Federated Learning with Packet Losses

Angelo Rodio, Giovanni Neglia, Fabio Busacca, Stefano Mangione, Sergio Palazzo, Francesco Restuccia, Ilenia Tinnirello

Federated Learning with Packet Losses

Angelo Rodio*, Giovanni Neglia*, Fabio Busacca†, Stefano Mangione‡,
Sergio Palazzo†, Francesco Restuccia§, Ilenia Tinnirello‡
*Inria, Université Côte d'Azur, France. Email: {firstname.lastname}@inria.fr;
†University of Catania, Italy; ‡University of Palermo, Italy; §Northeastern University, United States.

***Abstract*—This paper tackles the problem of training Federated Learning (FL) algorithms over real-world wireless networks with packet losses. Lossy communication channels between the orchestrating server and the clients affect the convergence of FL training as well as the quality of the learned model. Although many previous works investigated how to mitigate the adverse effects of packet losses, this paper demonstrates that FL algorithms over asymmetric lossy channels can still learn the optimal model, the same model that would have been trained in a lossless scenario by classic FL algorithms like `FedAvg`. Convergence to the optimum only requires slight changes to `FedAvg`: *i)* while `FedAvg` computes a new global model by averaging the received clients' models, our algorithm, `UPGA-PL`, updates the global model by a *pseudo-gradient* step; *ii)* `UPGA-PL` accounts for the potentially heterogeneous packet losses experienced by the clients to unbias the pseudo-gradient step. Still, `UPGA-PL` maintains the same computational and communication complexity as `FedAvg`. In our experiments, `UPGA-PL` not only outperforms existing state-of-the-art solutions for lossy channels (by more than 5 percentage points on test accuracy) but also matches `FedAvg`'s performance in lossless scenarios after less than 150 communication rounds.**

***Index Terms*—Federated Learning, Packet Loss.**

I. Introduction

Federated Learning (FL) [1], [2] involves a population of devices (typically referred to as *clients*) iteratively training a Machine Learning (ML) model over a network under the orchestration of a central server. At each global training round, the server sends the current global FL model to the clients, who individually train on their local datasets and send their locally-trained models back to the server. In many FL applications, such as training Google keyboard next-word prediction model, the clients are mobile devices such as smartphones or Internet of Things (IoT) devices, and the models are exchanged on wireless networks incurring potential transmission losses.

Lossy channels necessarily degrade the performance of FL training on wireless networks. Theoretical and experimental work [3]–[8] has shown that packet losses affect the quality of the final model towards which the FL training algorithms converge as well as their convergence rate. Under medium/high network background traffic, the authors in [3] measured a twofold training duration and a halved accuracy in the early stages of the training. Prior work [5]–[8] analyzed the convergence of state-of-the-art FL algorithms under different channel assumptions. Specifically, the authors of [8] proved the existence of a non-vanishing error due to the lossy channels, which prevents the convergence of their direct model aggregation scheme to the optimal model, and proposed to reduce this error by opportunely allocating resources (e.g., transmission power, radio blocks) to control packet losses. Similar approaches to the one proposed in [8] have been considered to mitigate the effect of packet loss on wireless networks, relying on automatic repeat request (ARQ) and forward error correction (FEC) techniques [9], [10].

Despite these efforts, packet losses are typically caused by external factors beyond the control of the orchestrating server and can therefore be unavoidable. Communication protocols may define a maximum number of retransmissions, but these retransmissions can still fail. More importantly, we point out that targeting high transmission reliability in FL-oriented applications may be sub-optimal, as it usually comes at the detriment of training time and/or resource usage, e.g., in terms of wider sub-channel bandwidth, higher energy consumption, or both. These issues are even more exacerbated in resource-constrained scenarios, such as the IoT, where increasing communication reliability may result in a reduced device lifetime or may not be feasible. Moreover, the iterative nature of the gradient methods used for ML model training makes them robust against limited errors at intermediate calculations [11].

For the aforementioned reasons, this paper diverges from prior work [4]–[10], which primarily focused on loss mitigation. Instead, we address the fundamental question of *whether FL algorithms can achieve optimal model convergence despite packet losses*. Our response is affirmative, necessitating only slight adjustments to the classic `FedAvg` [1] algorithm.

More in detail, we consider a FL framework where losses can occur in the downlink, uplink, or both, and loss probabilities can differ among clients. Indeed, the channel quality in wireless networks can vary according to per-user characteristics, such as the relative positioning of the transmitter and the receiver, the device transmission power, the selected frequency channel. As a result, the clients will not participate evenly in the training process, potentially leading to learning a biased ML model [12], [13]. Thus, the design of an aggregation strategy becomes critical to ensure the convergence of the FL model in the presence of packet losses. Previous works [8], [10] proposed direct model aggregation schemes, denoted in the following as DMA-PL and UDMA-PL. We will discuss and compare our approach to them in the rest of the paper.

This work makes the following novel contributions:

- We propose `UPGA-PL`, a novel algorithm that aggregates pseudo-gradients, instead of models, and considers the client loss probabilities. Its complexity is comparable to `FedAvg`;

This research was funded by the French National Research Agency (ANR) through the 3IA Côte d'Azur Investments in the Future project with reference number ANR-19-P3IA-0002; by Groupe La Poste, sponsor of Inria Foundation, in the framework of FedMalin Inria Challenge; by the U.S. Air Force Office of Scientific Research under contract number FA9550-23-1-0261; and by the U.S. Office of Naval Research under award number N00014-23-1-2221.

- We analytically prove UPGA-PL's convergence to the optimal model, the same model which would have been learned over ideal, lossless channels. This result proves UPGA-PL's ability to filter out the noise due to packet losses.
- We validate our analysis through numerical experiments. While losses largely affect the performance of the state-of-the-art algorithms [8], [10], UPGA-PL is robust to them and gains 5–10 percentage points on the test accuracy. Most importantly, even under severe losses, UPGA-PL achieves the same model's accuracy as FedAvg under lossless channels in less than 150 communication rounds.

II. Problem Description and Background

A central server and a set of clients $\mathcal{K} = \{1, \dots, K\}$ collaborate to train a global ML model $\boldsymbol{w} \in \mathbb{R}^n$ over a wireless network. For example, the model $\boldsymbol{w} \in \mathbb{R}^n$ can be the vector of parameters of a neural network architecture. Each client $k \in \mathcal{K}$ holds a local dataset D_k and has access to a loss function $\ell(\boldsymbol{w}, d_k) \to \mathbb{R}^+$ that evaluates the performance of the model $\boldsymbol{w}$ on a data sample $d_k \in D_k$. We define $F_k(\boldsymbol{w}) \coloneqq \frac{1}{|D_k|} \sum_{d_k \in D_k} \ell(\boldsymbol{w}, d_k)$ as the average loss computed after evaluating the performance of the model $\boldsymbol{w}$ on the k-th client's local dataset. The clients solve, under the coordination of the central server, the minimization of the global objective $F(\boldsymbol{w})$ via the following optimization problem:

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} \left[F(\boldsymbol{w}) \coloneqq \sum_{k \in \mathcal{K}} \alpha_k F_k(\boldsymbol{w}) \right], \tag{1}$$

where $\{\alpha_k\}_{k \in \mathcal{K}}$ are positive coefficients, chosen by the server, such that $\sum_{k \in \mathcal{K}} \alpha_k = 1$. They represent the weight assigned to each client's objective function F_k. Typical choices are: 1) $\alpha_k = 1/K \ \forall k \in \mathcal{K}$, the server giving equal weight to all clients, 2) $\alpha_k = |D_k|/|D|$, with $D = \cup_{k \in \mathcal{K}} D_k$, the server giving equal weight to each data sample.

Problem (1) is commonly solved through training algorithms executed for multiple rounds $\mathcal{T} = \{1, \dots, T\}$ during which server and clients communicate over a network. During communication round $t \in \mathcal{T}$, the server broadcasts the current global model $\boldsymbol{w}_t$ to the clients in $\mathcal{K}$. Each client $k \in \mathcal{K}$ initializes its local model $\boldsymbol{w}_{t,0}^k = \boldsymbol{w}_t$ and performs $E \geq 1$ local computations of Stochastic Gradient Descent (SGD):

$$\boldsymbol{w}_{t,j+1}^k = \boldsymbol{w}_{t,j}^k - \eta_t \nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \quad j = 0, \dots, E-1, \tag{2}$$

where $\eta_t > 0$ is an appropriately chosen learning rate, $\mathcal{B}_{t,j}^k$ is a random batch sampled from client-k's local dataset at round t and local iteration j, and $\nabla F_k(\cdot, \mathcal{B}) \coloneqq \frac{1}{|\mathcal{B}|} \sum_{d_k \in \mathcal{B}} \nabla \ell(\cdot, d_k)$ is an unbiased estimator of the local gradient $\nabla F_k(\cdot)$ evaluated on a random batch $\mathcal{B}$. After completing the local computations, each client $k \in \mathcal{K}$ produces a local model $\boldsymbol{w}_{t,E}^k$, and either transmits its local model or its model update $\Delta_t^k \coloneqq \boldsymbol{w}_{t,E}^k - \boldsymbol{w}_{t,0}^k = -\eta_t \sum_{j=0}^{E-1} \nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k)$ to the server, which aggregates and finally outputs a new version of the global model $\boldsymbol{w}_{t+1}$. The server can directly aggregate models computing $\boldsymbol{w}_{t+1}^{\text{DMA}} = \sum_{k \in \mathcal{K}} \alpha_k \boldsymbol{w}_{t,E}^k$: this scheme was introduced in the algorithm FedAvg [1], and we refer to it as Direct Model Aggregation (DMA). Alternatively, the server can consider the model updates Δ_t^k received by the clients as *pseudo-gradients*, aggregate them as $\Delta_t = \sum_{k \in \mathcal{K}} \alpha_k \Delta_t^k$, and finally apply a global pseudo-SGD step as $\boldsymbol{w}_{t+1}^{\text{PGA}} = \boldsymbol{w}_t + \Delta_t$. This aggregation scheme was proposed in [2]: it corresponds to FedOpt with SGD used both as server and client optimizer. We denote it as Pseudo-Gradients Aggregation (PGA).

The DMA and PGA aggregation schemes are equivalent under lossless channels. However, in typical FL applications the information is transmitted over lossy channels, which ultimately affect the workflow of the considered FL algorithm.

We consider the same scenario as in [8], [10]: due to downlink losses, only a subset of clients $\mathcal{P}_t^D \subseteq \mathcal{K}$ correctly receives the model $\boldsymbol{w}_t$ sent by the server and computes the local models $\{\boldsymbol{w}_{t,E}^k\}_{k \in \mathcal{P}_t^D}$. On the other hand, due to losses in the upstream, the server gathers the updates (either the models $\boldsymbol{w}_{t,E}^k$ or the pseudo-gradients Δ_t^k) only from a subset of clients $\mathcal{P}_t = \mathcal{P}_t^U \subseteq \mathcal{P}_t^D$. Note that if transmissions span multiple packets and only some packets are affected by losses, the recipient could still leverage the partial information correctly received. Previous literature [5], [6], [8]–[10] has ignored this possibility, which we plan to investigate in the future.

Since losses are random and can potentially differ among clients, the aggregation scheme plays an important role in the quality of the global ML model $\boldsymbol{w}_{t+1}$ learned by the FL training algorithm. Previous works [8], [10] considered the problem of FL training under lossy channels and proposed to generalize FedAvg's DMA aggregation strategy by letting the server aggregate all received models, i.e., the models of all clients $k \in \mathcal{P}_t$:

$$\boldsymbol{w}_{t+1}^{\text{DMA-PL}} = \frac{\sum_{k \in \mathcal{P}_t} \alpha_k \boldsymbol{w}_{t,E}^k}{\sum_{k \in \mathcal{P}_t} \alpha_k}. \tag{3}$$

We refer to this strategy as Direct Model Aggregation with Packet Loss (DMA-PL). The authors of [8] also analyzed the convergence of the DMA-PL aggregation scheme under the effect of packet losses. They showed the existence of a generally non-vanishing error between the model trained under a non-zero loss rate and the optimal model towards which the training converges in the absence of losses:

$$\mathbb{E}[F(\boldsymbol{w}_{t+1})] - F^* \leq \underbrace{A^t \left(F(\boldsymbol{w}_1) - F^*\right)}_{\text{vanishing term for small statistical heterogeneity}} + \underbrace{\frac{2\zeta_1}{L} \sum_{k \in \mathcal{K}} \alpha_k p_k \frac{1 - A^t}{1 - A}}_{\text{non-vanishing error due to statistical heterog. and packet loss}}, \tag{4}$$

where p_k denotes the probability that the server does not receive client-k's local model, $A = 1 - \frac{\mu}{L} + \frac{4\mu\zeta_2}{L} \sum_{k \in \mathcal{K}} \alpha_k p_k$, L and μ are the L-smooth and μ-strongly convex constants (they will be introduced in Assumptions 1, 2), and ζ_1, ζ_2 are parameters that quantify the *statistical heterogeneity* of the local datasets (the larger ζ_1 and ζ_2, the more heterogeneous the clients' data). We observe that, for non-zero loss probabilities and high statistical heterogeneity (large ζ_2), it is possible that the bound does not guarantee convergence (when $A \geq 1$). On

the contrary, for sufficiently small ζ_2, (4) predicts linear convergence to a neighborhood of the optimal solution, whose size is proportional to the loss probabilities $\{p_k\}_{k\in\mathcal{K}}$. Motivated by these results, reference [8] focuses on resource allocation to reduce loss probabilities and minimize the non-vanishing term.

Moreover, due to losses, only a subset of the clients contributes to updating the new model at each round. Previous works [12], [14] have studied partial client participation due to client sampling, i.e., when the server samples at each round a subset of clients $\mathcal{S}_t \subseteq \mathcal{K}$. Convergence results in [14] require *unbiased* sampling for DMA to converge to the optimal model, i.e., the sampling scheme should satisfy $\mathbb{E}_{\mathcal{S}_t}[\boldsymbol{w}_{t+1}] = \sum_{k\in\mathcal{K}} \alpha_k \boldsymbol{w}_{t,E}^k$ [14, Lemma 4], so that in expectation the k-th client contributes proportionally to its weight in the global objective (1). This observation suggests to unbias the DMA-PL scheme in (3) as follows:

$$\boldsymbol{w}_{t+1}^{\text{UDMA-PL}} = \sum_{k\in\mathcal{P}_t} \frac{\alpha_k}{1-p_k} \boldsymbol{w}_{t,E}^k, \tag{5}$$

so that the server counterbalances the more severe losses experienced by some clients with larger aggregation weights. We refer to this aggregation as Unbiased DMA-PL (UDMA-PL). However, by directly aggregating models, the UDMA-PL scheme suffers a possibly large variance due to the randomness in the set $\mathcal{P}_t$. Our analysis in Lemma 1 confirms that this variance leads to a non-vanishing term, which prevents UDMA-PL from converging to the optimal model. Moreover, our experimental results in Section IV confirm that UDMA-PL is not a practical solution.

In the next section, we present UPGA-PL, an unbiased aggregation scheme like UDMA-PL that filters out the noise due to losses and then succeeds in converging to the optimal model. To the best of our knowledge, only reference [6] showed a similar result for a decentralized FL algorithm, but it required uplink and downlink channels to have the same loss probabilities, which is uncommon in wireless networks.

III. Proposed Algorithm and Its Analysis

To solve the issues which characterized the DMA-PL and UDMA-PL aggregation schemes, we propose the Unbiased Pseudo-Gradient Aggregation strategy (UPGA-PL):

$$\boldsymbol{w}_{t+1}^{\text{UPGA-PL}} = \boldsymbol{w}_t + \sum_{k\in\mathcal{P}_t} \frac{\alpha_k}{1-p_k} \Delta_t^k. \tag{6}$$

Note that both UDMA-PL and UPGA-PL rely on the knowledge of the loss probabilities $\{p_k\}_{k\in\mathcal{K}}$. In practical scenarios, these probabilities can be estimated through channel measurements [15], [16].

As UDMA-PL, UPGA-PL is unbiased because it aggregates the pseudo-gradients with weights that compensate for clients' different loss probabilities. At the same time, by aggregating the pseudo-gradients $\{\Delta_t^k\}_{k\in\mathcal{P}_t}$ rather than the local models $\{\boldsymbol{w}_{t,E}^k\}_{k\in\mathcal{P}_t}$ (as UDMA-PL does), UPGA-PL can be seen as a stochastic approximation algorithm [17] with stepsize η_t (it is easy to verify that each Δ_t^k is proportional to η_t). Stochastic approximation theory suggests that convergence to the optimal model is guaranteed if η_t decreases fast enough to filter out the noise due to the randomness in the set $\mathcal{P}_t$ (i.e., $\sum_t \eta_t^2 < +\infty$), but also slow enough for the algorithm to be able to move from the initial tentative model ($\boldsymbol{w}_1$) to the optimal one (i.e., $\sum_t \eta_t = +\infty$). Our theoretical analysis below confirms these qualitative considerations: the UPGA-PL aggregation strategy enables the convergence of FL training algorithms to the optimal model even in the presence of lossy channels.

Algorithm 1: `UPGA-PL`

Input : Initial model $\boldsymbol{w}_1$; Weights $\boldsymbol{\alpha} = \{\alpha_k\}_{k\in\mathcal{K}}$;
Client loss probabilities $\boldsymbol{p} = \{p_k\}_{k\in\mathcal{K}}$;
Learning rates $\{\eta_t\}_{t\in\mathcal{T}}$; Local steps E.

1 **for** *global round* $t \in \mathcal{T}$ **do**
2 **for** *client* $k \in \mathcal{K}$, *in parallel* **do**
3 $\boldsymbol{w}_{t,0}^k = \boldsymbol{w}_t$;
4 **for** $j = 0, \dots, E-1$ **do**
5 $\boldsymbol{w}_{t,j+1}^k = \boldsymbol{w}_{t,j}^k - \eta_t \nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k)$;
6 $\Delta_t^k \leftarrow \boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t$;
7 Receive $\{\Delta_t^k\}$ from a subset $\mathcal{P}_t \subseteq \mathcal{K}$ of clients;
8 $\boldsymbol{w}_{t+1}^{\text{UPGA-PL}} \leftarrow \boldsymbol{w}_t + \sum_{k\in\mathcal{P}_t} \frac{\alpha_k}{1-p_k} \Delta_t^k$;

Output: Final model $\boldsymbol{w}_{T+1}$.

With abuse of language, we refer to the FL algorithm defined by the local update rule in (2) and the UPGA-PL aggregation scheme in (6) simply as `UPGA-PL`. The complete procedure is summarized in Algorithm 1. Similarly, we denote by `DMA-PL` and `UDMA-PL` the FL algorithms obtained replacing line 8 in Algorithm 1 with (3) and (5), respectively.

In the following, we analyze the convergence of `UPGA-PL`.

A. Convergence Analysis

For the analysis of the `UPGA-PL` algorithm, we make the following hypotheses. Assumptions 1 and 2 are standard in the literature on convex optimization [18, Sections 4.1, 4.2]. Assumptions 3 and 4 are standard hypothesis in the analysis of federated optimization algorithms [14], [19, Section 6.1].

Assumption 1. *$\{F_k\}_{k\in\mathcal{K}}$ are L-smooth: for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{w}) + \langle \nabla F_k(\boldsymbol{w}), \boldsymbol{v}-\boldsymbol{w}\rangle + \frac{L}{2}\|\boldsymbol{v}-\boldsymbol{w}\|_2^2$.*

Assumption 2. *$\{F_k\}_{k\in\mathcal{K}}$ are μ-strongly convex: for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{w}) + \langle \nabla F_k(\boldsymbol{w}), \boldsymbol{v}-\boldsymbol{w}\rangle + \frac{\mu}{2}\|\boldsymbol{v}-\boldsymbol{w}\|_2^2$.*

Assumption 3. *Let $\mathcal{B}_{t,j}^k$ be a random batch sampled from the k-th device's local data uniformly at random. The variance of stochastic gradients in each device is bounded: $\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\boldsymbol{w}_{t,j}^k)\right\|^2 \leq \sigma_k^2$ for $k \in \mathcal{K}$.*

Assumption 4. *The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k)\right\|^2 \leq G^2$ for $k \in \mathcal{K}$ and $t \in \mathcal{T}$, $j = 0, \dots, E-1$.*

We use the indicator variable ξ_t^k to denote the outcome of the t-th communication round between the server and the client k: ξ_t^k equals one if and only if the server correctly receives client-k's local model at round t.

Assumption 5. *At each round $t \in \mathcal{T}$, the communication outcomes $\{\xi_t^k\}_{k\in\mathcal{K}}$ are independent among clients. For each client $k \in \mathcal{K}$, the outcomes $\{\xi_t^k\}_{t\in\mathcal{T}}$ are independent and identically distributed (iid) over time with mean $\mathbb{E}[\xi_t^k] = 1 - p_k$.*

In Assumption 5, p_k denotes the probability that the overall communication between the server and client k fails either because client k does not receive the global model w_t or because later the server does not receive client-k's update Δ_t^k. If these events are independent, and p_{sk} and p_{ks} denote the downstream and upstream loss probabilities, respectively, then $p_k = 1 - (1 - p_{sk})(1 - p_{ks})$. If ARQ or FEC techniques are employed, then p_k can be interpreted as the residual loss probability experienced by the k-th client after potential retransmissions and/or error corrections, therefore our analysis remains agnostic to these methods.

Assumption 5 provides the flexibility for different loss probabilities across clients in the uplink and downlink transmissions, but does not consider spatial or temporal correlations, such as those arising from inter-channel interference or fading effects. We believe that results for Markov Chain gradient descent methods (where random samples are taken on the trajectory of a Markov chain) could be used to study the convergence of `UPGA-PL` under correlated channels [13], [20]. However, we defer this analysis to future work.

Convergence results for FL algorithms require to bound statistical heterogeneity in terms of some metric (e.g., [8], [12], [14]). We adopt the same metric introduced in [14]:

Definition 1. Let F^* and F_k^* be the minimum values of F and F_k, respectively. The parameter $\Gamma := F^* - \sum_{k\in\mathcal{K}} \alpha_k F_k^*$ quantifies the degree of data heterogeneity.

If the local datasets are identical, then the functions $\{F_k\}_{k\in\mathcal{K}}$ coincide and $\Gamma = 0$. In general, Γ is larger the more heterogeneous the local data distributions are.

Theorem 1 (proof in Appendix A) establishes `UPGA-PL` convergence under lossy channels. It builds upon [14], which considers the ideal lossless scenario. Our primary technical contribution is captured in Lemma 1 (Appendix A), with the additional term in (8) that accounts for the lossy channels.

Theorem 1 (Convergence under lossy channels)**.** *Let Assumptions 1–5 hold and L, μ, σ_k, G, p_k defined therein. Choose diminishing learning rates as $\eta_{t+1} = \frac{2/\mu}{8\kappa+t}$, with $\kappa := L/\mu$. Then, for each $t \in \mathcal{T}$, `UPGA-PL` satisfies:*

$$\mathbb{E}\left[F(\boldsymbol{w}_{t+1}^{\textit{UPGA-PL}})\right] - F^* \leq \underbrace{\frac{\kappa}{8\kappa + t}\left(\frac{2EC}{\mu} + 4L\left\|\boldsymbol{w}_1 - \boldsymbol{w}^*\right\|^2\right)}_{\textit{asymptotically vanishing term}}, \tag{7}$$

where:

$$C = \sum_{k\in\mathcal{K}} \alpha_k^2 \sigma_k^2 + 2(E-1)^2 G^2 + 6L\Gamma + \underbrace{EG^2 \sum_{k\in\mathcal{K}} \alpha_k^2 \frac{p_k}{1-p_k}}_{\textit{effect of lossy channels}}. \tag{8}$$

B. Discussion

a) `UPGA-PL` enables convergence under lossy channels: Theorem 1 proves that the objective $F(\boldsymbol{w})$, evaluated on the sequence of models $\{\boldsymbol{w}_t\}_{t>0}$ computed by `UPGA-PL`, converges in expectation to its minimum value F^*. Moreover, as the function F is strongly convex and then has a unique minimizer, the trained model converges also to the optimal one, i.e., $\lim_{t\to\infty} \mathbb{E}[\boldsymbol{w}_t^{\text{UPGA-PL}}] = \boldsymbol{w}^*$, where $\boldsymbol{w}^* \in \mathbb{R}^n \in \arg\min_{\boldsymbol{w}} F(\boldsymbol{w})$. The `UPGA-PL` aggregation strategy (with decreasing learning rates) does not suffer then from residual convergence errors as `DMA-PL` and `UDMA-PL` do.

b) The effect of packet losses on the convergence: The constant C (see (8)) quantifies the impact of lossy channels on the convergence in terms of the clients' loss probabilities $\{p_k\}_{k\in\mathcal{K}}$. As expected, the larger the loss probabilities, the larger is C and the slower the convergence predicted by the bound in (7). Moreover, the convergence rate in Theorem 1 ($\mathcal{O}(1/t)$) is comparable to the convergence rate of `FedAvg` in absence of losses under the same assumptions ($\mathcal{O}(1/(Et))$) [14].

c) Convergence speed vs. residual error: The bound in (4) suffers from a non-zero residual error but may achieve linear convergence (A^t decreases exponentially fast); our bound in (7) removes such error at the cost of a sublinear convergence rate, i.e., of slower convergence. One may then think that for a short duration of the training period, `DMA-PL` is preferable to `UPGA-PL`. In reality, the bound (4) achieves such a rate requiring the use of full gradients at each client (i.e., $\sigma_k^2 = 0$) and a single local gradient update at each communication round (i.e., $E = 1$) [8]; however, these assumptions do not correspond to FL practice [21].

IV. Experimental Results

A. Experimental setup

In the experiments, we consider a population with $K = 10$ clients. We split the population into two groups $G_i, i = 1, 2$, to which we associate different packet loss probabilities $p_i, i = 1, 2$. After evaluating different loss configurations, we present a challenging setting with $p_1 = 0.1$ and $p_2 = 0.9$.

We perform experiments on two datasets: the LEAF Synthetic Dataset for multinomial classification [22] and the real-world MNIST dataset for handwritten digit recognition [23]. To introduce statistical heterogeneity in the clients' datasets, we distribute the data among clients in a non-IID fashion. The LEAF Synthetic Dataset allows direct control of statistical heterogeneity through the parameters α and β: in our experiments, we set $\alpha = \beta = 1$. For MNIST, we generate a non-IID data distribution by splitting the labels among clients, with each client containing samples from only two classes [24]. We define Problem (1) with $\alpha_k = |D_k|/|D|$ $\forall k \in \mathcal{K}$.

For the Synthetic LEAF dataset, we train a linear classifier with a ridge penalization of parameter 5×10^{-4}, which defines a strongly convex objective function that well aligns with our theoretical assumptions. As for MNIST, we use a CNN architecture with two convolutional and two fully

connected layers, resulting in a non-convex objective function that introduces additional complexity to the learning process.

We compare UPGA-PL, in Algorithm 1, with DMA-PL (aggregation strategy in (3)), UDMA-PL (aggregation strategy in (5)), and an ideal lossless FedAvg (when $p_k = 0\ \forall k \in \mathcal{K}$). In the experiments, UDMA-PL and UPGA-PL rely on the knowledge of $\{p_k\}_{k\in\mathcal{K}}$. For all algorithms, we tuned the learning rate $\eta = \{\eta_t\}_{t\in\mathcal{T}}$ via grid search with values $\eta = \{10^{-3}, 10^{-2.5}, 10^{-2}, 10^{-1.5}, 10^{-1}\}$. The reported results are averaged over 10 random seeds.

B. Experimental results

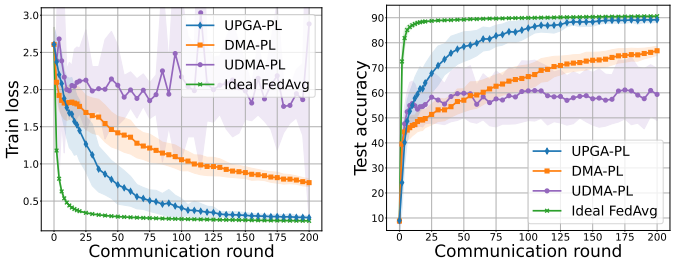
Figures 1–2 compare the train loss and test accuracy of DMA-PL, UDMA-PL, and UPGA-PL on the Synthetic LEAF and MNIST datasets. For both datasets, we include the reference performance of the ideal lossless FedAvg.

a) UPGA-PL outperforms the baselines and converges to the optimal model: The experimental results unanimously confirm the advantages of the UPGA-PL aggregation strategy in terms of train loss and test accuracy on the two datasets. Indeed, UPGA-PL improves the state-of-the-art solutions by 12 percentage points on the Synthetic LEAF dataset (Fig. 1b) and by 6 percentage points on the MNIST dataset (Fig. 2b). Moreover, UPGA-PL nearly attains the same performance as the FedAvg algorithm in lossless scenarios after around 150 communication rounds for both the Synthetic LEAF dataset (Fig. 1a) and the MNIST dataset (Fig. 2a).

In line with our theoretical analysis, the numerical experiments also confirm the effects of lossy channels on the convergence captured by Theorem 1 and discussed in Section III-B.

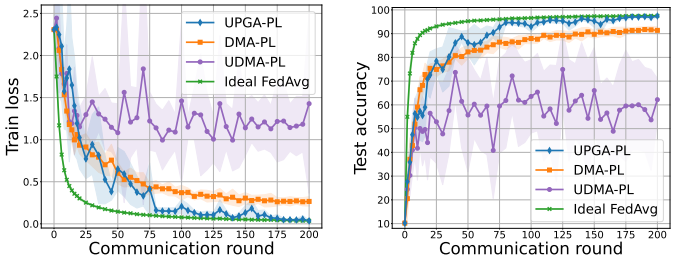
b) The effects of packet losses on the convergence: FL algorithms perform best under the ideal scenario with lossless channels ($p_k = 0\ \forall k \in \mathcal{K}$). Nevertheless, our experiments show that a severe amount of packet losses ($p_1 = 0.1, p_2 = 0.9$) slows down but does not prevent convergence to the optimal model, provided that UPGA-PL is used (UPGA-PL curve overlaps with FedAvg curve in the absence of losses).

c) Residual errors: DMA-PL and UDMA-PL suffer from non-vanishing errors. The residual error of DMA-PL is evident in Figure 2a, where its loss curve reaches a plateau around the value 0.3, while UPGA-PL converges to the value 0.0, as the ideal FedAvg. On the other hand, the UDMA-PL aggregation strategy, which should, in theory, unbias the DMA-PL scheme, does not filter the variance introduced by the lossy channels and suffers a noisy convergence: the UDMA-PL performance dramatically oscillates in the experiments, and its mean accuracy lies around 50–70%.



(a) Train loss (b) Test accuracy

Fig. 1: Train loss/test accuracy on the Synthetic LEAF dataset.



(a) Train loss (b) Test accuracy

Fig. 2: Train loss/test accuracy on the MNIST dataset.

V. Conclusions

This paper studied the problem of training FL algorithms over real-world wireless networks with lossy channels. We considered the presence of independent and identically distributed packet losses in the communication channels between the orchestrating server and the clients and we showed that the quality of the learned model is highly sensitive to the choice of the aggregation strategy. To mitigate the negative effects of packet losses, we proposed UPGA-PL, an algorithm that aggregates pseudo-gradients rather than models and that effectively converges to the optimal model under asymmetric lossy channels. While its complexity is comparable to FedAvg, under severe lossy settings UPGA-PL significantly outperformed the state-of-the-art solutions [8], [10] and attained very close performance to the optimal scenario with ideal, lossless channels at the cost of a slower convergence. Our work enabled optimal FL training under lossy channels, and—we believe—opened interesting research questions. For example, if losses affect only a part of the transmitted model, would it be possible for the clients or the server to take advantage of the partial information received instead of ignoring it (as DMA-PL, UDMA-PL, and UPGA-PL do)? What happens if the losses are correlated (e.g., due to inter-channel interference and/or fading)? What if they change over time?

Appendix

A. Proof of Theorem 1

For the proof, we define the sequence $\bar{\boldsymbol{w}}_{t,j} = \sum_{k\in\mathcal{K}} \alpha_k \boldsymbol{w}^k_{t,j}$. We denote:

$$\mathcal{B}_t = \{\mathcal{B}_{t,0}, \mathcal{B}_{t,1}, \ldots, \mathcal{B}_{t,E-1}\}; \tag{9}$$

$$\mathcal{H}_t = \{\mathcal{B}_1, \mathcal{P}_1, \mathcal{B}_2, \mathcal{P}_2, \ldots, \mathcal{B}_{t-1}, \mathcal{P}_{t-1}\}, \tag{10}$$

where $\mathcal{B}_{t,j} = \{\mathcal{B}^k_{t,j}\}_{k\in\mathcal{K}}$ is the set of random batches sampled at time (t, j) and $\mathcal{H}_t$ includes all history up to the t-th round.

Lemma 1. *Let Assumptions 4–5 hold, and* $\boldsymbol{w}_t = \boldsymbol{w}_t^{UPGA\text{-}PL}$. *Then:*

$$\mathop{\mathbb{E}}_{\mathcal{P}_t,\mathcal{B}_t|\mathcal{H}_t} \left\|\boldsymbol{w}_{t+1} - \bar{\boldsymbol{w}}_{t,E}\right\|^2 \leq \eta_t^2 E^2 G^2 \sum_{k\in\mathcal{K}} \alpha_k^2 \frac{p_k}{1-p_k}. \tag{11}$$

Conversely, for $\boldsymbol{w}_t = \boldsymbol{w}_t^{UDMA\text{-}PL}$, we have:

$$\mathop{\mathbb{E}}_{\mathcal{P}_t|\mathcal{H}_t,\mathcal{B}_t} \left\|\boldsymbol{w}_{t+1} - \bar{\boldsymbol{w}}_{t,E}\right\|^2 = \sum_{k\in\mathcal{K}} \alpha_k^2 \frac{p_k}{1-p_k} \left\|\boldsymbol{w}_{t,E}^k\right\|^2. \quad (12)$$

Proof of Lemma 1.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{P}_t|\mathcal{H}_t,\mathcal{B}_t} \left\| \boldsymbol{w}_t + \sum_{k\in\mathcal{K}} \frac{\alpha_k}{1-p_k} \left(\boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t\right) \xi_t^k - \bar{\boldsymbol{w}}_{t,E} \right\|^2 \\
&= \mathbb{V}\mathrm{ar}\left(\sum_{k\in\mathcal{K}} \frac{\alpha_k}{1-p_k} \left(\boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t\right) \xi_t^k \right) = \\
&= \sum_{k\in\mathcal{K}} \mathbb{V}\mathrm{ar}\left(\frac{\alpha_k}{1-p_k} \left(\boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t\right) \xi_t^k \right) = \\
&= \sum_{k\in\mathcal{K}} \frac{{\alpha_k}^2}{(1-p_k)^2} \left\|\boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t\right\|^2 \mathrm{Var}\left(\xi_t^k\right) = \\
&= \sum_{k\in\mathcal{K}} \alpha_k^2 \frac{p_k}{1-p_k} \left\|\boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t\right\|^2. \qquad (13)
\end{aligned}$$

Finally:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{P}_t,\mathcal{B}_t|\mathcal{H}_t} \left\|\boldsymbol{w}_{t+1} - \bar{\boldsymbol{w}}_{t,E}\right\|^2 &= \sum_{k\in\mathcal{K}} \alpha_k^2 \frac{p_k}{1-p_k} \mathop{\mathbb{E}}_{\mathcal{B}_t|\mathcal{H}_t} \left\|\boldsymbol{w}_{t,E}^k - \boldsymbol{w}_t\right\|^2 \\
&\leq \sum_{k\in\mathcal{K}} \alpha_k^2 \frac{p_k}{1-p_k} \eta_t^2 E^2 G^2. \qquad (14)
\end{aligned}$$

Conversely, for $\boldsymbol{w}_t = \boldsymbol{w}_t^{\text{UDMA-PL}}$, the same proof technique leads to the bound in (12), but the steps in (14) do not hold. □

Proof of Theorem 1.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{P}_t|\mathcal{H}_t,\mathcal{B}_t} \left\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\right\|^2 = \\
&= \mathop{\mathbb{E}}_{\mathcal{P}_t|\mathcal{H}_t,\mathcal{B}_t} \left\|\boldsymbol{w}_{t+1} - \bar{\boldsymbol{w}}_{t,E}\right\|^2 + \left\|\bar{\boldsymbol{w}}_{t,E} - \boldsymbol{w}^*\right\|^2. \qquad (15)
\end{aligned}$$

From [14, Lemma 1], [14, Lemma 2], and [14, Lemma 3], recursively:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{B}_t|\mathcal{H}_t} \left\|\bar{\boldsymbol{w}}_{t,E} - \boldsymbol{w}^*\right\|^2 \leq \\
&\leq (1-\eta_t\mu)^E \left\|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\right\|^2 + \eta_t^2 B \sum_{j=0}^{E-1} (1-\eta_t\mu)^j \qquad (16) \\
&\leq (1-\eta_t\mu) \left\|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\right\|^2 + \eta_t^2 EB, \qquad (17)
\end{aligned}$$

where $B = \sum_{k\in\mathcal{K}} \alpha_k^2 \sigma_k^2 + 6L\Gamma + 2(E-1)^2 G^2$.

Combining (15) and (17), and applying Lemma 1, we have:

$$\mathop{\mathbb{E}}_{\mathcal{P}_t,\mathcal{B}_t|\mathcal{H}_t} \left\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\right\|^2 \leq (1-\eta_t\mu) \left\|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\right\|^2 + \eta_t^2 EC. \quad (18)$$

The conclusion of the proof follows similar steps as [14, Theorem 1]. We require a learning rate $\eta_t \leq (\frac{1}{\mu}, \frac{1}{4L}) = \frac{1}{4L}$. Set $\eta_{t+1} \leq \frac{2/\mu}{8\kappa+t}$, with $\kappa := L/\mu$, such that $\eta_1 = \frac{1}{4L}$. Then:

$$\mathbb{E}\left[F(\boldsymbol{w}_{t+1})\right] - F^* \leq \frac{\kappa}{8\kappa+t} \left(\frac{2EC}{\mu} + 4L \left\|\boldsymbol{w}_1 - \boldsymbol{w}^*\right\|^2 \right). \quad (19)$$

□

REFERENCES

[1] B. McMahan, E. Moore *et al.*, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1273–1282.

[2] S. J. Reddi, Z. Charles *et al.*, "Adaptive Federated Optimization," in *International Conference on Learning Representations*, 2021.

[3] M. C. Eriş, B. Kantarci, and S. Oktug, "Unveiling the Wireless Network Limitations in Federated Learning," in *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Dec. 2021, pp. 262–267, iSSN: 2334-3125.

[4] R. Chandrasekaran, K. Ergun *et al.*, "FHDnn: Communication efficient and robust federated learning for AIoT networks," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 37–42.

[5] H. H. Yang, Z. Liu *et al.*, "Scheduling Policies for Federated Learning in Wireless Networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[6] H. Ye, L. Liang, and G. Y. Li, "Decentralized Federated Learning With Unreliable Communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 487–500, Apr. 2022.

[7] E. Baccarelli, M. Scarpiniti *et al.*, "AFAFed—Asynchronous Fair Adaptive Federated learning for IoT stream applications," *Computer Communications*, vol. 195, pp. 376–402, Nov. 2022.

[8] M. Chen, Z. Yang *et al.*, "A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[9] D. Wen, X. Li *et al.*, "An Overview of Data-Importance Aware Radio Resource Management for Edge Machine Learning," *Journal of Communications and Information Networks*, vol. 4, no. 4, pp. 1–14, 2019.

[10] X. Su, Y. Zhou *et al.*, "On Model Transmission Strategies in Federated Learning With Lossy Communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1173–1185, Apr. 2023.

[11] E. P. Xing, Q. Ho *et al.*, "Strategies and Principles of Distributed Machine Learning on Big Data," *Engineering*, vol. 2, no. 2, pp. 179–195, Jun. 2016.

[12] J. Wang, Q. Liu *et al.*, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 7611–7623.

[13] A. Rodio, F. Faticanti *et al.*, "Federated Learning under Heterogeneous and Correlated Client Availability," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, May 2023.

[14] X. Li, K. Huang *et al.*, "On the Convergence of FedAvg on Non-IID Data," in *International Conference on Learning Representations*, Apr. 2020.

[15] P. Benko and A. Veres, "A Passive Method for Estimating End-to-End TCP Packet Loss," in *IEEE Global Telecommunications Conference, 2002. GLOBECOM '02*, vol. 3, Nov. 2002, pp. 2609–2613 vol.3.

[16] M. Yajnik, S. Moon *et al.*, "Measurement and Modelling of the Temporal Dependence in Packet Loss," in *IEEE INFOCOM '99. Conference on Computer Communications.*, vol. 1, Mar. 1999, pp. 345–352 vol.1.

[17] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Springer, 2009, vol. 48.

[18] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, Jan. 2018.

[19] J. Wang, Z. Charles *et al.*, "A Field Guide to Federated Optimization," *arXiv:2107.06917 [cs]*, Jul. 2021.

[20] T. Sun, Y. Sun, and W. Yin, "On Markov Chain Gradient Descent," *arXiv:1809.04216 [math, stat]*, Sep. 2018.

[21] P. Kairouz, H. B. McMahan *et al.*, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[22] S. Caldas, S. M. K. Duddu *et al.*, "LEAF: A Benchmark for Federated Settings," *arXiv:1812.01097 [cs, stat]*, Dec. 2019.

[23] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[24] I. Achituve, A. Shamsian *et al.*, "Personalized Federated Learning With Gaussian Processes," in *Advances in Neural Information Processing Systems*, Oct. 2021.