HIERARCHICAL FEDERATED LEARNING ACROSS HETEROGENEOUS CELLULAR NETWORKS

M. S. H. Abad,[†] *E. Ozfatura,*[+] *D. Gündüz*[+] *and O. Ercetin*[†]

[†] Faculty of Engineering and Natural Sciences, Sabanci University, Turkey
[+] Department of Electrical and Electronic Engineering, Imperial College London, UK

ABSTRACT

We consider federated edge learning (FEEL), where mobile users (MUs) collaboratively learn a global model by sharing local updates on the model parameters rather than their datasets, with the help of a mobile base station (MBS). We optimize the resource allocation among MUs to reduce the communication latency in learning iterations. Observing that the performance in this centralized setting is limited due to the distance of the cell-edge users to the MBS, we introduce small cell base stations (SBSs) orchestrating FEEL among MUs within their cells, and periodically exchanging model updates with the MBS for global consensus. We show that this hierarchical federated learning (HFL) scheme significantly reduces the communication latency without sacrificing the accuracy.

***Index Terms*—** Cellular networks, federated learning, mobile edge processing, resource allocation.

1. INTRODUCTION

Vast amounts of data is generated today by mobile devices, from smart phones to autonomous vehicles, drones, and various Internet-of-things (IoT) devices. Machine learning (ML) is key to exploiting these massive datasets to make intelligent inferences and predictions. Most ML solutions are centralized; that is, they assume that the data collected from edge devices is available at a central server. However, offloading these huge datasets to an edge or cloud server over wireless links is often not feasible due to latency, bandwidth, or privacy constraints. A recently proposed alternative approach is federated edge learning (FEEL) [1–4], which enables ML at the network edge without offloading any data.

Federated learning (FL) is a collaborative ML framework [5, 6], where random subsets of devices are selected in an offline manner to update model parameters based on local datasets. Local models are periodically averaged from participating devices either with the help of a parameter server or through device-to-device communications.

This work has been funded by the European Research Council (ERC) through project BEACON (No. 725731) and by by H2020-MSCA-ITN-2015 project SCAVENGE under grant number 677854.

Although the communication bottleneck of FL has been acknowledged in the ML literature, implementation of these techniques on wireless networks, particularly in heterogeneous cellular networks (HCNs), and the successful orchestration for a large scale learning problem has not been addressed.

Recently, FL with a particular focus on wireless communications has been studied in several papers [1–3, 7–12]. Nevertheless, to the best of our knowledge, this is the first work addressing the communication latency of FL framework implemented over a cellular network, via seeking the optimal resource allocation policy for the mobile users (MUs). Further, this is the only work to consider MU clustering among SBSs to reduce the communication latency by reducing link distances, and by enabling spatial reuse of resources. Although, hierarchical schemes have been studied in [13–15], wireless nature of the communication medium is not taken into account.

In this paper, we focus on FEEL across HCNs, and introduce a communication-efficient hierarchical FL (HFL) framework. In this framework, MUs with local datasets are clustered around small-cell base stations (SBSs) to perform federated stochastic gradient descent (SGD) with decentralized datasets, and these SBSs communicate with a macro-cell base station (MBS) periodically to seek consensus on the shared model of the corresponding ML problem. In order to further reduce the communication latency of this hierarchical framework, we utilize gradient sparsification, and introduce an optimal resource allocation scheme for synchronous gradient updates. Our contributions in this paper can be summarized as follows:

- We introduce a HFL framework for HCNs and provide a holistic approach for the communication latency with a rigorous end-to-end latency analysis.
- We employ communication efficient FEEL techniques, in particular, sparsification and periodic averaging, and design a resource allocation strategy to minimize the end-to-end latency.
- Finally, we demonstrate our results by studying FEEL over a HCN to classify images from the CIFAR10

dataset, and show that the communication latency can be reduced dramatically without sacrificing the accuracy much.

2. SYSTEM MODEL

We focus on a scenario where K clients collaborate on solving an optimization problem in the following form;

$$\min_{\theta\in\mathbb{R}^Q} f(\theta) = \frac{1}{K}\sum_{i=1}^{K} \underbrace{\mathbb{E}_{\zeta\sim\mathcal{D}_i} F_i(\theta,\zeta)}_{:=f_i(\theta)}, \tag{1}$$

where $f_i(\theta)$ and $\mathcal{D}_i$ are the local expected loss function and distribution of data at ith client, respectively, and θ is the Q-length parameter model to be learned. FL framework is designed to orchestrate K clients to solve the given optimization problem in (1) without sharing their dataset.

In the general form of the FL framework, the central entity orchestrating the collaborative learning (e.g., MBS) first chooses $C \leq 1$ portion of the clients at the beginning of each iteration and sends them the current model. Then, the chosen clients perform H number of local updates by employing SGD over their own dataset. Following H local updates, clients send back their updated model to the central entity, where the models are averaged to seek a consensus. This process iterates until certain convergence requirements are satisfied. The general FL framework is presented in Algorithm 1. In particular, when $C = 1$ it is referred as the *full participation* scenario and when $H = 1$ it is called *federated SGD (FedSGD)*. In the scope of this paper, we will focus on the full participation scenario where clients are MUs and FL is orchestrated by a MBS.

Algorithm 1 Federated learning (FL)

1: **for** $t = 1, 2, \ldots$ **do**
2: Choose a subset of clients $\mathcal{K}_t \subseteq \mathcal{K}$: $|\mathcal{K}_t| = K \times C$
3: **for** $k \in \mathcal{K}_t$ **do**
4: Pull θ_t: $\theta_{k,0} = \theta_t$
5: **for** $\tau = 1, \ldots, H$ **do** local update:
6: Compute SGD: $\mathbf{g}_{k,\tau} = \nabla_\theta F_k(\theta_{k,\tau-1}, \zeta_{k,\tau})$
7: Update model: $\theta_{k,\tau} = \theta_{k,\tau-1} - \eta_t \mathbf{g}_{k,\tau}$
8: Push $\theta_{k,H}$
9: **Federated Averaging**: $\theta_{t+1} = \frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t} \theta_{k,H}$

2.1. Communication latency

We assume that the bandwidth available for communication is B Hz. We employ an orthogonal access scheme with OFDM, and assign distinct subcarriers to MUs. Denote by $M = \frac{B}{B_0} \geq K$ the number of subcarriers, where B_0 is the subcarrier spacing. We denote the channel gain between MU k and MBS on subcarrier m by $\gamma_{k,m} = |h_{k,m}|^2$, $k = 1, \ldots, M$, where $h_{k,m}$ is the complex channel coefficient. The distance of MU k to MBS is denoted by d_k, and the path loss exponent by α.

2.1.1. Uplink (UL) Latency

For the latency analysis, we consider a set of stationary policies $\Pi(\pi_r, \pi_p)$ which are tuples of subcarrier and power allocation schemes, respectively. The resource allocation policy π_r divides the available subcarriers among the MUs disjointly, i.e., $\pi_r = \{\mathcal{M}_1, \ldots, \mathcal{M}_K : \mathcal{M}_k \cap \mathcal{M}_l = \varnothing,\ \forall k \neq l\}$. We consider a time slotted channel model with stationary and ergodic time-varying channel gain over slots, and we employ the fixed-rate transmission policy with truncated channel inversion power allocation [16]. According to this policy, given π_r, at any transmission slot τ MU k allocates power $p_{k,m}(\tau) \propto \frac{1}{\gamma_{k,m}(\tau)}$ on subcarrier $m \in \mathcal{M}_k$ if the channel gain is above a certain threshold, i.e., $\gamma_{k,m}(\tau) \geq \gamma_{k,m}^{th}$, otherwise does not use that subcarrier. Under average power constraint, we limit our focus to those power allocation policies π_p satisfying

$$\mathbb{E}_{\gamma_k(\tau)}\left[\sum_{m\in\mathcal{M}_k} p_{k,m}(\tau)\right] \leq P_{max},\ \forall\, k, \tau. \tag{2}$$

Therefore, the power allocation policy π_p boils down to the set of threshold vectors, i.e., $\pi_p = \{\gamma_k^{th}\}_{k=1}^K$. Then, for given policy Π, the transmission rate of MU k, at transmission slot τ can be written as $R_k(\tau, \Pi) = \sum_{m\in\mathcal{M}_k} r_{k,m}(\gamma_{k,m}(\tau), \gamma_{k,m}^{th})$, where $r_{k,m}(\gamma_{k,m}(\tau), \gamma_{k,m}^{th})$ is the instantaneous rate of MU k on subcarrier m at transmission slot τ. Accordingly, the UL latency of MU k, under policy Π can be defined as

$$T_k^{UL}(\Pi) = \min\{T : \sum_{\tau=1}^{T} R_k(\tau, \Pi) \geq \hat{Q}\}, \tag{3}$$

where $\hat{Q}$ is the number of bits used to represent the model. Due to synchronization, the overall average UL latency is determined according to the MU with the highest UL latency, i.e., $\bar{T}^{UL}(\Pi) = \mathbb{E}_{\gamma_1,\ldots,\gamma_K} \max\{T_1^{UL}, \ldots, T_K^{UL}\}$, and the optimal policy $\Pi^\star$ is the one that minimizes $\bar{T}^{UL}(\Pi)$. To find the optimal policy $\Pi^\star$, we assume that

$$\frac{\mathbb{E}_{\gamma_1,\ldots,\gamma_K} \min_k\{\sum_{\tau=1}^{T} R_k(\tau,\Pi)\}}{T} = \min_k \underbrace{\mathbb{E}_{\gamma_k} R_k(\Pi)}_{\triangleq \bar{R}_k}, \tag{4}$$

which is reasonable when T is large. Hence, finding the optimal policy $\Pi^\star$ is equivalent to solving the following optimization problem:

$$\max_{\Pi} \min_{k=1,\ldots,K} \bar{R}_k(\Pi). \tag{5}$$

Power allocation policy π_p depends on π_r, which allocates subcarriers, through (2). Given a π_r, $\pi_p^\star$ can be found as in [16]. Hence, to solve (5) we use the following sequential optimization framework where we first initialize a resource allocation policy π_r, then find the optimal conditional policy $\pi_p^\star|\pi_r$, and finally update the resource allocation policy π_r according to MU with the lowest expected UL rate, and repeat the process.

2.1.2. Downlink (DL) Latency

We assume that the MBS also uses $\hat{Q}$ bits to compress the global model. We employ a multicast policy and assume that the MBS allocates its available power uniformly over all the subcarriers. Let $SNR_{k,m}(\tau)$ denote the signal-to-noise ratio (SNR) of worker k on subcarrier m. The instantaneous multicast DL rate of MU k becomes:

$$R_k(\tau) = \sum_{m\in\mathcal{M}} B_0 \log_2(1 + SNR_{k,m}(\tau)), \tag{6}$$

where $SNR_{k,m}(\tau) = \frac{P_{max}\gamma_{k,m}(\tau)}{MN_0B_0d_k^\alpha}$. The multicast will terminate when $\hat{Q}$ bits are received by all the workers. The average multicast latency, T^{DL}, can be computed as follows:

$$\bar{T}^{DL} = \mathbb{E}\left[\max_k \min\{T : T_s \sum_{\tau=1}^{T} R_k(\tau) \geq \hat{Q}\}\right]. \tag{7}$$

Per iteration, the end-to-end latency of the FL protocol is given by $T^{FL} = \bar{T}^{UL} + \bar{T}^{DL}$.

3. HIERARCHICAL FEDERATED LEARNING (HFL)

In large scale networks, where many MUs distributed across a large cell and participate in FL, the communication latency may be prohibitively large due to the limited bandwidth and the weak channels of cell edge users. To this end we propose the hierarchical FL framework, where MUs are clustered according to their locations and seek a consensus on the model with the help of SBSs according to FedSGD framework. In particular, at each iteration, MUs in a cluster send their local gradient estimates to the assigned SBS for aggregation, and the SBSs send back the average of the received estimates to their associated MUs to update their model accordingly.

Distance based MU clustering not only reduces the communication distance, and hence the latency, but also allows the spatial reuse of available communication resources. On the other hand, limiting the gradient communications within clusters may prevent convergence to a single parameter model (i.e., global consensus). To this end, we combine the intra-cluster gradient aggregation method with inter-cluster model averaging strategy, such that after every H consecutive intra-cluster SGD iterations, SBSs send their local model updates to the MBS to establish a global consensus.

The HFL algorithm is presented in Algorithm 2. Denote by $\mathcal{C}_l$ the set of MUs belonging to cluster $l = 1, \ldots, L$, with L being the number of SBSs. During intra-cluster iterations, the local gradient estimates of the MUs are aggregated within the clusters. SBS l aggregates the gradients from the MUs in its cluster (line 6). This average is then sent back by the SBS to the MUs in its cluster, and the models at all clusters are updated. After H iterations, all SBSs transmit their models to the MBS through UL fronthaul links. The MBS calculates the model average (line 10), and multicasts it back to the SBSs over the DL fronthaul links. Upon receiving the model update, the SBSs share it with the MUs in their clusters. Hence, after H iterations all the MUs share a common parameter vector, globally.

Algorithm 2 HFL

1: **for** $t = 1, \ldots, T$ **do**
2: **for** $k \in \mathcal{K}$ **do**
3: Compute SGD: $\mathbf{g}_{k,t} = \nabla_\theta F_k(\theta_{k,t-1}, \zeta_{k,t})$
4: **Execute FedSGD in the clusters**:
5: **for** $l = 1, \ldots L$ **do**
6: $\mathbf{g}_{l,t} = \frac{1}{|\mathcal{C}_l|}\sum_{k\in\mathcal{C}_l} \mathbf{g}_{k,t}$.
7: Update model at SBS: $\theta_{l,t} = \theta_{l,t-1} - \eta_t \mathbf{g}_{l,t}$
8: **if** $t \mid H$ **then**
9: **Execute FedAvg among the clusters**:
10: Update model at MBS: $\theta_t = \frac{1}{L}\sum_{l=1}^{L} \theta_{l,t}$
11: SBS pulls model from MBS: $\theta_{l,t} = \theta_t$
12: Client pulls model from SBS: $\theta_{k,t} = \theta_{l,t}$

We assume that there is no interference between MUs located more than D_{th} from each other. Clusters are colored so that any two clusters with the same color are separated by at least distance D_{th} to minimize the interference between clusters. If N_c colors are used in total, the available OFDM subcarriers are divided into N_c groups, and the subcarriers in each group are allocated to clusters with a particular color. Consequently, the number of available OFDM subcarriers for each clusters is approximately M/N_c.

In the local gradient update step of HFL communication latency analysis follows similarly to the one for centralized FL in Sec. 2 with the number of subcarriers M/N_c. Denote by Γ_l^U, and Γ_l^D, the UL and DL latencies in cluster l, respectively. After H iterations, SBSs send their model updates to the MBS. Let U^{SBS}, R^{SBS} be the UL and DL rates of SBSs to the MBS, respectively. The UL and DL latencies at each period of H iterations become $\Theta^U = \frac{\hat{Q}}{U^{SBS}}$ and $\Theta^D = \frac{\hat{Q}}{R^{SBS}}$, respectively. There is also the latency of transmitting the average model by SBSs to their associated MUs. The average latency associated with one period of HFL becomes $\Gamma^{per} = \max_{l\in\mathcal{L}} H\left(\Gamma_l^U + \Gamma_l^D\right) + \Theta^U + \Theta^D + \max_l \Gamma_l^D$, and the average per iteration latency of HFL is $\Gamma^{HFL} = \frac{\Gamma^{per}}{H}$.

Table 1: Latency comparison between HFL and centralized FL as a function of the path-loss exponent α and H.

	$H=2$	$H=4$	$H=6$
$\alpha=2.7$	5.6	6.4	6.6
$\alpha=3.1$	16	17.5	18.5
$\alpha=3.5$	35	39.5	41.5

4. NUMERICAL RESULTS

To further improve the performance of the HFL framework, we use the *momentum* SGD for the local FedSGD and utilize sparsification strategy to reduce the communication load. We implement the FedSGD for clusters following the guidelines in [17].

We consider 28 MUs uniformly distributed across a circular area with radius 750 meters. We consider hexagonal clusters, where the diameter of circle inscribed in each is 500 meters. The SBSs are located at the center of the hexagons. We assume that the fronthaul links are 100 times faster than the UL and DL between MUs and SBSs. The total number of clusters is 7. We assume 600 subcarriers with subcarrier spacing of 30 KHz. The maximum transmit powers of the MBS, SBSs, and the MUs are 20W, 6.3W, and 0.2W, respectively [18].

In our numerical analysis, we consider the image classification problem over the CIFAR-10 dataset with 10 different image classes [19], and train the ResNet18 architecture [20] in a federated manner. The dataset is divided across MUs in an independent and identically distributed (IID) manner. We utilize some large batch training tricks, such as scaling the learning rate η and employing a warm-up phase [21]. We set the batch size for training to $\beta = 64$. We set the initial learning rate to 0.25, and consider the first 5 epochs as the gradual warm-up phase where training starts with $\eta = 0.1$, which is increased linearly at each iteration until it reaches the initial learning rate. Following the guidelines in [13], we train the network for 300 epochs, and at the end of the 150th epoch we drop the initial learning rate by a factor of 10; and similarly, after the 225th epoch we drop the learning rate by another factor of 10. Finally, we apply 99%, 90%, 90% and 90% sparsification for MU UL, MU DL, SBS UL, and SBS DL, respectively.

We first study the reduction in latency achieved by HFL compared to centralized FL. We measure the latency reduction by comparing the latency of HFL, Γ^{HFL}, with that of centralized FL, T^{FL}. In particular, we evaluate the latency improvement factor defined as $\frac{T^{FL}}{\Gamma^{HFL}}$.

Clustering reduces the communication distance, and as a result, improves the SNR. The amount of improvement depends on the amount of reduction in path-loss. In Table 1, the latency improvement factor for different H and path-loss exponent (α) values is shown. We can observe that HFL brings significant improvement in terms of latency especially when the path-loss exponent is high, i.e., in urban environments, where we also expect most FL applications to take place.

Table 2: Top 1 accuracy results for different strategies.

Baseline	92.48 ± 0.13
FL	89.23 ± 0.42
HFL, $H=2$	90.27 ± 0.11
HFL, $H=4$	90.474 ± 0.20
HFL, $H=6$	91.03 ± 0.19

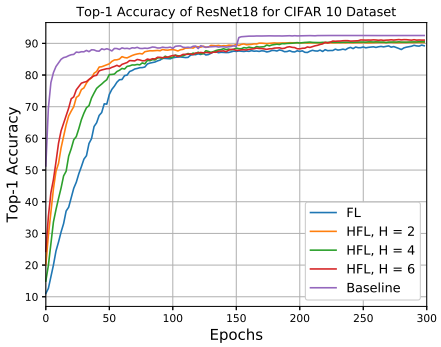

Fig. 1: Top-1 accuracy

The convergence of top-1 accuracy achieved by centralized FL and HFL algorithms are shown in Fig. 1. We observe that the latency improvement of HFL over centralized FL does not compromise its accuracy. In fact, a closer look at the accuracy (averaged over 5 runs) in Table 2 shows that HFL is able to achieve higher accuracy than centralized FL in all the cases. The *mean* $\pm$ *standard deviation* results for the last epoch is reported in Table 2, where the *Baseline* result is obtained by training a single network on the whole training set. We observe only a small degradation in the accuracy of HFL with respect to this bound. We believe that this degradation is mainly due to the use of momentum SGD at each MU instead of a global momentum and due to sparsification.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we introduced an analyzed a HFL framework implemented across a HCN. We provide a complete end-to-end latency analysis for the communication steps including both UL and DL phases. Then using this result we showed that hierarchical framework can speed up the training by reducing the communication latency. As a future extension of this work we are planning to study to the non-IID data distribution scenario [15, 22, 23].

6. REFERENCES

[1] M. Mohammadi Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," 2019.

[2] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding, "Federated learning via over-the-air computation," 2018.

[3] Guangxu Zhu, Yong Wang, and Kaibin Huang, "Low-latency broadband analog aggregation for federated edge learning," 2018.

[4] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah, "Wireless network intelligence at the edge," *CoRR*, vol. abs/1812.02858, 2018.

[5] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 20–22 Apr 2017, vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR.

[6] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konecný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander, "Towards federated learning at scale: System design," 2019.

[7] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Communicating or computing over the MAC: Function-centric wireless networks," *IEEE Transactions on Communications*, pp. 1–1, 2019.

[8] Deniz Gündüz, Paul de Kerret, Nicholas D. Sidiropoulos, David Gesbert, Chandra Murthy, and Mihaela van der Schaar, "Machine learning in the air," 2019.

[9] Jin-Hyun Ahn, Osvaldo Simeone, and Joonhyuk Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," 2019.

[10] M. Mohammadi Amiri, T. M. Duman, and D. Gündüz, "Collaborative machine learning at the wireless edge with blind transmitters," 2019.

[11] Jinke Ren, Guanding Yu, and Guangyao Ding, "Accelerating DNN training in wireless federated edge learning system," *CoRR*, vol. abs/1905.09712, 2019.

[12] N. H. Tran, W. Bao, A. Zomaya, N. Minh N.H., and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019*, April 2019, pp. 1387–1395.

[13] Tao Lin, Sebastian U. Stich, and Martin Jaggi, "Don't use large mini-batches, use local SGD," 2018.

[14] Fan Zhou and Guojing Cong, "A distributed hierarchical SGD algorithm with sparse global reduction," 2019.

[15] Lumin Liu, Jun Zhang, S. H. Song, and Khaled Ben Letaief, "Edge-assisted hierarchical federated learning with non-iid data," *CoRR*, vol. abs/1905.06641, 2019.

[16] A. J. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1218–1230, Oct 1997.

[17] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *International Conference on Learning Representations*, 2018.

[18] Wieslawa Wajda, Gunther Auer, Per Skillermark, and Ylva Jading, "Infso-ict-247733 earth deliverable d 2 . 3 energy efficiency analysis of the reference systems , areas of improvements and target breakdown," 2012.

[19] Alex Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[21] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He, "Accurate, large minibatch SGD: training imagenet in 1 hour," 2017.

[22] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra, "Federated learning with non-iid data," *CoRR*, vol. abs/1806.00582, 2018.

[23] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek, "Robust and communication-efficient federated learning from non-iid data," *CoRR*, vol. abs/1903.02891, 2019.