2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) | 978-1-6654-6975-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/WIMOB55322.2022.9941355

Hierarchical Federated Learning for the Next Generation IoT

Merkourios Simos*, Pavlos S. Bouzinis*, Panagiotis D. Diamantoulakis*,
Panagiotis Sarigiannidis†, and George K. Karagiannidis*
*Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece
†Department of Informatics and Telecommunications Engineering, University of Western Macedonia, 501 00 Kozani, Greece
e-mails: {merkours, mpouzinis, padiaman}@auth.gr, psarigiannidis@uowm.gr, geokarag@auth.gr

***Abstract*—Federated Learning is a promising decentralized machine learning approach, which has the potential to realize the vision of next-generation internet-of-things (NGIoT), by offering intelligent services and meeting the privacy and low latency requirements. By leveraging the combination of edge servers, as intermediate model aggregators, and the central cloud server, as global model aggregator, the concept of Hierarchical Federated Learning (HFL) has recently emerged. In this paper, we aim to minimize the delay of a global HFL round, under user energy requirements. We jointly optimize the computation and communication resources, as well as the user-edge assignment, in order to minimize the overall delay. The formulated non-convex combinatorial problem, is optimally solved by being decomposed into two disjoint subproblems, namely the resource allocation and user-edge assignment. Finally, the simulation results demonstrate the effectiveness of the proposed methods in terms of delay reduction, compared to selected benchmarks, while insights for the networks' behavior are provided.**

***Index Terms*—Hierarchical federated learning, next-generation Internet-of-Things**

I. Introduction

Recent advances in artificial intelligence (AI) and big data have placed machine learning (ML) at the core of technological development. Coupled with the exponential increase of smart devices and the potential of next-generation wireless systems, ML and the internet-of-things (IoT) have converged to enable the evolution of disruptive data-driven services [2]. Until now, ML schemes are centralized, exploiting the massive supply of available training data. However, the transmission of huge datasets has proven impractical due to latency, and privacy concerns over the sensitivity of user-generated data have created an incentive against data centralization.

Driven by such demands, distributed ML is getting widespread attention from both industry and academia, with the most popular approach being Federated Learning (FL) framework [3]. FL is privacy-preserving and decentralized, and has the potential of being incorporated into next-generation IoT (NGIoT) technology. Among others, some foundational challenges of NGIoT in the context of intelligent services, is the real-time decision making and the enhanced privacy provision [4]. The prominent characteristic of FL is that devices do not share their raw data with any third party, e.g., a cloud server. Instead, a shared model is collaboratively and distributively trained by the users, while the server regularly updates and redistributes the global model to the learners. The original implementation of FL is cloud-based, which enables the participation of millions of devices, matching the model accuracy of traditional ML frameworks. However it may suffer from inevitable inefficiencies in the communication with the server, owing this to high data traffic and network congestion [5]. On the other hand, by leveraging the computation capabilities of Mobile Edge Computing (MEC) [6], edge-based FL builds on the emergence of reliable, low-latency edge networks to achieve quick training, although producing less accurate models, due to limited user participation [7].

The above implementations point to an evident tradeoff between computational efficiency and model performance. However, these two goals are not mutually exclusive. Indeed, a hybrid edge-cloud scheme, called Hierarchical Federated Learning (HFL) has shown great potential in achieving both large-scale data access and efficient wireless communication [8], [9]. In the considered architecture, users train the learning model locally, and subsequently send the training parameters to the edge servers which perform edge-based federated averaging. A further layer of model averaging is then implemented on top of the edge network, via fronthaul links from the edge servers to a centralized cloud server, which generates the global training model. In this manner, multiple clusters of users are enabled to participate simultaneously, leading to a distributed federated learning framework, which promises to meet the demands of NGIoT, by providing reduced communication latency, scheduling flexibility, enhanced scalability and robust, accurate models.

Although several works have examined and optimized the performance of FL in a wireless environment [10], [11], HFL is still in its infancy. Previous works have explored various challenges in the application of HFL. For instance, authors in [8], examined the concept of HFL, by proving the convergence of the training algorithm and demonstrating its fundamental benefits compared to existing FL schemes. Moreover, in [12], the problem of user-edge assignment in the presence of non-IID data was investigated, aiming to enhance the perfor-

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957406.

A part of the results leading to this work, was based on the diploma thesis of M. Simos [1].

mance of FL, while in [13], a communication-efficient HFL framework for heterogenous cellular networks was introduced. Furthermore, authors in [14] proposed a resource allocation scheme for a cost-minimization problem in the HFL scenario, in terms of latency and energy cost. In more detail, in [14], a suboptimal solution for the user-edge association problem was proposed, without taking into account though the individual energy limitations of the users.

Although the aforementioned works examined various aspects of the HFL implementation and optimization, there are still open issues that need to be addressed in the context of HFL. Firstly, ensuring decreased delay during the HFL procedure is of paramount importance, in order to guarantee fast convergence of the global training model [15], [16]. Secondly, when aiming towards decreased delay tolerance of wireless devices, their energy constraints, which are not taken into account in [14], become critical and also lie in the objectives of NGIoT. Hence, a joint optimization of the available radio and computing resources should be performed, for achieving reduced latency and energy savings, which was not considered in [8], [12], [13]. In addition to this, the user-edge assignment in HFL is of significant importance, since it can highly affect the performance in terms of latency and energy consumption. Therefore, the user-edge association should be tactfully conducted, while certain devices may be excluded for participation into the FL process.

Driven by the aforementioned motives, we propose an efficient scheme for minimizing the total delay during a HFL round, subject to user energy requirements. We formulate and solve the problem of minimizing the total latency of a HFL round, by jointly optimizing the computational and communication resources and user-edge assignment. To solve this demanding non-convex combinatorial optimization problem, we decompose it two disjoint subproblems. At first, we derive the optimal resource allocation of users for each user-edge association. Following that, we obtain the optimal user-edge assignment which minimizes the total delay of a HFL round. Simulation results are provided, which illustrate the effectiveness of the proposed HFL scheme. More specifically, the demonstrated results point to an increased performance of the proposed scheme, comparing with various benchmarks, in terms of delay reduction. To this end, insights on the selection of specific design parameters are presented.

II. System Model

A. Hierarchical FL Model

We consider a HFL system, which consists of N wireless users, indexed as $n \in \mathcal{N} = \{1, 2, ..., N\}$ and S distributed edge servers (ES) indexed as $s \in \mathcal{S} = \{1, 2, ..., S\}$, which connect to a central cloud server via fronthaul links. Each user collects a local set of data samples, denoted by $\mathcal{D}_n = \{\boldsymbol{x}_{n,d}, y_{n,d}\}_{d=1}^{D_n}$, where $D_n = |\mathcal{D}_n|$ are the total data samples, $\boldsymbol{x}_{n,d}$ is an input vector and $y_{n,d}$ is the corresponding output. The trained model is described by the parameter vectors $\boldsymbol{w}$, while the accuracy of the model is measured with the use of a loss function $f(\boldsymbol{w}, \boldsymbol{x}, y)$. In general, the goal of the training process, is to find the global model parameter $\boldsymbol{w}$, which minimizes the loss function on the whole data set and is given by

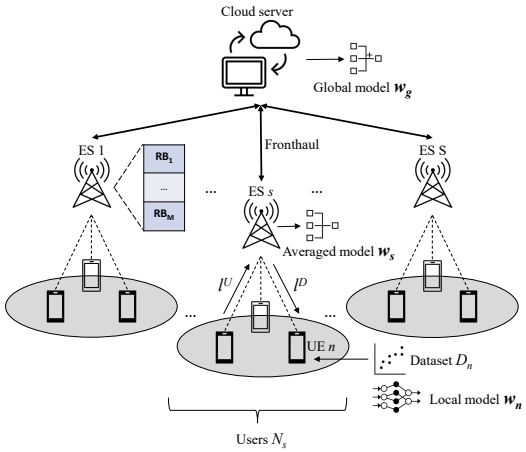


Fig. 1. Hierarchical Federated Learning system model.

$$F(\boldsymbol{w}) = \frac{\sum_{n=1}^{N} D_n f_n(\boldsymbol{w})}{\sum_{n \in \mathcal{N}} D_n}, \tag{1}$$

where $f_n(\boldsymbol{w})$ is the local loss function on the dataset $\mathcal{D}_n$, defined as $f_n(\boldsymbol{w}) \triangleq \frac{1}{D_n} \sum_{d \in \mathcal{D}_n} f(\boldsymbol{w}, \boldsymbol{x}_{n,d}, y_{n,d}), \ \forall n \in \mathcal{N}$.

The Hierarchical FL model consists of two communication layers. The first layer, defined as the *local layer*, refers to the links between the participating users and the ESs, and is in essence, the conventional FL model. The second layer, which we will call the *global layer*, describes the communication among the ESs and the cloud server. A description of these layers is presented below.

1) Local Layer: We consider a single ES s and a subset $\mathcal{N}_s \subseteq \mathcal{N}$ of users connected to it. We assume that M is the total number of the available orthogonal RBs that each ES can provide. Therefore, $\mathcal{N}_s$ contains M users, i.e., $|\mathcal{N}_s| = M$. Moreover, the subsets $\mathcal{N}_s$ are non-overlapping, i.e., $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset, \forall i, j \in \mathcal{S}, i \neq j$. Hence, each user is connected to a single ES. Firstly, an arbitrary set of initial parameters $\boldsymbol{w}_s$ is selected by each ES. Each ES broadcasts $\boldsymbol{w}_s$ to its connected users. Each user receives the parameters, and performs K_1 steps of the gradient descent method using the local dataset $\mathcal{D}_n$. The model update of the n-th user, during the j-th step, is described as $\boldsymbol{w}_n^{(j)} = \boldsymbol{w}_n^{(j-1)} - \eta \nabla f_n \left(\boldsymbol{w}_n^{(j-1)} \right), \quad j = 1, ..., K_1$, where η denotes the learning rate. After K_1 steps of gradient descent, i.e., K_1 local iterations, each user transmits the local parameters $\boldsymbol{w}_n \triangleq \boldsymbol{w}_n^{(K_1)}$ to the connected ES, which computes the weighted model average, as

$$\boldsymbol{w}_s = \frac{\sum_{n \in \mathcal{N}_s} D_n \boldsymbol{w}_n}{D_s}, \tag{2}$$

where D_s is the sum of the data by all users connected to the s-th ES, i.e., $D_s = \sum_{n \in \mathcal{N}_s} D_n$. Finally, the users' local computation and parameter transmission processes, along with the model aggregation at the ESs, are repeated for K_2 times, meaning that K_2 parameter transmissions are conducted between the users and the ESs.

2) Global Layer: Following the completion of the local layer procedure by each ES, the locally aggregated model parameters are forwarded to the cloud server via the fronthaul links. At this stage, the server calculates the global model parameter $\boldsymbol{w}_g$ by weighted averaging, described by the expression

$$\boldsymbol{w}_g = \frac{\sum_{s=1}^{S} D_s \boldsymbol{w}_s}{D}, \tag{3}$$

where $D = \sum_{s\in\mathcal{S}} D_s$, is the sum of the data samples' size collected by all ESs. Subsequently, the updated model is channeled via the fronthaul links back to the ESs, which updates their training parameters accordingly. The entire process is repeated for an arbitrary number of rounds, until the global model achieves a required accuracy. Upon convergence, the global parameters $\boldsymbol{w}_g$ are broadcast one last time to all users, concluding the HFL scheme.

B. Computation and Communication Model

1) Overview: Hereinafter, we focus on a single global HFL round, while the following analysis can be similarly carried out for any arbitrary round. As mentioned previously, each ES can serve a finite number of users, determined by the number of its available orthogonal RBs, which we consider equal to M. We assume that the total number of users N, is greater than the overall number of RBs among all ESs, i.e., $N > S \times M$. Hence, only a subset of the users can be scheduled for participation in the FL task. Moreover, a user may communicate with at most one ES per round, and may use only one RB for the message transmission. This is modelled through the resource block vector $\boldsymbol{a}_n = [a_{n,1}, \ldots, a_{n,S}]$, $\forall n \in \mathcal{N}$, where $a_{n,s} \in \{0, 1\}$ is a binary variable, indicating whether a resource block of the s-th ES is allocated to the n-th user. Following that, since each user may connect to at most one ES per round, it holds that $\sum_{s=1}^{S} a_{n,s} \leq 1$, $\forall n \in \mathcal{N}$. In addition to this, since each ES serves M users, it also holds that $\sum_{n=1}^{N} a_{n,s} = M$, $\forall s \in \mathcal{S}$. In Fig. 1., the HFL system model is illustrated. Finally, we assume that the user channel gains remain unchanged during a HFL round, while they can be accurately estimated by the ESs.

2) Computation & Transmission: After receiving the initial FL model parameters $\boldsymbol{w}_g$ from the cloud server, each ES sets $\boldsymbol{w}_s \leftarrow \boldsymbol{w}_g$, and broadcast them to its connected users. The transmission delay over the downlink between the n-th user and the s-th ES is expressed as

$$l_{n,s}^{\mathrm{D}} = \frac{Z}{B_{\mathrm{d}} \log_2\left(1 + \frac{P_{\mathrm{d}} h_{n,s}}{B_{\mathrm{d}} N_0}\right)}, \quad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{4}$$

where Z is the number of bits required to transmit $\boldsymbol{w}_s$, P_{d} is the downlink power of the ESs and N_0 is thermal noise. The downlink bandwidth is denoted as B_{d} while $h_{n,s}$ is the channel gain between the n-th user and the s-th ES. Channel gain is given as $h_{n,s} = |g_{n,s}|^2 d_{n,s}^{-\mathrm{PL}}$, where the complex random variable $g_{n,s} \sim \mathcal{CN}(0, 1)$ is the small scale fading, while $d_{n,s}$ is the distance between the n-th and the s-th ES, and PL denotes the path loss exponent. It is noted that, for ease of presentation, we assume flat fading channels, i.e., $h_{n,s}$ stays unchanged across all RBs of the s-th ES. However, the analysis can be easily extended for the case of frequency selective channels.

In the local layer, participating users perform K_1 steps of local training, and subsequently transmit their local model parameters to their assigned ES. The computational delay per local iteration can be expressed as

$$l_n^{\mathrm{C}} = \frac{\omega_n D_n}{\vartheta_n}, \quad \forall n \in \mathcal{N}, \tag{5}$$

with ϑ_n being the frequency of the n-th users' CPU clock speed, while ω_n denotes the number of CPU cycles required per bit data. Moreover, the communication delay during the uplink parameter transmission of the n-th user to the s-th ES, can be written as

$$l_{n,s}^{\mathrm{U}} = \frac{Z}{B_{\mathrm{u}} \log_2\left(1 + \frac{P_n h_{n,s}}{B_{\mathrm{u}} N_0}\right)}, \quad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{6}$$

where P_n is the n-th users' transmit power, while we assume that each user utilizes the same uplink bandwidth B_{u} per RB.

Following that, each ES collects the trained parameters and aggregates them. After K_2 iterations of the local HFL layer, each ES transmits its model parameters to the cloud server. Subsequently, the cloud server generates the updated global model and broadcasts it back to the ESs. We assume that the fronthaul link capacity is much higher than that of the wireless medium. Thus, the communication delay among the central server and the ESs is considered negligible. Moreover, we ignore the computational delay caused by the ESs and the central server, since they are equipped with powerful computational capabilities, while also the model averaging is not a computationally-intensive task. As a result, the n-th user's total delay in a HFL round is given by

$$l_n = \sum_{s=1}^{S} a_{n,s} l_{n,s}, \quad \forall n \in \mathcal{N}, \tag{7}$$

where

$$l_{n,s} = K_2 \left(l_{n,s}^{\mathrm{D}} + K_1 l_n^{\mathrm{C}} + l_{n,s}^{\mathrm{U}}\right), \quad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{8}$$

and represents the total delay of the n-th user, given that is connected to the s-th ES. It consists of the downlink delay for receiving the model parameters by the connected ES, the computation delay, for executing K_1 local iterations and the uplink delay, for transmitting the trained parameters to the ES.

3) Energy Consumption: The n-th user's total energy consumption for a HFL round is given by

$$e_n = K_2 \left(K_1 \varsigma \omega_n \vartheta_n^2 D_n + E_n\right), \quad \forall\, n \in \mathcal{N}, \tag{9}$$

with ς being the energy consumption coefficient related with the hardware architecture of each device, $E_n = P_n l_n^{\mathrm{U}}$, and $l_n^{\mathrm{U}} = \sum_{s=1}^{S} a_{n,s} l_{n,s}^{\mathrm{U}}$. It is noted that the first term of (9) represents the energy consumption dedicated for computation purposes, while E_n denotes the consumed energy related with the model parameter transmission to the ESs.

III. HFL Round Delay Minimization

In accordance with the low-latency requirements of NGIoT, our objective goal is to minimize the duration of a HFL round. Since the server has to wait for all users to terminate the transmission of their parameters, the total latency per HFL round is exclusively determined by the slowest user. Therefore, the total/global delay of a HFL round can be written as

$$l_{\mathrm{g}} = \max_{n \in \mathcal{N}} l_n. \tag{10}$$

A. Problem Formulation

We now proceed to the formulation of the optimization problem for minimizing the total delay during a HFL round l_{g}, which can be written as

$$\min_{\boldsymbol{A},\boldsymbol{\vartheta},\boldsymbol{E}} \max_{n \in \mathcal{N}} \left(\sum_{s=1}^{S} a_{n,s} l_{n,s} \right) \tag{11}$$

$$\textbf{\textit{s.t.}}\ a_{n,s} \in \{0,1\}, \quad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{11a}$$

$$\sum_{s=1}^{S} a_{n,s} \leq 1, \quad \forall n \in \mathcal{N}, \tag{11b}$$

$$\sum_{n=1}^{N} a_{n,s} = M, \quad \forall s \in \mathcal{S}, \tag{11c}$$

$$K_2 \left(K_1 \varsigma \omega_n \vartheta_n^2 D_n + E_n \right) \leq e_{\max}, \quad \forall n \in \mathcal{N}, \tag{11d}$$

$$0 \leq \vartheta_n \leq \vartheta_{\max}, \quad E_n \geq 0, \quad \forall n \in \mathcal{N}, \tag{11e}$$

where $l_{n,s}$ is given by (8), $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_N]$ is a matrix containing all user allocation vectors, $\boldsymbol{\vartheta}, \boldsymbol{E}$ are the vectors related with the CPU clock speed and the transmission energy of each user respectively, while we replace P_n with $P_n = \frac{E_n}{l_n^{\mathrm{U}}}$. Constraints (11b) and (11c) describe the properties of the user allocation vectors, while (11e) and (11d) define the maximum available clock speed $\vartheta_{\max}$, and total energy $e_{\max}$ of each user.

B. Proposed Solution

The problem in (11) is a non-convex combinatorial problem, which is generally hard to solve. Our analysis decouples it into disjoint sub-problems which are described below.

1) Individual user-ES delay minimization: First, we focus on minimizing $l_{n,s},\ \forall n \in \mathcal{N}, \forall s \in \mathcal{S}$, aiming to obtain the minimum delay of an individual user, for each possible user-ES matching. It should be highlighted, that the minimization of the delay among the n-th user and the s-th ES, i.e., $l_{n,s}$, is not subject to the residual users' resources. Thus, it can be individually performed for each $n \in \mathcal{N}, s \in \mathcal{S}$. Following that, in order to properly formulate and manipulate the optimization problem, we relax the equality in (6) to an inequality constraint, yielding

$$l_{n,s}^{\mathrm{U}} \geq \frac{Z}{B_{\mathrm{u}} \log_2 \left(1 + \frac{E_n h_{n,s}}{l_{n,s}^{\mathrm{U}} B_{\mathrm{u}} N_0} \right)}, \quad \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{12}$$

while we consider $l_{n,s}^{\mathrm{U}}$ as an optimization variable. Hence, the considered individual user-ES delay minimization problem can be formulated $\forall n \in \mathcal{N}, s \in \mathcal{S}$, as follows:

$$\min_{l_{n,s}^{\mathrm{U}}, \vartheta_n, E_n} l_{n,s} \tag{13}$$

$$\textbf{\textit{s.t.}}\ l_{n,s} = K_2 \left(\frac{Z}{B_{\mathrm{d}} \log_2 \left(1 + \frac{P_{\mathrm{d}} h_{n,s}}{B_{\mathrm{d}} N_0} \right)} + l_{n,s}^{\mathrm{U}} + K_1 \frac{\omega_n D_n}{\vartheta_n} \right), \tag{13a}$$

$$Z - l_{n,s}^{\mathrm{U}} B_{\mathrm{u}} \log_2 \left(1 + \frac{E_n h_{n,s}}{B_{\mathrm{u}} N_0 l_{n,s}^{\mathrm{U}}} \right) \leq 0, \tag{13b}$$

$$K_2 \left(K_1 \varsigma \omega_n \vartheta_n^2 D_n + E_n \right) \leq e_{\max}, \tag{13c}$$

$$0 \leq \vartheta_n \leq \vartheta_{\max},\ E_n \geq 0, \tag{13d}$$

where (13b) is related with (12). It is easy to verify that the problem in (13) is jointly convex with respect to $l_{n,s}^{\mathrm{U}}, \vartheta_n, E_n$, since the objective function and the constraints are convex. However, the derivation of explicit closed-form solutions is prevented, due to the nature of the problem. In order to efficiently solve (13), we first consider a fixed value of $\vartheta_n = \bar{\vartheta}_n$. Therefore, the problem in (13) can be transformed, $\forall n \in \mathcal{N}, \forall s \in \mathcal{S}$, as

$$\min_{l_{n,s}^{\mathrm{U}}, E_n} l_{n,s}^{\mathrm{U}} \tag{14}$$

$$\textbf{\textit{s.t.}}\ l_{n,s}^{\mathrm{U}} B_{\mathrm{u}} \log_2 \left(1 + \frac{E_n h_{n,s}}{B_{\mathrm{u}} N_0 l_{n,s}^{\mathrm{U}}} \right) \geq Z, \tag{14a}$$

$$K_2 \left(K_1 \varsigma \omega_n \bar{\vartheta}_n^2 D_n + E_n \right) \leq e_{\max}, \tag{14b}$$

$$E_n, l_{n,s}^{\mathrm{U}} \geq 0, \tag{14c}$$

The optimal $l_{n,s}^{\mathrm{U}}, E_n$, can be obtained as in the following lemma.

Lemma 1: The optimal E_n, l_n^{U} of problem (14), for a fixed $\vartheta_n = \bar{\vartheta}_n$, are given by

$$E_n^*(\bar{\vartheta}_n) = \frac{e_{\max}}{K_2} - K_1 \varsigma \omega_n \bar{\vartheta}_n^2 D_n, \tag{15}$$

and

$$l_{n,s}^{\mathrm{U}*}(\bar{\vartheta}_n) = -\frac{\beta}{\frac{\beta}{\alpha} + \mathcal{W}\left(-\frac{\beta}{\alpha} e^{\frac{\beta}{\alpha}} \right)}, \tag{16}$$

where $\alpha = \frac{h_{n,s} E_n^*(\bar{\vartheta}_n)}{B_{\mathrm{u}} N_0}$, $\beta = \frac{Z \ln(2)}{B_{\mathrm{u}}}$, and $\mathcal{W}(\cdot)$ is the Lambert W function.

Proof: Firstly, since $\log_2(\cdot)$ is an increasing function with respect to E_n, while the objective is to minimize $l_{n,s}^{\mathrm{U}}$, the constraint in (14b) should be satisfied with equality, leading to

$$E_n^*(\bar{\vartheta}_n) = \frac{e_{\max}}{K_2} - K_1 \varsigma \omega_n \bar{\vartheta}_n^2 D_n, \tag{17}$$

which implies that users utilize their whole available energy. Following that, it is straightforward to show that the function

$$f(l_{n,s}^{\mathrm{U}}) = l_{n,s}^{\mathrm{U}} B_{\mathrm{u}} \log_2 \left(1 + \frac{E_n h_{n,s}}{B_{\mathrm{u}} N_0 l_{n,s}^{\mathrm{U}}} \right), \quad \forall l_{n,s}^{\mathrm{U}} \geq 0, \tag{18}$$

is an increasing function with respect to $l_{n,s}^{\mathrm{U}}$. Thus, since the goal is to minimize $l_{n,s}^{\mathrm{U}}$, the optimal $l_{n,s}^{\mathrm{U}*}$ is given when the constraint (14a) is satisfied with equality, as initially implied by (6). After some mathematical manipulations, we conclude to (16) and the proof is completed. ■

TABLE I
SIMULATION SETTINGS

Parameter	Value
Number of ESs	$S = 4$
Number of users	$N = 50$
RBs per ES	$M = 5$
Learning rate	$\eta = 0.001$
Bandwidth	$B_{\mathrm{d}} = B_{\mathrm{u}} = 100$KHz
Downlink power	$P_{\mathrm{d}} = 10$ W
Thermal noise	$N_0 = -174$ dBm/Hz
CPU cycles per bit	$\omega = 10$
Energy consumption coef.	$\varsigma = 10^{-27}$
Maximum CPU speed, $\forall n \in \mathcal{N}$	$\vartheta_{\max} = 2$ GHz

Following this analysis, the optimal $l^*_{n,s}(\bar{\vartheta}_n)$ is given by

$$l^*_{n,s}(\bar{\vartheta}_n) = K_2 \left(\frac{Z}{B_{\mathrm{d}} \log_2\left(1+\frac{P_{\mathrm{d}} h_{n,s}}{B_{\mathrm{d}} N_0}\right)} + l^{\mathrm{U}*}_{n,s}(\bar{\vartheta}_n) + K_1 \frac{\omega_n D_n}{\bar{\vartheta}_n} \right). \tag{19}$$

Note that a fixed ϑ_n was considered for the aforementioned analysis. Following that, a binary search can be applied for calculating the optimal ϑ^*_n and subsequently obtaining $l^*_{n,s}$ from (19). The convexity of the problem in (13), guarantees the convergence of the proposed algorithm towards the optimal solutions. This concludes the solution of step one of the optimization problem.

2) Global delay minimization: Based on the previously obtained optimal delay values $l^*_{n,s}$ for each user-edge pair, the HFL total delay minimization problem in (11) can be rewritten as

$$\min_{\boldsymbol{A}} \max_{n \in \mathcal{N}} \left(\sum_{s=1}^{S} a_{n,s} l^*_{n,s} \right) \tag{20}$$

$$\textbf{\textit{s.t.}} \sum_{s=1}^{S} a_{n,s} \leq 1, \quad \forall n \in \mathcal{N}, \tag{20a}$$

$$\sum_{n=1}^{N} a_{n,s} = M, \quad \forall s \in \mathcal{S}, \tag{20b}$$

$$a_{n,s} \in \{0,1\}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, \tag{20c}$$

where the optimal user-assignment matrix $\boldsymbol{A}$ has to be obtained. The problem in (20) is an integer-linear programming problem and can be solved with standard methods. However, it is easy to verify that the problem is equivalent to the *Linear Bottleneck Assignment Problem (LBAP)* [17], thus it can be solved efficiently and optimally through the utilization of bipartite graphs.

IV. PERFORMANCE EVALUATION AND DISCUSSION

For the purposes of model evaluation, we consider $N = 50$ users, uniformly distributed within a 4 km^2 rectangular area. The network consists of $S = 4$ ESs, arranged in a square with edge lengths equal to 1km. All network parameters are listed in Table I and maintain their respective values, unless specified otherwise.

The FL task is selected to be the classification of handwritten digits using an IID subset of the widely-used MNIST dataset, while each user is equipped with 50 samples. The classifier is a 3-layer feed-forward neural network with a 62-node hidden layer, utilizing adaptive moment estimation, with a learning rate of 0.001 and a mini-batch size of 10. The loss function f is defined as the cross entropy loss. We measured the accuracy of the FL model at each point in time, for different values of K_1, K_2 as well as for different RB sizes. We first compare the performance of the proposed HFL

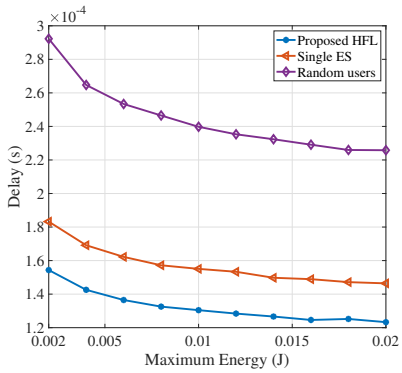


Fig. 2. Impact of the users' maximum energy $e_{\max}$, on the delay of a HFL round.

protocol with various benchmark schemes, in terms of delay reduction, i.e., the delay of a global HFL round. In all figures, the term delay refers to the duration of a single global HFL round. For the considered simulation, the parameters K_1 and K_2 have both been set equal to 1. The results are illustrated in Fig. 2, which have been extracted by means of Monte Carlo simulation. Regarding the benchmark schemes, *Random Users* implies that the users where randomly selected and assigned to the ESs, while *Single ES* refers to the deployment of a single ES. In the later scheme, we have assumed that the number of RBs is 20, in order to conduct a fair comparison by considering an equal amount of the total networks' RBs for all schemes. By observing Fig. 2, it is obvious that the proposed HFL protocol significantly outperforms the random user assignment scheme, verifying the importance of employing an optimal user-edge assignment. Moreover, the results validate the benefits of deploying multiple ES's into the network, in opposition to the *single ES* network configuration.

In Fig. 3, the impact of the RBs' number on the total delay of a FL round was evaluated. It can be observed that, as the number of RBs increases the total delay also increases, for all the considered schemes. This is reasonable, since by increasing the RBs number, i.e., increasing the number of participating users, it is more likely that a user suffer from bad channel conditions, leading to increased delay. Recall that the total delay of a HFL round is being enforced by the slowest scheduled device, in the particular round. Moreover, it is obvious that the proposed HFL scheme outperforms the *Single ES*. The performance gap between them increases along with the number of RBs. An interpretation of this result is that the spatial spreading of the available RBs, through the employment of several ES's, improves the network's coverage

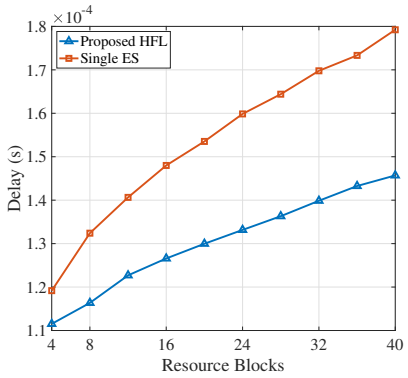
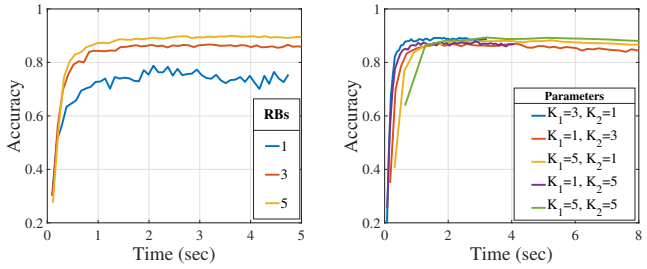


Fig. 3. Impact of the resource blocks number, on the delay of a HFL round.

and its ability to select stronger users for participating into the FL task.

The above results pertain to the wireless network aspect of the proposed scheme. They are thus independent of the specific FL task and the underlying dataset, as well as the inner model structure, which provides flexibilty regarding the design parameters investigated below.

In Fig 4(a), the model accuracy across time is demonstrated for different RB sizes per edge server. It is clear that a bigger RB size, meaning more participating users, leads to both better accuracy and faster convergence. In Fig 4(b), different K_1, K_2 settings are evaluated. It can be observed that the settings $(3, 1)$ and $(5, 5)$ exhibit the highest accuracy. Moreover, the case $(3, 1)$ demonstrates the highest convergence speed, while $(5, 5)$ the lowest. These results indicate that many user-edge communication rounds are not necessarily beneficial. For instance, the case $(5, 5)$ demonstrates high accuracy but slow convergence, owing to the frequent user-edge communication and the increased number of local iterations. It should be noted, though, that the model's behaviour is likely to be task-specific, and the trade-off balance between model accuracy and convergence time may differ for other FL tasks.



(a) Impact of the RBs number on the model's convergence.

(b) Impact of K_1, K_2 on the model's convergence.

Fig. 4. Accuracy over time.

V. Conclusions

In this paper, we have presented a delay-optimized HFL scheme for the NGIoT. More specifically, we have jointly optimized the communication, computation resources, as well as the user-edge assignment, resulting in optimal delay during a HFL round. Simulation results verify the effectiveness of the proposed HFL scheme, highlighting the performance gains of HFL over the single ES federated learning implementation. Finally, insights are provided regarding the effects of key design parameters which characterize the HFL framework.

References

[1] M. Simos, "Wireless hierarchical federated learning: Delay optimization and resource allocation," Diploma thesis, Dep. of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 2021. [Online]. Available: http://ikee.lib.auth.gr/record/332616/files

[2] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015. [Online]. Available: https://science.sciencemag.org/content/349/6245/255

[3] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[4] A. Brékine *et al.*, "Building a Roadmap for the Next Generation Internet of Things. Research, Innovation and Implementation 2021 – 2027 (Scoping Paper)," M. Brynskov, F. M. Facca, and G. Hrasko, Eds., Sep. 2019.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST) Volume 10 Issue 2, Article No. 12, January 2019*, Feb. 2019.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[7] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, pp. 1205–1221, 2019.

[8] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. on Communications (ICC)*, 2020.

[9] Z. Zhao, C. Feng, H. H. Yang, and X. Luo, "Federated-learning-enabled intelligent fog radio access networks: Fundamental theory, key techniques, and future trends," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 22–28, 2020.

[10] 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 Tran. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2020.

[11] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1387–1395.

[12] N. Mhaisen, A. Awad, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Trans. on Netw. Sci. Eng.*, pp. 1–1, 2021.

[13] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8866–8870.

[14] 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.*, vol. 19, no. 10, pp. 6535–6548, 2020.

[15] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (WFL) for 6G networks - Part I: Research challenges and future trends," *IEEE Commun. Lett.*, pp. 1–1, 2021.

[16] ——, "Wireless federated learning (WFL) for 6G networks - Part II: The compute-then-transmit NOMA paradigm," *IEEE Commun. Lett.*, pp. 1–1, 2021.

[17] H. N. Gabow and R. E. Tarjan, "Algorithms for two bottleneck optimization problems," *J. Algorithms*, vol. 9, no. 3, pp. 411–417, 1988.