Federated Learning over Wireless Networks: Optimization Model Design and Analysis

Nguyen H. Tran, Wei Bao, Albert Zomaya
School of Computer Science
The University of Sydney
Sydney, Australia
{nguyen.tran, wei.bao, albert.zomaya}@sydney.edu.au

Minh N.H. Nguyen, Choong Seon Hong
Department of Computer Science and Engineering
Kyung Hee University
Yongin, South Korea
{minhnhn, cshong}@khu.ac.kr

***Abstract*—There is an increasing interest in a new machine learning technique called Federated Learning, in which the model training is distributed over mobile user equipments (UEs), and each UE contributes to the learning model by independently computing the gradient based on its local training data. Federated Learning has several benefits of data privacy and potentially a large amount of UE participants with modern powerful processors and low-delay mobile-edge networks. While most of the existing work focused on designing learning algorithms with provable convergence time, other issues such as uncertainty of wireless channels and UEs with heterogeneous power constraints and local data size, are under-explored. These issues especially affect to various trade-offs: (i) between computation and communication latencies determined by learning accuracy level, and thus (ii) between the Federated Learning time and UE energy consumption. We fill this gap by formulating a Federated Learning over wireless network as an optimization problem FEDL that captures both trade-offs. Even though FEDL is non-convex, we exploit the problem structure to decompose and transform it to three convex sub-problems. We also obtain the globally optimal solution by charactering the closed-form solutions to all sub-problems, which give qualitative insights to problem design via the obtained optimal FEDL learning time, accuracy level, and UE energy cost. Our theoretical analysis is also illustrated by extensive numerical results.**

***Index Terms*—Distributed Machine Learning over Wireless Networks, Federated Learning, Optimization Decomposition.**

I. Introduction

The rise in interest in maintaining the privacy of consumer data [1] has led to the emergence of a new class of machine learning techniques that exploit the participation of a number of mobile phone users. One such popular technique is called Federated Learning [2]–[4]. This learning technique allows the users to collaboratively build a shared learning model while preserving all training data on their own user equipment (UE). In particular, a UE computes the updates to the current global model on its local training data, which is then aggregated and fed-back by a central server, so that all UEs have access to the same global model in order to compute their new updates. This process is repeated until an accuracy level of the learning model is reached. By this way, the user data privacy is well protected because local training data are not shared, thus it decouples the machine learning from acquiring, storing, and training data in datacenters as conventional approaches.

In the scope of machine learning, there is a dilemma that UEs' local data are valuable for training models, which will greatly increase user experiences; however, these data are also privacy sensitive in nature so that it is risky to log the data to datacenters for model training. For example, location-based services such as the app Waze [5], can help users avoid heavy-traffic roads and thus reduce the congestion. However, in this application, users have to share their own locations to the server. Federated Learning is a great solution to address such dilemma: instead of sending the raw data, only intermediate gradient values are sent. Furthermore, the modern smart UE can be considered as a personal computer with powerful integrated processors (e.g., Hexagon DSP with Qualcomm Hexagon Vector eXtensions on Snapdragon 835 [4]) for heavy computing tasks, and plethora of sensors (e.g., cameras, microphones, and GPS) for collecting a wealth amount of data, which ensures the feasibility of Federated Learning to foster more intelligent applications.

There are many reasons to support Federated Learning. First, with the recent advances of edge computing, Federated Learning can be easily implemented in reality. Equipped with a large amount of computing resources at the edge, a centralized powerful datacenter is no longer a must. Instead, the model training can be completed in a distributed fashion, and the delay of uploading a huge amount of raw data can be reduced. Second, Federated Learning greatly facilitates an unprecedented large-scale flexible data collection and model training. For example, a crowd of smart devices can proactively sense and collect data during the day hours, then they jointly feedback and update the global model during the night hours, to improve the efficiency and accuracy for next-day usage. We envision that such approach will boost a new generation of smart services, such as smart transportation, smart shopping, and smart hospital.

With all the promising benefits, Federated Learning also comes with new challenges to tackle. On one hand, similar to other machine learning schemes, one of the most critical performance metrics of Federated Learning is the learning time it takes to converge to a predefined accuracy level. However, different from conventional machine learning approaches, Federated Learning time includes not only the UE computation time (which depend on UEs' CPU types and local

data sizes), but also the communication time of all UEs (which depends on UE channel gains and update data size). Thus, the first question is: *whether UEs should spend more time on computation to achieve high learning accuracy and less communication updates, or vice versa?* On the other hand, due to the limited battery of the participants, how UE resources such as computing and transmission powers are allocated to minimize the energy consumption is the main concern. Thus, the second question is: *how to strike a balance between two conflicting goals of minimizing Federated Learning time and UE energy consumption?* Furthermore, these two questions cannot be answered separately due to their couplings.

To address these questions, we provide first-of-its-kind "Federated Learning over Wireless Networks" problem design and analysis, which can be summarized as follows:

- We pose the Federated Learning over wireless networks problem (FEDL) that capture two trade-offs: (i) learning time versus UE energy consumption by using Pareto efficiency model, and (ii) computation versus communication learning time by finding the optimal learning accuracy parameter (Section III.)
- Despite non-convex nature of FEDL, we exploit its special structure and use the variable decomposition approach to split and transform FEDL into three convex sub-problems. We show that the first two sub-problems can be solved separately, then their solutions are used to obtain the solution to the third sub-problem. By analysing the closed-form solution to each sub-problem, we obtain qualitative insights into the impact of the Pareto-efficient controlling knob to the optimal: (i) computation and communication learning time, (ii) UE resource allocation, and (iii) learning accuracy. Finally, the combined solution to all sub-problems can provide the globally optimal solution to FEDL (Section IV.)
- We further provide extensive numerical results to examine the: (i) impact of UE heterogeneity, (ii) Pareto curve between UE energy cost and system learning time, and (iii) the impact of the proportion of computation over communication time on the optimal accuracy level (Section V). Finally, we present the conclusions and some possible future directions of Federated Learning over wireless networks (Section VI.)

II. Related Works

Due to Big Data applications and complex models such as Deep Learning, training machine learning models needs to be distributed over multiple machines, giving rise to the researches on decentralized machine learning [6]–[8]. However, most of the algorithms in these works are designed for machines having balanced and i.i.d. data and being connected to high-throughput networks such as datacenters.

From a different motivation, Federated Learning (and related on-device intelligence approaches), which has attracted many attentions recently [2], [3], [9], exploits the collaboration of mobile devices that can be large in number, slow and/or unstable in Internet connections, and have non-i.i.d. and unbalanced data locally. However, most of these works focused on designing algorithms to improve the convergence of learning time, unconcerned about other limiting factors such as *wireless communication and energy-limited* nature of mobile UEs that can affect the performance of Federated Learning. To address this gap, we study how the computation and communication characteristics of UEs can affect to their energy consumption, learning time convergence, and accuracy level of Federated Learning, considering *heterogeneous* UEs in terms of data size, channel gain, computing and transmission power capabilities.

III. System Model

We consider a wireless multi-user system consisting of one base station (BS) and a set $\mathcal{N}$ of N UEs. Each participating UE n stores a local data set $\mathcal{D}_n$, with its size is denoted by D_n. Then, we can define the total data size by $D = \sum_{n=1}^{N} D_n$. In an example of the supervised learning setting, at UE n, $\mathcal{D}_n$ defines the collection of data samples given as a set of input-output pairs $\{x_i, y_i\}_{i=1}^{D_n}$, where $x_i \in \mathbb{R}^d$ is an input sample vector with d features, and $y_i \in \mathbb{R}$ is the labeled output value for the sample x_i. The data can be generated through the usage of UE, for example, via interactions with mobile apps. With these UEs data, several machine learning applications can be employed for wireless networks such as predicting the BS's load in next hours for dynamic BS load balancing, or predicting the next hovering position of drones so that their coverage is optimized.

In a typical learning problem, for a sample data $\{x_i, y_i\}$ with input x_i (e.g., the response time of various apps inside the UE), the task is to find the *model parameter* w that characterizes the output y_i (e.g., label of BS load, such as high or low, in next hours) with the loss function $f_i(w)$. Some examples of the loss function are $f_i(w) = \frac{1}{2}(x_i^T w - y_i)^2, y_i \in \mathbb{R}$ for linear regression and $f_i(w) = \{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$ for support vector machine. The loss function on the data set of UE n is defined as

$$J_n(w) := \frac{1}{D_n} \sum\nolimits_{i \in \mathcal{D}_n} f_i(w). \tag{1}$$

Then, the learning model is the minimizer of the following global loss function minimization problem

$$\min_{w \in \mathbb{R}^d} J(w) := \sum\nolimits_{n=1}^{N} \frac{D_n}{D} J_n(w). \tag{2}$$

A. Federated Learning over Wireless Networks

In this section, we adapt the Federated Learning framework [9] to the wireless networks as the following algorithm.

FEDL:

1) At the UE side, there are two phases at t^{th} update:
Computation. Each UE n solves its local problem

$$w_n^{(t)} = \arg\min_{w_n \in \mathbb{R}^d} F_n\Big(w_n \,|\, w^{(t-1)}, \nabla J^{(t-1)}\Big) \tag{3}$$

with a local accuracy[1] $0 \leq \theta \leq 1$ (i.e., $||\nabla F_n(w^{(t)}|| \leq \theta\, ||\nabla F_n(w^{(t-1)}||)$).

Communication. All UEs share the wireless environment to transmit $w_n^{(t)}$ and the gradient $\nabla J_n^{(t)}$ to BS.

2) At the BS, the following information is aggregated

$$w^{(t+1)} = \frac{1}{N} \sum\nolimits_{n=1}^{N} w_n^{(t)} \tag{4}$$

$$\nabla J^{(t+1)} = \frac{1}{N} \sum\nolimits_{n=1}^{N} \nabla J_n^{(t)} \tag{5}$$

and fed-back to all UEs. This process is iterative until a global accuracy $0 \leq \varepsilon \leq 1$ is achieved (i.e., $||\nabla J(w^{(t)}|| \leq \varepsilon ||\nabla J(w^{(t-1)}||)$).

We briefly summarize the above algorithm in what follows. To solve problem (2), FEDL uses an iterative approach that requires a number of *global iterations* (i.e., communication rounds) to achieve an global accuracy level ε. In each global iteration, there are interactions between the UEs and BS. Specifically, a participating UE, in each computation phase, will minimize its objective $F_n(w_n)$ in (3) using local training data $\mathcal{D}_n$. Minimizing F_n also takes multiple *local iterations* up to an accuracy threshold θ that is common to all UEs. As in [2], [6], the computation phase is *synchronous* such that all UEs have to finish solving their local problems before entering the communication phase to transmit their updates to the BS by using a wireless medium sharing scheme (e.g., time-sharing similar to TDMA).

The BS then aggregates the local model parameters and gradients, i.e., w_n and the ∇J_n, $\forall n$, respectively, to update and then broadcast the global model parameters and gradients, i.e., w and the ∇J according to (4) and (5), respectively, which are required for participating UEs to minimize their $F_n(w_n)$[2], in the next global iteration. We see that the BS does not access the local data $\mathcal{D}_n$, $\forall n$, thus *preserving data privacy*.

For strongly convex objective $J(w)$, the general upper bound on global iterations is shown to be [6]

$$K(\varepsilon, \theta) = \frac{\mathcal{O}(\log(1/\epsilon))}{1-\theta}, \tag{6}$$

which is affected by both global accuracy ε and local accuracy θ. For example, when ε and θ are small (more accurate), FEDL needs to runs more global iterations. On the other hand, each global iteration consists of both computation and uplink communication time. Since the downlink bandwidth is larger than that of uplink and the BS power is much higher than UE's transmission power, the downlink time is negligible compared to uplink time and thus is not considered in this work. The computation time, however, depends on the number of local iterations, which is upper bounded by $\mathcal{O}(\log(1/\theta))$ for a wide range of iterative algorithms to solve (3) such as gradient descent, coordinate descent, or stochastic dual coordinate descent [10]. In [6], it is shown that FEDL performance does not depend on which algorithms is used in computation phase as long as the convergence time of that algorithm is upper-bounded by $\mathcal{O}(\log(1/\theta))$. Denote the time of one local iteration by T_{cmp}, then the computation time in one global iteration is $v \log(1/\theta) T_{cmp}$ for some positive constant v that depends on the data size and condition number of the local problem [10]. Denoting the communication time in one global iteration by T_{com}, the total time of one global iteration of FEDL is defined as

$$T_{glob}(T_{cmp}, T_{com}, \theta) := T_{com} + v \log(1/\theta) T_{cmp}. \tag{7}$$

In this work, we consider a fixed global accuracy ϵ, so we normalize $\mathcal{O}(\log(1/\epsilon))$ to 1 so that $K(\theta) = \frac{1}{1-\theta}$ for ease of presentation. Furthermore, we also normalize v to 1 since we can absorb v into T_{cmp} as the upper bound of one local computation iteration. Thus the upper-bound of FEDL learning time is $K(\theta)\, T_{glob}(\theta)$. Henceforth, we omit the word "upper-bound" for brevity. In the next sub-sections we will present how computation and communication time relate to UEs' energy consumption.

B. Computation Model

We denote the number of CPU cycles for UE n to execute one sample of data by c_n, which can be measured offline [11] and is known as a priori. Since all samples $\{x_i, y_i\}_{i \in \mathcal{D}_n}$ have the same size (i.e., number of bits), the number of CPU cycles required for UE n to run one local iteration is $c_n D_n$. Denote the CPU-cycle frequency of the UE n by f_n. Then the CPU energy consumption of UE n for one local iteration of computation can be expressed as follows [12]

$$E_n^{cmp}(f_n) = \sum\nolimits_{i=1}^{c_n D_n} \frac{\alpha_n}{2} f_n^2 = \frac{\alpha_n}{2} c_n D_n f_n^2, \tag{8}$$

where $\alpha_n/2$ is the effective capacitance coefficient of UE n's computing chipset. Furthermore, the computation time per local iteration of the UE n is $\frac{c_n D_n}{f_n}$, $\forall n$. We denote the vector of f_n by $f \in \mathbb{R}^n$.

C. Communication Model

In FEDL, regarding to the communication phase of UEs, we consider a time-sharing multi-access protocol for UEs. We note that this time-sharing model is not restrictive because other schemes, such as OFDMA, can also be applied to FEDL. The achievable transmission rate (nats/s) of UE n is defined as follows:

$$r_n = B \ln\big(1 + \frac{h_n p_n}{N_0}\big), \tag{9}$$

where B is the bandwidth, N_0 is the background noise, p_n is the transmission power, and h_n is the channel gain of the UE n. We assume that h_n is constant during the learning time of FEDL[3]. Denote the fraction of communication time allocated

[1]Here $\theta = 0$ means the local problem is required to be solved optimally, and $\theta = 1$ means no progress for local problem [9].

[2]One example, from [9], is $F_n(w_n) = J_n(w_n) - \big(\nabla J_n^{(t-1)} - \beta_1 \nabla J^{(t-1)}\big)^T w_n + \frac{\beta_2}{2} ||w_n - w_n^{(t-1)}||^2$ where $\beta_1, \beta_2 \geq 0$ are parameters.

[3]We treat the case of random h_n by adding the outage probability constraint, e.g., for Rayleigh fading channel, $\Pr\big(\frac{h_n p_n}{N_0} < \gamma\big) \leq \zeta$ where γ is the SNR threshold and ζ is the bounded probability [13]. This constraint is equivalent to $p_n \geq \frac{-\gamma N_0}{\log(1-\zeta)}$ and can be integrated to the constraint (17) without changing any insights of the considered problem.

to UE n by τ_n, and the data size (in nats) of both w_n and ∇J_n by s_n. Because the dimensions of vectors w_n and ∇J_n are fixed, we assume that their sizes are constant throughout the FEDL learning. Then the transmission rate of each UE n is

$$r_n = s_n/\tau_n, \tag{10}$$

which is shown to be the most energy-efficient transmission policy [14]. Thus, to transmit s_n within a time duration τ_n, the UE n's energy consumption is

$$E_n^{com}(\tau_n) = \tau_n p_n = \tau_n \, p_n(s_n/\tau_n), \tag{11}$$

where the power function is

$$p_n(s_n/\tau_n) := \frac{N_0}{h_n}\Big(e^{\frac{s_n/\tau_n}{B}} - 1\Big) \tag{12}$$

according to (9) and (10). We denote the vector of τ_n by $\tau \in \mathbb{R}^n$.

D. Problem formulation

Define the total energy consumption of all UEs for each global iteration by E_{glob}, which is expressed as follows:

$$E_{glob}(f, \tau, \theta) := \sum\nolimits_{n=1}^{N} E_n^{com}(\tau_n) + \log(1/\theta) E_n^{cmp}(f_n).$$

Then, we consider an optimization problem, abusing the same name FEDL, as follows

$$\underset{f,\tau,\theta,T_{com},T_{cmp}}{\text{minimize}} \quad K(\theta)\big[E_{glob}(f, \tau, \theta) + \kappa \, T_{glob}(T_{cmp}, T_{com}, \theta)\big] \tag{13}$$

$$\text{subject to} \quad \sum\nolimits_{n=1}^{N} \tau_n \le T_{com}, \tag{14}$$

$$\max_n \frac{c_n D_n}{f_n} = T_{cmp}, \tag{15}$$

$$f_n^{min} \le f_n \le f_n^{max}, \ \forall n \in \mathcal{N}, \tag{16}$$

$$p_n^{min} \le p_n(s_n/\tau_n) \le p_n^{max}, \ \forall n \in \mathcal{N}, \tag{17}$$

$$0 \le \theta \le 1. \tag{18}$$

Minimize both UEs' energy consumption and the Federated Learning time are conflicting. For example, the UEs can save the energy by setting the lowest frequency level all the time, but this will certainly increase the learning time. Therefore, to strike the balance between energy cost and learning time, the weight κ (Joules/second), used in the objective as an amount of additional energy cost that FEDL is willing to bear for one unit of learning time to be reduced, captures the Pareto-optimal tradeoff between the UEs' energy cost and the Federated Learning time. For example, when most of the UEs are plugged in, then UE energy cost is not the main concern, thus κ can be large.

While constraint (14) captures the time-sharing uplink transmission of UEs, constraint (15) defines that the computing time in one local iteration is determined by the "bottleneck" UE (e.g., with large data size and low CPU frequency). The feasible regions of CPU-frequency and transmit power of UEs are imposed by constraints (16) and (17), respectively. We note that (16) and (17) also capture the heterogeneity of UEs with different types of CPU and transmit chipsets. The last constraint restricts the feasible range of the local accuracy.

IV. Solutions to FEDL

We see that FEDL is non-convex due to the constraint (15) and several products of two functions in the objective function. However, in this section we will characterize FEDL's optimal solution by decomposing it into multiple convex sub-problems.

We consider the first case when θ is fixed, then FEDL can be decomposed into two sub-problems as follows:

$$\textsf{SUB1:} \quad \underset{f,T_{cmp}}{\text{minimize}} \quad \sum\nolimits_{n=1}^{N} E_n^{cmp}(f_n) + \kappa T_{cmp} \tag{19}$$

$$\text{subject to} \quad \frac{c_n D_n}{f_n} \le T_{cmp}, \ \forall n \in \mathcal{N}, \tag{20}$$

$$f_n^{min} \le f_n \le f_n^{max}, \ \forall n \in \mathcal{N}. \tag{21}$$

$$\textsf{SUB2:} \quad \underset{\tau,T_{com}}{\text{min.}} \quad \sum\nolimits_{n=1}^{N} E_n^{com}(\tau_n) + \kappa T_{com} \tag{22}$$

$$\text{s.t.} \quad \sum\nolimits_{n=1}^{N} \tau_n \le T_{com}, \tag{23}$$

$$p_n^{min} \le p_n(s_n/\tau_n) \le p_n^{max}, \forall n. \tag{24}$$

While SUB1 is a CPU-cycle control problem for the computation time and energy minimization, SUB2 can be considered as an uplink power control to determine the UEs' fraction of time sharing to minimize the UEs energy and communication time. We note that the constraint (15) of FEDL is replaced by an equivalent one (20) in SUB1. We can consider T_{cmp} and T_{com} as virtual deadlines for UEs to perform their computation and communication updates, respectively.

It can be observed that both SUB1 and SUB2 are convex problems. We note that the constant factors $K(\theta)\log(1/\theta)$ and $K(\theta)$ of SUB1 and SUB2's objectives, respectively, are omitted since they have no effects to these sub-problems' solutions.

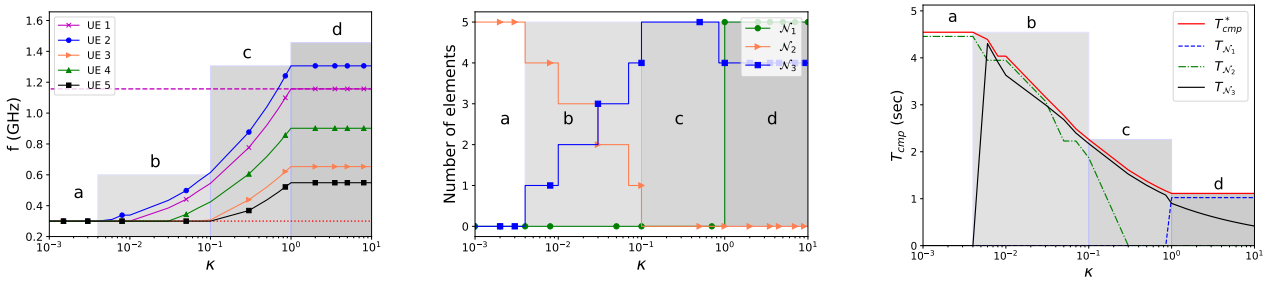
A. SUB1 Solution

We first propose Algorithm 1 in order to categorize UEs into one of three groups: $\mathcal{N}_1$ is a group of "bottleneck" UEs that always run its maximum frequency, $\mathcal{N}_2$ is the group of "strong" UEs which can finish their tasks before the computational virtual deadline even with the minimum frequency, and $\mathcal{N}_3$ is the group of UEs having the optimal frequency inside the interior of their feasible sets.

Lemma 1. *The optimal solution to SUB1 is as follows*

$$f_n^* = \begin{cases} f_n^{max}, & \forall n \in \mathcal{N}_1, \\ f_n^{min}, & \forall n \in \mathcal{N}_2, \\ \frac{c_n D_n}{T_{cmp}^*}, & \forall n \in \mathcal{N}_3, \end{cases} \tag{25}$$

$$T_{cmp}^* = \max\big\{T_{\mathcal{N}_1}, T_{\mathcal{N}_2}, T_{\mathcal{N}_3}\big\}, \tag{26}$$



(a) Optimal CPU frequency of each UE. (b) Three subsets outputted by Alg. 1. (c) Optimal computation time.

Fig. 1: Solution to SUB1 with five UEs. For wireless communication model, the UE channel gains follow the exponential distribution with the mean $g_0(d_0/d)^4$ where $g_0 = -40$ dB and the reference distance $d_0 = 1$ m. The distance between these devices and the wireless access point is uniformly distributed between 2 and 50 m. In addition, $B = 1$ MHz, $\sigma = 10^{-10}$ W, the transmission power of devices are limited from 0.2 to 1 W. For UE computation model, we set the training size D_n of each UE as uniform distribution in $5-10$ MB, c_n is uniformly distributed in $10-30$ cycles/bit, f_n^{max} is uniformly distributed in $1.0-2.0$ GHz, $f_n^{min} = 0.3$ GHz. Furthermore, $\alpha = 2\times10^{-28}$ and the UE update size $s_n = 25,000$ nats (≈4.5 KB).

Algorithm 1 Finding $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ in Lemma 1

Sort UEs such that $\frac{c_1 D_1}{f_1^{min}} \le \frac{c_2 D_2}{f_2^{min}} \ldots \le \frac{c_N D_N}{f_N^{min}}$
Input: $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \emptyset$, $\mathcal{N}_3 = \mathcal{N}$, $T_{\mathcal{N}_3}$ in (29)
for $i = 1$ to N **do**
  if $\max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{max}} \ge T_{\mathcal{N}_3} > 0$ and $\mathcal{N}_1 == \emptyset$ **then**
    $\mathcal{N}_1 = \mathcal{N}_1 \cup \left\{m : \frac{c_m D_m}{f_m^{max}} = \max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{max}}\right\}$
    $\mathcal{N}_3 = \mathcal{N}_3 \setminus \mathcal{N}_1$ and update $T_{\mathcal{N}_3}$ in (29)
  end if
  if $\frac{c_i D_i}{f_i^{min}} \le T_{\mathcal{N}_3}$ **then**
    $\mathcal{N}_2 = \mathcal{N}_2 \cup \{i\}$
    $\mathcal{N}_3 = \mathcal{N}_3 \setminus \{i\}$ and update $T_{\mathcal{N}_3}$ in (29)
  end if
end for

where $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3 \subseteq \mathcal{N}$ are three subsets of UEs produced by Algorithm 1 and

$$T_{\mathcal{N}_1} = \max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{max}} \tag{27}$$

$$T_{\mathcal{N}_2} = \max_{n\in\mathcal{N}_2} \frac{c_n D_n}{f_n^{min}} \tag{28}$$

$$T_{\mathcal{N}_3} = \left(\frac{\sum_{n\in\mathcal{N}_3} \alpha_n (c_n D_n)^3}{\kappa}\right)^{1/3}. \tag{29}$$

All proofs in this paper (by analysing the KKT condition) are omitted due to limited space. From Lemma 1, first, we see that the optimal solution depends not only on the existence of these subsets, but also on their virtual deadlines $T_{\mathcal{N}_1}$, $T_{\mathcal{N}_2}$, and $T_{\mathcal{N}_3}$, in which the longest of them will determine the optimal virtual deadline T^*_{cmp}. Second, from (25), the optimal frequency of each UE will depend on both T^*_{cmp} and the subset it belongs to. We note that depending on κ, some of the three sets (not all) are possibly empty sets, and by default $T_{\mathcal{N}_i} = 0$ if $\mathcal{N}_i$ is an empty set, $i = 1, 2, 3$. Next, by varying κ, we observe the following special cases.

Corollary 1. *The optimal solution to SUB1 can be divided into four regions as follows.*

a) $\kappa \le \min_{n\in\mathcal{N}} \alpha_n (f_n^{min})^3$:
$\mathcal{N}_1$ and $\mathcal{N}_3$ are empty sets. Thus, $\mathcal{N}_2 = \mathcal{N}$, $T^*_{com} = T_{\mathcal{N}_2} = \max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{min}}$, and $f_n^* = f_n^{min}, \forall n \in \mathcal{N}$.

b) $\min_{n\in\mathcal{N}} \alpha_n (f_n^{min})^3 < \kappa \le \left(\max_{n\in\mathcal{N}_2} \frac{c_n D_n}{f_n^{min}}\right)^3$:
$\mathcal{N}_2$ and $\mathcal{N}_3$ are non-empty sets, whereas $\mathcal{N}_1$ is empty. Thus, $T^*_{cmp} = \max\{T_{\mathcal{N}_2}, T_{\mathcal{N}_3}\}$, and $f_n^* = \max\left\{\frac{c_n D_n}{T^*_{cmp}}, f_n^{min}\right\}, \forall n \in \mathcal{N}$.

c) $\left(\max_{n\in\mathcal{N}_2} \frac{c_n D_n}{f_n^{min}}\right)^3 < \kappa \le \frac{\sum_{n\in\mathcal{N}_3} \alpha_n \left(c_n D_n\right)^3}{\left(\max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{max}}\right)^3}$:
$\mathcal{N}_1$ and $\mathcal{N}_2$ are empty sets. Thus $\mathcal{N}_3 = \mathcal{N}$, $T^*_{cmp} = T_{\mathcal{N}_3}$, and $f_n^* = \frac{c_n D_n}{T_{\mathcal{N}_3}}, \forall n \in \mathcal{N}$.

d) $\kappa > \frac{\sum_{n\in\mathcal{N}_3} \alpha_n \left(c_n D_n\right)^3}{\left(\max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{max}}\right)^3}$:
$\mathcal{N}_1$ is non-empty. Thus $T^*_{cmp} = T_{\mathcal{N}_1}$, and

$$f_n^* = \begin{cases} f_n^{max}, & \forall n \in \mathcal{N}_1 \\ \max\left\{\frac{c_n D_n}{T_{\mathcal{N}_1}}, f_n^{min}\right\}, & \forall n \in \mathcal{N} \setminus \mathcal{N}_1 \end{cases} \tag{30}$$

We illustrate Corollary 1 in Fig. 1 with four regions[4] as follows.

a) Very low κ (i.e., $\kappa \le 0.004$): Designed for solely energy minimization. In this region, all UE runs their CPU at the lowest cycle frequency f_n^{min}, thus T^*_{cmp} is determined by the last UEs that finish their computation with their minimum frequency.

b) Low κ (i.e., $0.004 \le \kappa \le 0.1$): Designed for prioritized energy minimization. This region contains UEs of both $\mathcal{N}_2$ and $\mathcal{N}_3$. T^*_{cmp} is governed by which subset has higher virtual computation deadline, which also determines the optimal CPU-cycle frequency of $\mathcal{N}_3$. Other UEs with light-loaded data, if exist, can run at the most energy-saving mode f_n^{min} yet still finish their task before T^*_{cmp} (i.e., $\mathcal{N}_2$).

[4]All closed-form solutions are also verified by the solver IPOPT [15].

c) Medium κ (i.e., $0.1 \leq \kappa \leq 1$): Designed for balancing computation time and energy minimization. All UEs belong to $\mathcal{N}_3$ with their optimal CPU-cycle frequency strictly inside the feasible set.

d) High κ (i.e., $\kappa \geq 1$): Designed for prioritized computation time minimization. High value κ can ensure the existence of $\mathcal{N}_1$, consisting the most "bottleneck" UEs (i.e., heavy-loaded data and/or low f_n^{max}) that runs their maximum CPU-cycle in (30) (top) and thus determines the optimal computation time T_{cmp}^*. The other "non-bottleneck" UEs either (i) adjust a "right" CPU-cycle to save the energy yet still maintain their computing time the same as T_{cmp}^* (i.e., $\mathcal{N}_3$), or (ii) can finish the computation with minimum frequency before the "bottleneck" UEs (i.e., $\mathcal{N}_2$) as in (30) (bottom).

B. SUB2 Solution

Before characterizing the solution to SUB2, from (12) and (24), we first define two bounded values for τ_n as follows

$$\tau_n^{max} = \frac{s_n}{B \ln(h_n N_0^{-1} p_n^{min} + 1)}, \tag{31}$$

$$\tau_n^{min} = \frac{s_n}{B \ln(h_n N_0^{-1} p_n^{max} + 1)}, \tag{32}$$

which are the maximum and minimum possible fractions of T_{com} that UE n can achieve by transmitting with its minimum and maximum power, respectively. We also define a new function $g_n : \mathbb{R} \to \mathbb{R}$ as

$$g_n(\kappa) = \frac{s_n/B}{1 + W\left(\frac{\kappa N_0^{-1} h_n - 1}{e}\right)}, \tag{33}$$

where $W(.)$ is the Lambert W-function. We can consider $g_n(.)$ as an indirect "power control" function that helps UE n control the amount of time it should transmit an amount of data s_n by adjusting the power based on the weight κ. This function is strictly decreasing (thus its inverse function $g_n^{-1}(\cdot)$ exists) reflecting that when we put more priotity on minimizing the communication time (i.e., high κ), UE n should raise the power to finish its transmission with less time (i.e., low τ_n).

Lemma 2. *The solution to SUB2 is as follows*

a) If $\kappa \leq g_n^{-1}(\tau_n^{max})$, *then*

$$\tau_n^* = \tau_n^{max} \tag{34}$$

b) If $g_n^{-1}(\tau_n^{max}) < \kappa < g_n^{-1}(\tau_n^{min})$, *then*

$$\tau_n^{min} < \tau_n^* = g_n(\kappa) < \tau_n^{max} \tag{35}$$

c) If $\kappa \geq g_n^{-1}(\tau_n^{min})$, *then*

$$\tau_n^* = \tau_n^{min}, \tag{36}$$

and $T_{com}^* = \sum_{n=1}^{N} \tau_n^*$.

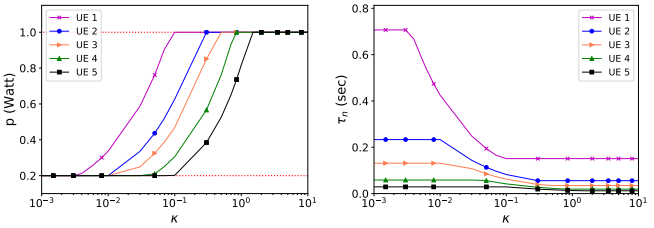
This lemma can be explained through the lens of network economics. If we interpret the FEDL system as the buyer and UEs as sellers with the UE powers as commodities, then the inverse function $g_n^{-1}(\cdot)$ is interpreted as the price of energy that UE n is willing to accept to provide power service for FEDL to reduce the learning time. There are two properties of this function: (i) the price increases with repect to UE power, and (ii) the price sensitivity depends on UEs characteristics, e.g., UEs with better channel quality can have lower price, whereas UEs with larger data size s_n will have higher price. Thus, each UE n will compare its energy price $g_n^{-1}(\cdot)$ with the "offer" price κ by the system to decide how much power it is willing to "sell". Then, there are three cases corresponding to the solutions to SUB2.

a) Low offer: If the offer price κ is lower than the minimum price request $g_n^{-1}(\tau_n^{max})$, UE n will sell its lowest service by transmitting with the minimum power p_n^{min}.

b) Medium offer: If the offer price κ is within the range of an acceptable price range, UE n will find a power level such that the corresponding energy price will match the offer price.

c) High offer: If the offer price κ is higher than the maximum price request $g_n^{-1}(\tau_n^{min})$, UE n will sell its highest service by transmitting with the maximum power p_n^{max}.

Lemma 2 is further illustrated in Fig. 2, showing how the solution to SUB2 varies with respect to κ. It is observed from this figure that due to the UE heterogeneity of channel gain, $\kappa = 0.1$ is a medium offer to UEs 2, 3, and 4, but a high offer to UE 1, and low offer to UE 5.



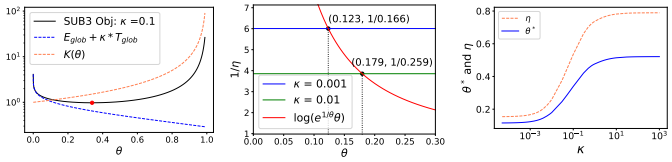
(a) UEs' optimal transmission power. (b) UEs' optimal transmission time .

Fig. 2: The solution to SUB2 with five UEs. The numerical setting is the same as that of Fig. 1.

While SUB1 and SUB2 solutions share the same threshold-based dependence, we observe their differences as follows. In SUB1 solution, the optimal CPU-cycle frequency of UE n depends on the optimal T_{cmp}^*, which in turn depends on the loads (i.e., $\frac{c_n D_n}{f_n}$, $\forall n$) of all UEs. Thus all UE load information is required for the computation phase. On the other hand, in SUB2 solution, each UE n can independently choose its optimal power by comparing its price function $g_n^{-1}(\cdot)$ with κ so that collecting UE information is not needed. The reason is that the synchronization of computation time in constraint (20) of SUB1 requires all UE loads, whereas the UEs' time-sharing constraint (23) of SUB2 can be decoupled by comparing with the fixed "offer" price κ.

C. SUB3 Solution

We observe that the solutions to SUB1 and SUB2 have no dependence on θ so that the optimal T_{com}^*, T_{cmp}^*, f^*, and τ^*



(a) (b) (c)

Fig. 3: The solution to SUB3 with five UEs: : a) Convexity of SUB3, b) Unique θ^* for each η, and c) Impact of κ on η and θ^*. The numerical setting is the same as that of Fig. 1.

can be determined based on κ according to Lemmas 1 and 2. However, these solutions will affect to the third sub-problem of FEDL, as will be shown in what follows.

SUB3:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & K(\theta)\left[E_{glob}(f^*,\tau^*,\theta) + \kappa\, T_{glob}(T^*_{cmp}, T^*_{com}, \theta)\right] \\ \text{subject to} \quad & 0 \le \theta \le 1. \end{aligned} \tag{37}$$

Lemma 3. *There exists a unique solution θ^* of the convex problem SUB3 satisfying the following equation:*

$$\frac{1}{\eta} = \log\left(e^{1/\theta^*}\theta^*\right) \tag{38}$$

where

$$\eta = \frac{\sum_{n=1}^{N} E_n^{cmp}(f_n^*) + \kappa T^*_{cmp}}{\sum_{n=1}^{N}\left[E_n^{cmp}(f_n^*) + E_n^{com}(\tau_n^*)\right] + \kappa\left[T^*_{cmp} + T^*_{com}\right]}. \tag{39}$$

The convexity and unique solution to SUB3 are illustrated in Figs. 3a and 3b, respectively. From its definition, η is the fraction of total computation cost (including all UE energy and computing time cost) of the computation phase over the aggregated communication and computation costs. We then have some observations from Lemma 3. First, according to (38), it can be shown that $0 < \theta^* < \eta < 1$, which is also illustrated in Fig. 3b (with two different values of κ) and Fig. 3c. Thus, small computation cost (compared to communication cost) implies small θ^*, which means UEs need to run a large number of local iterations in computation phase to reduce the number of global iterations $K(\theta^*)$ due to more expensive communication cost. Second, the impact of κ on η and θ^* is illustrated in Fig. 3c as follows.

a) When κ is small enough such that the energy cost dominates the time cost, e.g., $\kappa \le 10^{-3}$, then $\eta \approx \frac{\sum_{n=1}^{N} E_n^{cmp}(f_n^*)}{\sum_{n=1}^{N}\left[E_n^{cmp}(f_n^*)+E_n^{com}(\tau_n^*)\right]}$ (which approaches to a constant when κ falls into case a) of Corollary 1 and Lemma 2). Small values of η in this case (i.e, $\eta \le 0.17$) indicate that computation energy is much smaller than communication energy; thus UEs are better to perform more local computations and less communications, explaining the corresponding small value of θ^* (i.e., $\theta^* \le 0.12$).

b) When κ is in a range such that the energy cost is comparable to the time cost, e.g., $10^{-3} \le \kappa \le 10$, increasing κ makes increase η (because $\eta < 1$), indicating that the computation cost is increasingly more expensive than communication cost. Thus θ^* also increases, reflecting FEDL preference on more communication and less computation.

c) When κ is large enough such that the energy cost dominates the time cost (i.e., $\kappa \ge 10$), then $\eta \approx \frac{T^*_{cmp}}{T^*_{cmp}+T^*_{com}}$ (which approaches to a constant when κ falls into case d) of Corollary 1 and case c) of Lemma 2). Large values of η in this case (i.e, $\eta \ge 0.79$) indicate that computation time is much larger than communication time; thus UEs are better to perform less local computations and more communications, explaining the corresponding large value of θ^* (i.e., $\theta^* \ge 0.53$).

D. FEDL Solution

Theorem 1. *The globally optimal solution to FEDL is the combined solutions to three sub-problems SUB1, SUB2, and SUB3.*

The proof of this theorem is straightforward. The idea is to use the KKT condition to find the stationary points of FEDL. Then we can decompose the KKT condition equations into three groups, each of them matches exactly to the KKT condition of SUB1, SUB2, and SUB3, which can be solved for unique closed-form solution as in Lemmas 1, 2, and 3, respectively. Thus this unique stationary point is also the globally optimal solution to FEDL.

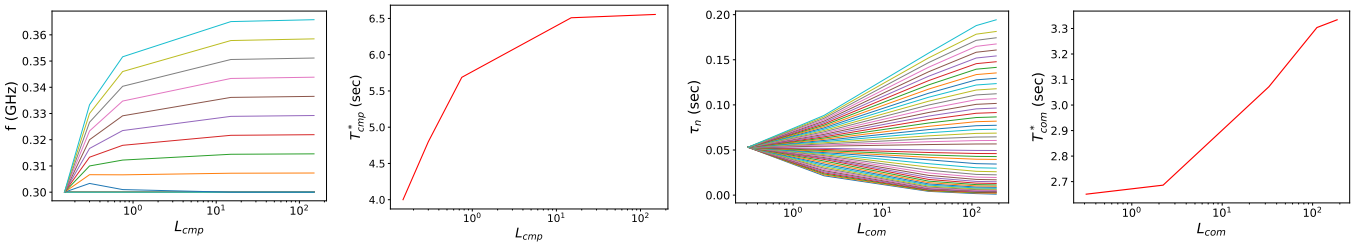
We then have some discussions on the combined solution to FEDL. First, we see that SUB1 and SUB2 solutions can be characterized independently, which can be explained that each UE often has two separate processors: one CPU for mobile applications and another baseband processor for radio control function. Second, neither SUB1 nor SUB2 depends on θ because the communication phase in SUB2 is clearly not affected by the local accuracy of the computing problem, whereas SUB2 considers the computation cost in one local iteration. However, the solutions to SUB1 and SUB2, which can reveal how much communication cost is more expensive than computation cost, are decisive factors to determine the optimal level of local accuracy. Therefore, we can sequentially solve SUB1 and SUB2 first, then SUB3 to achieve the optimal solutions to FEDL.

V. Numerical Results

In this section, both the communication and computation models follow the same setting as in Fig. s 1, except the number of UEs is increased to 50, and all UEs have the same $f_n^{max} = 2.0$ GHz, $c_n = 20$ cycles/bit. Furthermore, we define two new parameters, addressing the UE heterogeneity regarding to computation and communication phases in FEDL, respectively, as follows

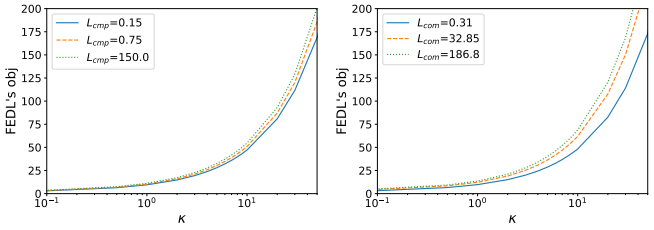
$$L_{cmp} = \frac{\max_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{max}}}{\min_{n\in\mathcal{N}} \frac{c_n D_n}{f_n^{min}}} \tag{40}$$

$$L_{com} = \frac{\max_{n\in\mathcal{N}} \tau_n^{min}}{\min_{n\in\mathcal{N}} \tau_n^{max}}. \tag{41}$$



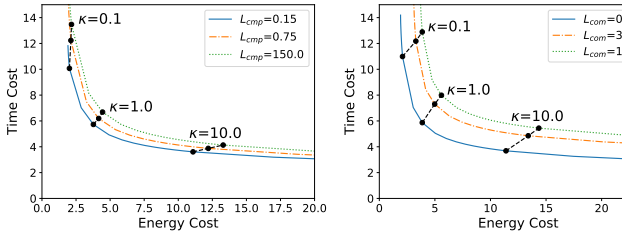
(a) Impact of L_{cmp} on f_n^* (b) Impact of L_{cmp} on T_{cmp}^* (c) Impact of L_{com} on τ_n^* (d) Impact of L_{com} on T_{com}^*

Fig. 4: Impact of UE heterogeneity on SUB1 and SUB2 with $\kappa = 0.07$.



(a) Impact of L_{cmp} (b) Impact of L_{com}

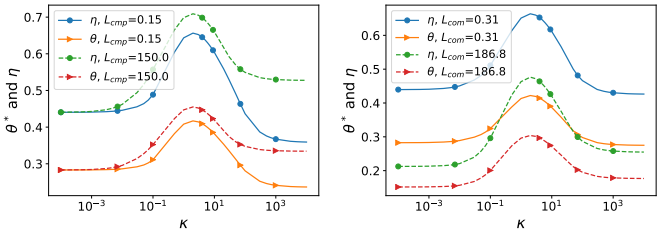
Fig. 5: Impact of UE heterogeneity on FEDL.



(a) Impact of L_{cmp} and κ. (b) Impact of L_{com} and κ.

Fig. 6: Pareto-optimal points of FEDL.

We see that higher values of L_{cmp} and L_{com} indicate higher levels of UE heterogeneity. For example, $L_{cmp} = 1$ ($L_{com} = 1$) can be considered as high heterogeneity level due to unbalanced data distributed and/or UE configuration (unbalanced channel gain distribution) such that UE with their minimum frequency (maximum transmission power) still have the same computation (communication) time as those with maximum frequency (minimum transmission power). The level of heterogeneity is controlled by two different settings. To vary L_{cmp}, the training size D_n is generated with the fraction $\frac{D^{min}}{D^{max}} \in \{1., 0.2, 0.001\}$ but the average data of all UEs is kept at the same value 7.5 MB for varying values of L_{cmp}. On the other hand, to vary L_{com}, the distance between these devices and the BS is generated such that $\frac{d^{min}}{d^{max}} \in \{1., 0.2, 0.001\}$ but the average distance of all UEs is maintained at 26 m for different values of L_{com}. Here D^{min} and D^{max} (d^{min} and d^{max}) are minimum and maximum data size (BS-to-UE distance), respectively. In all scenarios, we fix $L_{cmp} = 0.3$ when varying L_{com} and fix $L_{com} = 0.48$ when varying L_{cmp}.



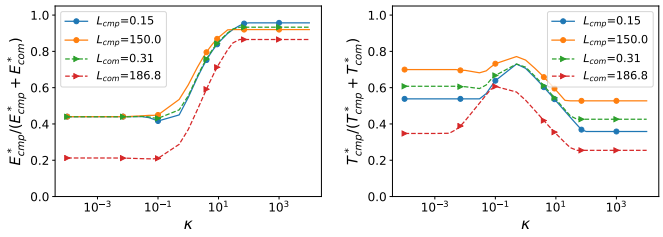
(a) (b)

Fig. 7: Impact of κ on η and θ^*.

1) Impact of UE heterogeneity: We first examine the impact of UE heterogeneity on SUB1 and SUB2 in Fig. 4, which shows that increasing L_{cmp} and L_{com} enforces the optimal f_n^* and τ_n^* having more diverse values, and thus makes increase the computation and communication time T_{cmp}^* and T_{com}^*, respectively. As expected, we observe that the high level of UE heterogeneity has negative impact on the FEDL system, as illustrated in Figs. 6a and 6b, such that the total cost (objective of FEDL) is increased with higher value of L_{cmp} and L_{com} respectively. However, in this setting, when T_{cmp} is comparable to T_{com}, e.g., 6.2 versus 2.9 seconds at $L_{cmp} = L_{com} = 10$, the impact of L_{com} on the total cost is more profound than that of L_{cmp}, e.g., at $\kappa = (0.1, 1, 10)$, the total cost of FEDL increases (1.09, 1.11, 1.10) times and (1.62, 1.40, 1.43) times, when L_{cmp} and L_{com} are increased from 0.15 to 150, and from 0.31 to 186.8, respectively.

On the other hand, with a different setting such that T_{cmp} dominates T_{com}, e.g., 80 versus 7.8 seconds at $L_{cmp} = L_{com} = 10$, the impacts of L_{cmp} and L_{com} on total cost are comparable, e.g., at $\kappa = (0.1, 1, 10)$, the total cost of FEDL increases (1.14, 1.72, 1.65) times and (1.36, 1.21, 1.23) times, when L_{cmp} and L_{com} are increased from 0.05 to 50, and from 0.17 to 181.42, respectively.

2) Pareto Optimal trade-off: We next illustrate the Pareto curve in Fig. 6. This curve shows the trade-off between the conflicting goals of minimizing the time cost $K(\theta)T_{glob}$ and energy cost $K(\theta)E_{glob}$, in which we can decrease one type of cost yet with the expense of increasing the other one. This figure also shows that the Pareto curve of FEDL is more efficient when the system has low level of UE heterogeneity (i.e., small L_{cmp} and/or L_{com}).



(a) (b)

Fig. 8: Proportion of computation energy and time.

3) Impact of η: We quantify the impact of η on the optimal θ^* by varying κ in Fig. 7. Similar to the observations after Lemma 3, when κ is very small or very large, the value of η, which drives the corresponding value of θ^*, is determined by the proportions shown in Fig. 8a or 8b, respectively, which also drives the corresponding value of θ^*. However, in this setting with 50 UEs, when κ is very large, η and θ^* decrease to small values, which is in contrast to the scenario with 5 UEs in Fig. 3c (i.e, large η and θ^* with large κ). The main reason of this difference is that communication time scales with increasing number of UEs due to wireless sharing nature, which makes the time portion small when κ is large, as shown in Fig. 8b. The final observation from Figs. 7 and 8 is that the higher L_{cmp} (L_{com}), the higher T^*_{cmp} (T^*_{com}), and thus higher (lower) portion of computation time, which makes higher (lower) values of η and θ^*.

VI. Conclusions and Future Works

We studied a decentralized machine learning scheme, Federated Learning, in which the training model is distributed to participating mobile devices performing training computation over their local data. Despite the benefit of data privacy, it is not clear that how the computation and communication latencies, especially when Federated Learning is employed over wireless networks, impact on the system learning time. We addressed this issue by first providing a model of implementing Federated Learning over wireless networks, in which each mobile device does not only compute individual learning task, but is also scheduled to transmit personal update in a time-sharing fashion. By decomposing the problem into three sub-problems with convex structure, we then characterized how the computation and communication latencies of mobile devices affect to various trade-offs between the UE energy consumption, system learning time, and learning accuracy parameter, and also quantified the impact of UE heterogeneity on the system cost.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2016R1D1A1B01015320). Dr. CS Hong is the corresponding author.

References

[1] W. H. Report, "Consumer Data Privacy in a Networked World: A Framework for Protecting Privacy and Promoting Innovation in the Global Digital Economy," *Journal of Privacy and Confidentiality*, Mar. 2013.

[2] 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*, Apr. 2017, pp. 1273–1282.

[3] "Federated Learning: Collaborative Machine Learning without Centralized Training Data," http://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[4] "We are making on-device AI ubiquitous," https://www.qualcomm.com/news/onq/2017/08/16/we-are-making-device-ai-ubiquitous, Aug. 2017.

[5] "Free Community-based GPS, Maps & Traffic Navigation App — Waze," https://www.waze.com/.

[6] C. Ma, J. Konečný, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, Jul. 2017.

[7] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning," in *IEEE INFOCOM 2018*, Apr. 2018, pp. 63–71.

[8] J. Hamm, A. C. Champion, G. Chen, M. Belkin, and D. Xuan, "Crowd-ML: A Privacy-Preserving Learning Framework for a Crowd of Smart Devices," in *IEEE ICDCS*, Jun. 2015, pp. 11–20.

[9] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated Optimization: Distributed Machine Learning for On-Device Intelligence," *arXiv:1610.02527 [cs]*, Oct. 2016.

[10] J. Konečný, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, Sep. 2017.

[11] A. P. Miettinen and J. K. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," in *USENIX HotCloud'10*, Berkeley, CA, USA, 2010, pp. 4–4.

[12] T. D. Burd and R. W. Brodersen, "Processor Design for Portable Systems," *Journal of VLSI Signal Processing System*, vol. 13, no. 2-3, pp. 203–221, Aug. 1996.

[13] S. Kandukuri and S. Boyd, "Optimal power control in interference-limited fading wireless channels with outage-probability specifications," *IEEE Transactions on Wireless Communications*, vol. 1, no. 1, pp. 46–55, Jan. 2002.

[14] B. Prabhakar, E. U. Biyikoglu, and A. E. Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *IEEE INFOCOM 2001*, vol. 1, 2001, pp. 386–394.

[15] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.