tag{10}$$

where $\mathbb{E}_{|\boldsymbol{t}_i^u}$ denotes conditional expectation given the vector $\boldsymbol{t}_i{}^u\triangleq\left\{t_i^{u'}\right\}_{u'=1}^{u}$.

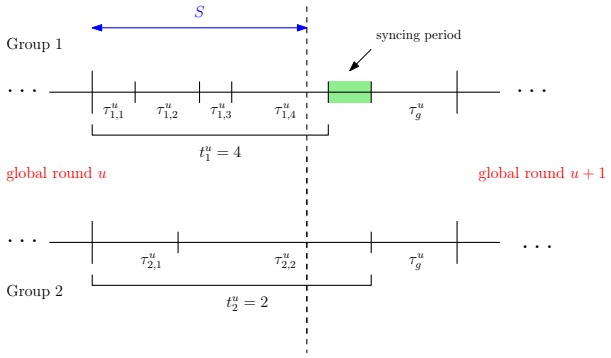


Fig. 1: Example sample path of global rounds and local iterations of 2 groups with wall-clock times considerations.

The GPS then updates its global model as

$$x^{u+1} = x^u + \sum_{i\in\mathcal{N}_g} \frac{|\mathcal{N}_i|}{\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|} \frac{1}{t_i^u}\left(x_i^{u,t_i^u} - x_i^{u,0}\right)$$
$$= x^u - \frac{\alpha}{\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|} \sum_{i\in\mathcal{N}_g} \frac{1}{t_i^u} \sum_{l=0}^{t_i^u-1} \sum_{k\in\mathcal{N}_i} \tilde{g}_{i,k}\left(x_i^{u,l}\right), \tag{11}$$

and broadcasts x^{u+1} to the LPSs to begin global round $u+1$. We assume that the global aggregation and broadcasting processes consume i.i.d. τ_g^u's wall-clock times. An example of the HFL setting considered is depicted in Fig. 1.

The overall HFL training process stops after a total *system time* T. The value of T represents the allowed time budget for training in delay-sensitive applications. Within T, the total number of global rounds will be given by

$$\mathcal{U} \triangleq \min\left\{m : \sum_{u=1}^{m} \max_{i\in\mathcal{N}_g}\left\{\sum_{l=1}^{n} \tau_{i,l}^u\right\} + \tau_g^u \geq T\right\}. \tag{12}$$

We coin the proposed algorithm *delay sensitive HFL*. In the sequel, we analyze its performance in terms of the wall-clock times, number of clients, and other system parameters. We then discuss how to optimize the choice of the sync time S to guarantee better learning outcomes.

III. Main Results

In this section, we present the convergence analysis for the proposed HFL setting. We have the following typical assumptions about the loss function and SGD [3]:

Assumption 1. (*Smoothness*). Loss functions are L-smooth: $\forall x, y \in \mathbb{R}^d$, there exists $L > 0$ such that

$$F_{i,k}(y) \leq F_{i,k}(x) + \langle \nabla F_{i,k}(x), y - x\rangle + \frac{L}{2}\|y - x\|^2. \tag{13}$$

Assumption 2. (*Unbiased Gradient*). The gradient estimate at each client satisfies

$$\mathbb{E}\tilde{g}_{i,k}(x) = \nabla F_{i,k}(x). \tag{14}$$

Assumption 3. (*Bounded Gradient*). There exists a constant $G > 0$ such that the stochastic gradient's second moment is bounded as

$$\mathbb{E}\|\tilde{g}_{i,k}(x)\|^2 \leq G^2. \tag{15}$$

Assumption 4. (*Bounded Variance*). There exists a constant $\sigma > 0$, such that the variance of the stochastic gradient is bounded as

$$\mathbb{E}\|\tilde{g}_{i,k}(x) - \nabla F_{i,k}(x)\|^2 \leq \sigma^2. \tag{16}$$

It is worth noting that we conduct our analysis without relying on the convexity of the loss function at any entity in the system. According to our proposed algorithm, after each global round, the group clients will resume their local training from the aggregated global model instead of the their latest local one. Hence, we need to quantify the *deviation* between the two parameter models through the following lemma[1]:

Lemma 1 *For $0 \leq \alpha \leq \frac{1}{L}$, the delay sensitive HFL algorithm satisfies the following $\forall u, i$:*

$$\mathbb{E}_{|\boldsymbol{t}_i^u}\left\|x^{u+1,0} - x_i^{u,t_i^u}\right\|^2 \leq 2\alpha^2\left(\left(t_i^u\right)^2 + \frac{|\mathcal{N}_g|}{\left(\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|\right)^2} \sum_{j\in\mathcal{N}_g\setminus\{i\}} \left(|\mathcal{N}_j|\right)^2\right) G^2. \tag{17}$$

Remark 1 *The first term in the bound in Lemma 1 represents the contribution of group i while the second one reflects the impact of all groups in the deviation between the parameter models. It is obvious that more local iterations lead to more deviation between the local and the global models. Note that local iterations are the sole determinant of the deviation in case of having one group only (e.g., when there is no hierarchy); having two or more groups carries an additive effect on the deviation as seen in the second term.*

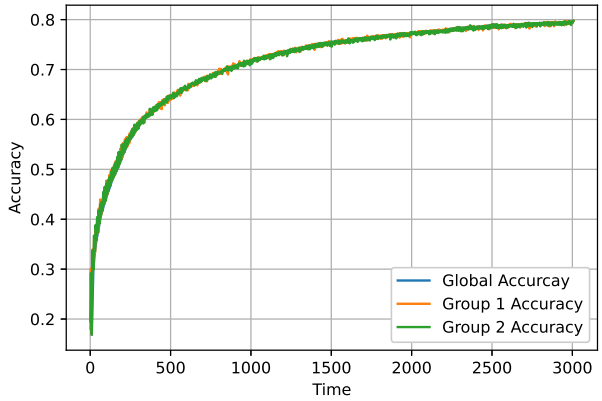
Remark 2 *In case of having only one group in the system, one gets a better bound than that in [6], which is given by $4\alpha^2\left(t_i^u\right)^2 G^2$ (two times the bound in (17) for $|\mathcal{N}_g| = 1$).*

Lemma 1 serves as a building block for our main convergence theorems of the proposed delay sensitive HFL. These are mentioned next.

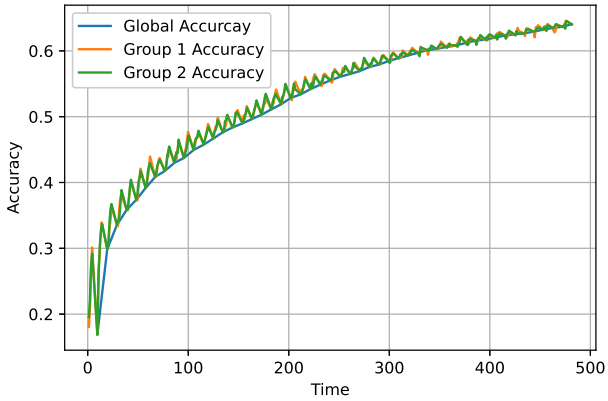
Theorem 1 (Convergence Analysis per Group) *For $0 \leq \alpha \leq \frac{1}{L}$, the delay sensitive HFL algorithm achieves the following group i bound for a given $\mathcal{U}$:*

$$\frac{1}{\sum_{u=1}^{\mathcal{U}} t_i^u} \sum_{u=1}^{\mathcal{U}} \sum_{l=1}^{t_i^u} \mathbb{E}_{|\boldsymbol{t}_i^u}\left\|\nabla f_i(x_i^{u,l-1})\right\|^2 \leq \frac{2}{\alpha\sum_{u=1}^{\mathcal{U}} t_i^u}\left(\mathbb{E}_{|\boldsymbol{t}_i^u} f_i\left(x_i^{1,0}\right) - \mathbb{E}_{|\boldsymbol{t}_i^u} f_i\left(x_i^{\mathcal{U},t_i^{\mathcal{U}}}\right)\right)$$

[1]Due to space limits, we omit the proofs in this paper.



(a) Performance over the training time budget.



(b) Performance during the early training phase.

Fig. 2: HFL system with 10 clients per group with $c_1 = c_2 = 1$, $c_g = 5$ and $S = 5$.

$$+ \alpha L \frac{1}{|\mathcal{N}_i|}\sigma_i^2 + 2\left(\frac{1+(L+2)k\alpha^2}{\alpha\sum_{u=1}^{\mathcal{U}} t_i^u}\right)\mathcal{U}G^2 + \frac{2(L+2)\alpha}{\sum_{u=1}^{\mathcal{U}} t_i^u}\sum_{u=1}^{\mathcal{U}}\left(t_i^u\right)^2 G^2. \quad (18)$$

Theorem 2 (Global Convergence Analysis) *For $0 \leq \alpha \leq \frac{1}{L}$, the delay sensitive HFL algorithm achieves the following global bound for a given $\mathcal{U}$:*

$$\frac{1}{\mathcal{U}}\sum_{u=1}^{\mathcal{U}} \mathbb{E}_{|\boldsymbol{t}_i^u} \left\|\nabla f\left(x^u\right)\right\|^2 \leq \frac{2}{\alpha}\frac{1}{\mathcal{U}}\left(\mathbb{E}_{|\boldsymbol{t}_i^u} f\left(x^1\right) - \mathbb{E}_{|\boldsymbol{t}_i^u} f\left(x^{\mathcal{U}+1}\right)\right) + \frac{\alpha L|\mathcal{N}_g|\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|\sigma_i^2}{\left(\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|\right)^2} + \frac{1}{\mathcal{U}}\sum_{u=1}^{\mathcal{U}}\frac{4L^2}{\left(\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|\right)^2}|\mathcal{N}_g|\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|^2 3\alpha^2\left(t_i^{u-1}\right)^2 + \frac{1}{\mathcal{U}}\sum_{u=1}^{\mathcal{U}}\frac{4L^2}{\left(\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|\right)^2}|\mathcal{N}_g|\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|^2\frac{1}{t_i^u}\sum_{l=0}^{t_i^u-1}\alpha^2 l^2 G^2 + \frac{12\alpha^2L^2G^2}{\left(\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|\right)^4}|\mathcal{N}_g|^2\sum_{i\in\mathcal{N}_g}|\mathcal{N}_i|^2\sum_{j\in\mathcal{N}_g\setminus\{i\}}|\mathcal{N}_j|^2. \quad (19)$$

Observe that the sync time S controls the upper bounds in the theorems above by statistically controlling the number of local iterations. Now let us assume that there exists a *minimum* local iteration time for group i, i.e., a lower bound:

$$\tau_{i,l}^u \geq c_i, \text{ a.s., } \forall l, u. \quad (20)$$

Then, one gets a *maximum* number of local iterations

$$t_i^u \leq t_i^{\max} \triangleq \left\lceil \frac{S}{c_i} \right\rceil, \text{ a.s., } \forall u. \quad (21)$$

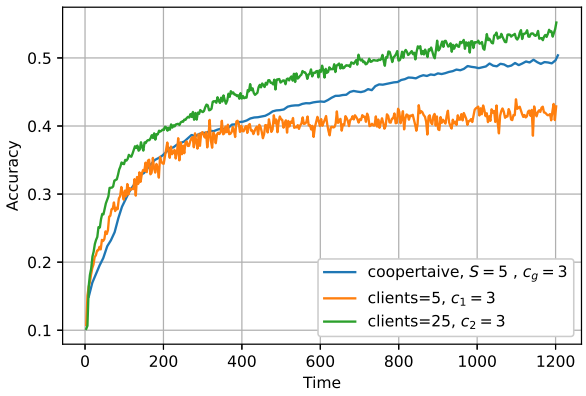


Fig. 3: Significance of group cooperation under non-i.i.d data.

Based on the above bound, one can get the following global convergence guarantee.

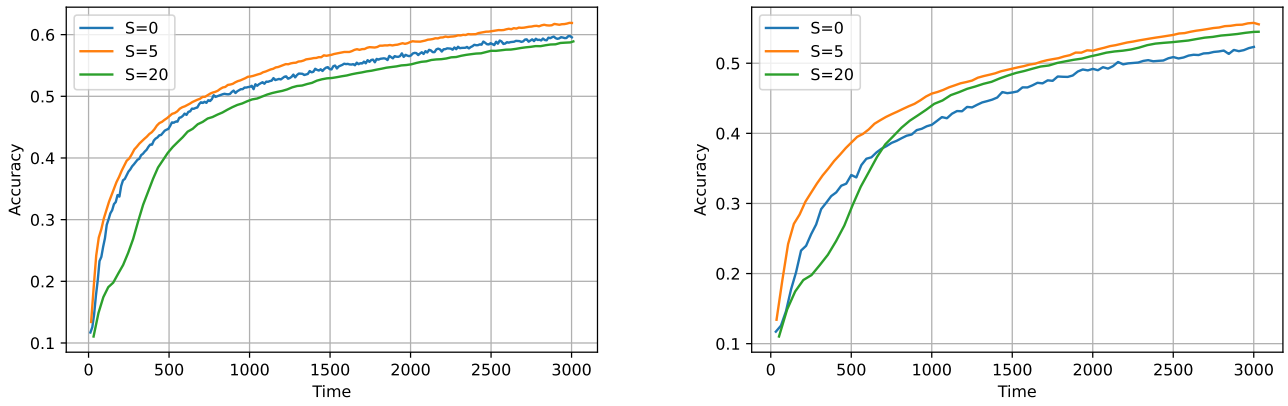
Corollary 1 (*Global Convergence Guarantee*) *For a given $\{t_i^{\max}\}$, setting $\alpha = \min\{\frac{1}{\sqrt{\mathcal{U}}}, \frac{1}{L}\}$, the delay sensitive HFL algorithm achieves $\frac{1}{\mathcal{U}}\sum_{u=1}^{\mathcal{U}}\mathbb{E}\left\|\nabla f\left(x^u\right)\right\|^2 = \mathcal{O}(\frac{1}{\sqrt{\mathcal{U}}})$.*

Therefore, for a finite sync time S, as the training time T increases, the number of the global communication rounds $\mathcal{U}$ also increases, and hence Corollary 1 shows that gradient converges to 0 sublinearly.

IV. Experiments

In this section, we present some simulation results for the proposed delay sensitive HFL algorithm to verify the findings from the theoretical analysis.

Datasets and Model. We consider an image classification supervised learning task on the CIFAR-10 dataset [7].A convolution neural network (CNN) is adopted with two 5x5



(a) $c_g = 10$ (b) $c_g = 30$

Fig. 4: Impact of the global shift parameter c_g on choosing the sync time S.

convolution layers, two 2x2 max pooling layers, two fully connected layers with 120 and 84 units, respectively, ReLu activation, a final softmax output layer and cross entropy loss.

Federated Learning Setting. Unless otherwise stated, we have 30 clients randomly distributed across 2 groups. The groups have similar data statistics. We consider shifted exponential delays [8]: $\tau^u_{i,l} \sim \exp(c_i, 10)$ and $\tau^u_g \sim \exp(c_g, 10)$.

Discussion. In Fig. 2, we show the evolution of both groups' accuracies and the global accuracy across time. The zoomed-in version in Fig. 2b shows the high (SGD) variance in the performance of the two groups especially during the earlier phase of training. Then, with more averaging with the GPS, the variance is reduced.

In Fig. 3, the significance of collaborative learning is emphasized. We run three experiments, one for each group in an isolated fashion, and one under the HFL setting. First, while we do not conduct our theoretical analysis under heterogeneous data distribution, we consider a non-iid data distribution among the two groups in this setting, and we see that our proposed algorithm still *converges*. Second, it is clear that the performances of the group with less number of clients under heterogeneous data distribution and isolated learning will be deteriorated. However, aided by HFL, its performance improves while the other group's performance is not severely decreased, which promotes *fairness* among the groups.

In Fig. 4, we show impact of the sync time S on the performance, by varying the GPS shift parameter c_g. We see that for $c_g = 10$, $S = 0$ outperforms $S = 20$. Note that $S = 0$ corresponds to a centralized system (non-hierarchical). Increasing the shift parameter to $C_g = 30$, however, the situation is different. Although in both figures $S = 5$ is the optimum choice, but in case the system has an additional constraint on communicating with the GPS, $S = 20$ will be a better choice, especially that the accuracy gain will not be sacrificed much. It is also worth noticing that the training time budget T plays a significant role in choosing S; in Fig. 4b, $S = 0$ (always communicate with the GPS) outperforms $S = 20$ as long as $T \leq 500$, and the opposite is true afterwards. This means that in some scenarios, the hierarchical setting may not be the optimal setting (which is different from the findings in [3]); for instance, if the system has a hard time constraint in learning, it may prefer to make use of communicating with GPS more frequently to get the advantage of learning the resulting models from different data.

V. Conclusion

A delay sensitive HFL algorithm has been proposed, in which the effects of wall-clock times and delays on the overall accuracy of FL is investigated. A sync time S governs how many local iterations are allowed at LPSs before forwarding to the GPS, and a system time T constrains the overall training period. Our theoretical and simulation findings reveal that the optimal S depends on different factors such as the delays at the LPSs and the GPS, the number of clients per group, and the value of T. Multiple insights are drawn on the performance of HFL in time-restricted settings.

References

[1] B. McMahan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. International Conf. Artificial Intelligence Statistics*, April 2017.

[2] J. Wang, S. Wang, R.-R. Chen, and M. Ji. Demystifying why local aggregation helps: Convergence analysis of hierarchical SGD. In *Proc. AAAI*, June 2022.

[3] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief. Client-edge-cloud hierarchical federated learning. In *Proc. IEEE ICC*, June 2020.

[4] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu. HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning. *IEEE Trans. Wireless Commun.*, 19(10):6535–6548, October 2020.

[5] A. Wainakh, A. S. Guinea, T. Grube, and M. Mühlhäuser. Enhancing privacy via hierarchical federated learning. In *Proc. IEEE EuroS&PW*, September 2020.

[6] H. Yu, S. Yang, and S. Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proc. AAAI*, January 2019.

[7] A. Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. Available online.

[8] B. Buyukates and S. Ulukus. Timely communication in federated learning. In *Proc. IEEE INFOCOM*, May 2021.