

FL-Broadcast: Efficient Decentralized Learning on Wireless Networks

Yann Busnel
Institut Mines-Télécom, IRISA
Palaiseau, France

Léo Lavaur
University of Luxembourg, SnT
Luxembourg-Ville, Luxembourg

Hervé Rivano
INSA Lyon, INRIA, CITI
Villeurbanne, France

Abstract—By showing that it was possible to make learning models converge on distributed data without centralizing them, federated learning has renewed the way collaborative learning problems are considered. To fully distribute the architecture, a resurgence of decentralized algorithms can thus be observed, such as gossip learning. All these approaches nevertheless involve substantial exchanges between the nodes of the network, which can become problematic in the case of wireless networks, where bandwidth is limited because of the shared medium. To limit communication costs and improve model convergence, we propose here a new decentralized learning approach, **FL-Broadcast**, which exploits the common medium to disseminate model updates. This paper introduces the foundations of the **FL-Broadcast** algorithm and highlights, through evaluation, its advantages over gossip learning on wireless networks.

Index Terms—gossip learning, decentralized learning, federated learning, wireless networks, broadcast.

I. INTRODUCTION

The emergence of federated learning (FL) [1] marked a turning point in the field of collaborative learning by making it possible to train models on distributed data without requiring their centralization or sharing. This approach preserves data privacy while reducing communication cost with respect to distributed stochastic gradient descent (DSGD). The algorithm itself nevertheless remains centralized, since a server orchestrates the learning tasks and takes charge of collecting and aggregating the models. Decentralized architectures, which make it possible to remove this dependency on a central server (such as gossip learning), often translate into higher communication costs [2].

At first glance, the number of communications can prove limiting when deployment takes place on wireless networks, in particular because of the shared transmission medium and half-duplex communications. Indeed, contention alone significantly increases latency and decreases the efficiency of protocols based on *unicast* data exchanges, as soon as their size becomes significant. This is true even under the hypothesis of an ideal MAC layer, *e.g.*, a simplified model of the 802.11 CSMA/CA MAC layer, abstracting away bandwidth fluctuations or the probability of packet losses due to interference. These constraints are exacerbated in dynamic environments, such as drone or smartphone mesh networks, where node mobility and density complicate model synchronization, as well as in low-bandwidth use cases, like sensor networks.

To address these issues, we propose **FL-Broadcast**, an innovative approach that exploits the very nature of the wireless medium to disseminate model updates in a simultaneous and efficient manner. By taking advantage of a *broadcast* communication model, our method significantly reduces the number of transmissions required, thus accelerating model convergence while minimizing the use of network resources. This paper presents the algorithmic concepts of **FL-Broadcast**, analyzes its performance, and illustrates its advantages compared to unicast gossip learning on a physical topology, when wireless network properties are considered.

II. PROBLEM STATEMENT AND MODELING

Epidemic protocols, as used in *gossip learning* [2], often rely on an *overlay*. This virtual graph is constructed over of the network's physical topology and connects nodes independently of their physical neighbors. While it provides the capability to adjust the virtual topology, *e.g.*, for improving the convergence of the federated learning [3], such an overlay comes at a price. The required number of messages for routing, retransmissions, or neighbor discovery, can quickly saturate a constrained communication medium, in particular in random access wireless networks.

FL-Broadcast aims to optimize both network usage and model convergence when epidemic learning is deployed on wireless networks without an overlay. In this opposite trade-off, the overhead of the virtual topology is spaded, but the model exchanges are limited to the physical connectivity of the nodes and cannot be adjusted.

We study the problem under the following assumptions:

- Each node transmits with an omnidirectional radio that allows it to communicate in *unicast* or in *broadcast* with the nodes within range. All nodes have the same range.
- Communication time is significantly shorter than local learning time.

We use the simplistic yet classic unit disk graph $G = (V, E)$ network model: each node $v \in V$ represents a learning participant, and each edge $\{u, v\} \in E$ indicates that nodes u and v are within radio range and can communicate. We assume that a single communication channel is used, for which contention is resolved by a perfect (lossless) CSMA/CA MAC layer: if a node u transmits, then $\forall \{u, w\} \in E$, w can neither transmit nor receive any other message.

III. FL-BROADCAST

FL-Broadcast adapts the epidemic learning algorithm proposed by Hegedűs et al. [2], where the participants share their models, similarly to FedAvg [1]. Each client executes two loops in parallel and without synchronization with the others. The first processes receptions: as long as it is not transmitting, a client u is constantly listening and collects the model broadcasts within its communication range. The models are shared alongside a timestamp t_u indicating the time of the update (timestamp since initialization). The models are recorded locally in a collection W_u of tuples (t_u, ω_u) , where ω_u represents the model parameters. Each model is present there only once and is replaced if a newer model from the same sender is received. The second loop is the training loop. Each client performs a predefined learning task (by default one *epoch* of stochastic gradient descent—SGD) and updates its model ω_u . Upon completion, it sends its model to its neighbors as soon as the network is available, or after waiting for a synchronization beacon (*beacon*). When it has completed its learning, it sends its model in turn as soon as the network is available, potentially waiting for the synchronization beacon (*beacon*). In unicast mode, a breadth-first search allowed it to discover its neighbors; it then selects one at random and sends its model to it via broadcast, which is not necessary. Learning can be repeated locally as long as the conditions are not met (*continuous* mode) or performed only once per loop (*reactive* mode). Finally, the client adds its own model to W_u and aggregates its content as an average weighted by the age of the models. A pseudocode representation of the algorithm is provided in Algorithm 1.

IV. EVALUATION METHODOLOGY

To confirm the efficiency of FL-Broadcast on a wireless network, the evaluation reuses the discrete event-based simulator developed for FTM-Broadcast [4] and adapts it to decentralized learning. This simulator allows modeling a wireless network with nodes that communicate in unicast or broadcast, while considering the constraints specific to this type of networks (*e.g.*, communication time, contention, random backoff). The simulator takes the network topology as input, so we evaluate FL-Broadcast on two 14-node topologies: a *clique*, where each node can communicate with all the others, and a *sparse* topology of diameter 7, obtained from a random draw (depicted in Figure 1). The network simulator also covers the algorithmic aspect of decentralized learning, such as local computation time, aggregation of received models, or sharing of models with neighbors. The simulator outputs a trace of the events that occur during the simulation, which we convert to a *dependency graph* to schedule the different aggregation and computation operations, before executing them concretely using Flower¹, a federated learning framework in Python. Each client performs one local SGD epoch per round, and the reported curves are computed from the global evaluation loss returned by Flower after each

¹<https://flower.ai>

Algorithm 1 Algorithm FL-Broadcast. ω_u is the model of client u , trained at time t_u on its local dataset $\mathcal{D}_u$ by performing gradient descent on its objective function $\mathcal{L}$. T is the beacon interval, η the learning rate, and W_u the list of models collected by u since the last aggregation.

```

1:  $W_u \leftarrow \emptyset$  ▷ List of models at client  $u$ 
2:  $t_{\text{last\_send}} \leftarrow 0$  ▷ Last sending timestamp
3: procedure TRAININGLOOP
4:   loop
5:      $\omega_u \leftarrow \omega_u - \eta \nabla_{\omega_u} \mathcal{L}(\omega_u; \mathcal{D}_u)$ 
6:      $t_u \leftarrow \text{get\_timestamp}()$  ▷ Current timestamp
7:     if  $T_{\text{beacon}} > 0$  then
8:       if  $t_u - t_{\text{last\_send}} \geq T$  then
9:          $\text{SEND}((t_u, \omega_u))$ 
10:         $t_{\text{last\_send}} \leftarrow t_u$ 
11:       end if
12:     else
13:        $\text{SEND}((t_u, \omega_u))$  ▷ Sending after each iteration
14:     end if
15:     if  $W_u \neq \emptyset$  then
16:        $W_u \leftarrow W_u \cup \{(t_u, \omega_u)\}$  ▷ Add local model
17:        $\omega_u \leftarrow \text{TimeWeightedAverage}(W_u)$ 
18:        $W_u \leftarrow \emptyset$ 
19:     end if
20:   end loop
21: end procedure
22: procedure RECEPTIONLOOP
23:   loop
24:     if reception of  $(t_j, \omega_j)$  then
25:        $W_u \leftarrow W_u \cup \{(t_j, \omega_j)\}$ 
26:     end if
27:   end loop
28: end procedure
29: function TIMEWEIGHTEDAVERAGE( $W_u$ )
30:   return  $\frac{\sum_{(t_j, \omega_j) \in W_u} t_j \cdot \omega_j}{\sum_{(t_j, \omega_j) \in W_u} t_j}$ 
31: end function

```

aggregation. For this study, the learning task uses CIFAR-10 data split over 14 nodes according to a non-IID Dirichlet partition ($\alpha = 0.5$), so that the class distribution remains strongly unbalanced across clients. As previously stated, the CSMA/CA MAC layer is assumed perfect and lossless to isolate the effect of contention on model dissemination; the gains reported here should therefore be read as the outcome of a controlled communication model rather than as an end-to-end wireless throughput claim. In doing so, it is possible to compare FL-Broadcast with the classical federated algorithm FedAvg [1], by measuring convergence as a function of real elapsed time, but also in number of *rounds* (local computation steps performed simultaneously by each node).

V. EVALUATION RESULTS

To validate the implementation of FL-Broadcast, we compare its convergence—measured in terms of the num-

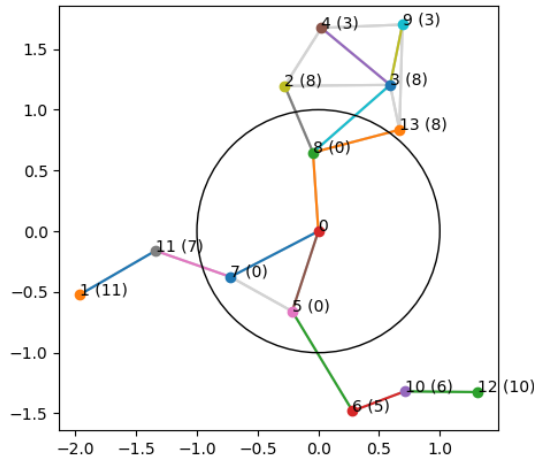


Figure 1. Sparse topology (random unit disk graph) used for the evaluation of FL-Broadcast. The arcs of a BFS tree are in color to help visualizing the diameter of 7. The other edges are displayed in light gray.

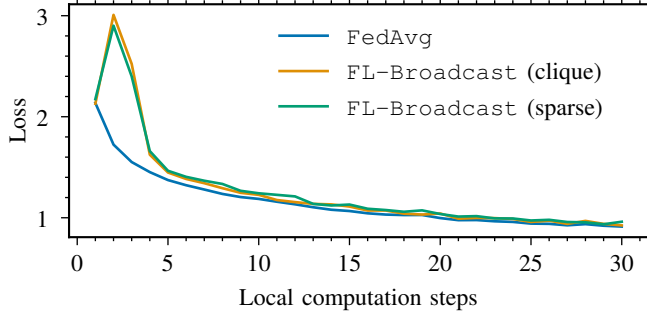


Figure 2. Comparison of convergence between FedAvg and FL-Broadcast with two types of network topologies. The global evaluation loss returned by Flower after each round is presented as a function of local computation steps.

ber of local computation steps—with that of FedAvg (*cf.* Figure 2). FL-Broadcast is deployed in broadcast and reactive mode on a complete clique topology, with a 20-second synchronization beacon. Since this duration is much larger than the sum of the communication and computation times, all nodes were able to send and receive the models of all the others before each aggregation, which makes the behavior of FL-Broadcast comparable to that of FedAvg. Only the model weighting differs: where FedAvg weights the models according to the number of data points [1], FL-Broadcast uses a weighting according to the age of the models received at aggregation time. However, in this scenario, the age of the models is proportional to local computation time, itself proportional to the number of data points. The two approaches are therefore similar, as confirmed by the Figure 2: the curves follow each other, once a few initialization rounds have passed. The performance of FL-Broadcast on the sparse topology is also comparable to that of centralized learning, even in the absence of direct communication between each pair of nodes. The remaining analyses focus on this sparse topology, which is more representative of real wireless networks.

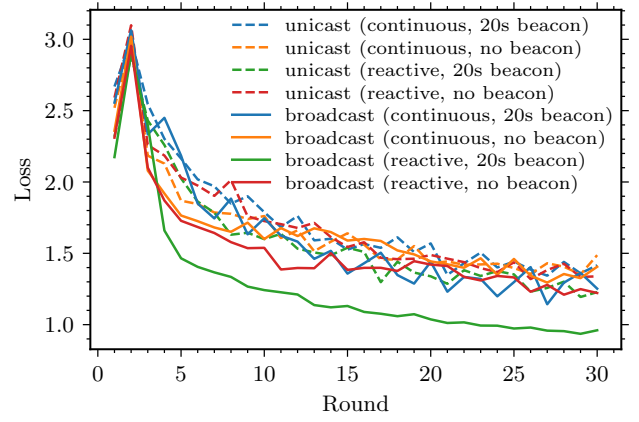


Figure 3. Comparison of convergence between the unicast (equivalent to classical gossip learning) and broadcast versions of FL-Broadcast, with several combinations of parameters, as a function of local computation steps.

Figure 3 compares the convergence of FL-Broadcast as a function of its communication mode (unicast *vs.* broadcast), the presence or absence of a synchronization beacon, and the learning mode (continuous *vs.* reactive). Observed in number of rounds, convergence is similar among the different configurations, with the notable exception of reactive mode with beacon and broadcast communication, which converges faster. In this configuration, the nodes can indeed aggregate more models at each step, benefiting both from the local dissemination of models among radio neighbors and from the synchronization provided by the beacon. The other configurations show comparable performance, indicating that the efficiency of learning remains preserved, whatever the communication or synchronization mode.

Figures 4a and 4b present the real-time convergence of FL-Broadcast according to its configurations. Here, the differences are more marked: broadcast mode systematically outperforms unicast mode, thanks to more efficient use of the communication medium. This superiority is observed both in terms of convergence speed (*i.e.*, the time needed to reach a given performance) and in terms of final performance reached after a given number of operations. The presence of a synchronization beacon logically slows real-time convergence, because of the waiting periods imposed on the nodes.

The superiority of FL-Broadcast in the presented scenarios is explained by several factors. First, the use of broadcast allows a node to transmit its model to all its neighbors in a single operation, thus reducing the total number of transmissions required. Second, the implementation of the epidemic algorithm means that a node can propagate its model only after updating it with the models received from its neighbors. Thus, a broadcast transmission will trigger simultaneous updates in several nodes, maximizing the impact of each communication, whereas a unicast communication benefits only a single node at a time. In this configuration, epidemic learning can no longer be assimilated to independent random walks, but rather to a wave-like diffusion of information.

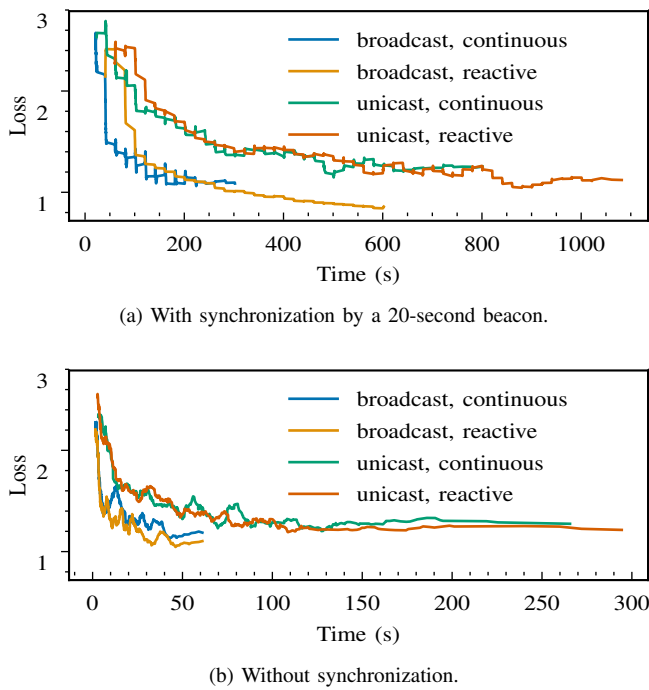


Figure 4. Comparison of convergence of the different versions of FL-Broadcast as a function of elapsed simulation time. The plots present the smoothed average (over 20 values) of the local evaluation loss reported after each aggregation step.

VI. RELATED WORK

Optimizing the communication of FL over a wireless network has already received attention in recent years, notably because its centralized nature is suboptimal for mesh wireless networks. Mertens et al. [5] propose to use gossip averaging (*i.e.*, average models as they are moved through the network) to minimize communication when clients report their models to the server. With similar network assumptions, FedBroadcast [6] leverages broadcasts to decrease the number of messages sent for each step of the process. However, unlike FL-Broadcast, both still assume federated learning in the strict sense and use a central server for most of the aggregation. Chen et al. [7] study synchronous D-SGD over wireless networks with probabilistic random access under a collision model, and optimize a common access probability to maximize the expected number of successful links. In contrast, FL-Broadcast adapts asynchronous gossip learning (*i.e.*, nodes collect and aggregate full models received from their radio neighbors) and transmissions are governed by channel availability and an optional beacon rather than a random-access probability that depends on the graph's topology. Despite its simplifying assumptions, this work simulates a complete MAC layer with a more realistic implementation.

VII. CONCLUSION

This work presents FL-Broadcast, an optimization of epidemic learning for wireless networks, taking advantage of broadcast communication to improve the efficiency of model exchanges. The evaluation, carried out on a simulator taking

into account the constraints specific to wireless networks, suggests the superiority of FL-Broadcast with respect to a classical unicast implementation, both in terms of convergence speed and final performance reached. This preliminary work nevertheless calls for extension in several respects. First, the evaluation presented above is limited to two topologies, one dataset, and one non-IID partition, under a perfect MAC model, which limit assessing its robustness and generalizability. Moreover, in a real deployment, time-weighted aggregation would require sufficiently synchronized clocks. Future work will evaluate the impact of clock offset and drift on this weighting and explore alternative aggregation strategies that are more robust to these factors. Although the advantage of using broadcast in this scenario is intuitive and supported for other types of algorithms in the literature [8], it would also benefit from being demonstrated formally. Finally, an algorithm of this type seems particularly suited to dynamic deployments where topology may evolve over time, whereas the present contribution focuses on a static topology.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, [Communication-efficient learning of deep networks from decentralized data](#). In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.
- [2] I. Hegedűs, G. Danner, and M. Jelasity, [Gossip Learning as a Decentralized Alternative to Federated Learning](#). In *Distributed Applications and Interoperable Systems*, 2019.
- [3] B. Le Bars, A. Bellet, M. Tommasi, E. Lavoie, and A.-M. Kermarrec, [Refined convergence and topology learning for decentralized sgd with heterogeneous data](#). In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, 2023.
- [4] Y. Busnel and H. Rivano, [FTM-Broadcast: Efficient Network-wide Ranging](#). In *2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2023.
- [5] J. Mertens, L. Galluccio, and G. Morabito, [Federated learning through model gossiping in wireless sensor networks](#). In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2021.
- [6] H. Tian, H. Zhang, J. Jia, M. Dong, and K. Ota, [Fed-Broadcast: Exploit Broadcast Channel for Fast Convergence in Wireless Federated Learning](#). *IEEE Internet of Things Journal*, 2023.
- [7] Z. Chen, M. Dahl, and E. G. Larsson, [Decentralized Learning over Wireless Networks: The Effect of Broadcast with Random Access](#). In *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023.
- [8] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, [Broadcast Gossip Algorithms for Consensus](#). *IEEE Transactions on Signal Processing*, 2009.