A Joint Decentralized Federated Learning and Communications Framework for Industrial Networks

Stefano Savazzi, Sanaz Kianoush, Vittorio Rampa
Consiglio Nazionale delle Ricerche (CNR)
IEIIT institute, Milano
Email: {name.surname}@ieiit.cnr.it

Mehdi Bennis
Centre for Wireless Communications
University of Oulu, Finland
Email: mehdi.bennis@oulu.fi

***Abstract*—Industrial wireless networks are pushing towards distributed architectures moving beyond traditional server-client transactions. Paired with this trend, new synergies are emerging among sensing, communications and Machine Learning (ML) co-design, where resources need to be distributed across different wireless field devices, acting as both data producers and learners. Considering this landscape, Federated Learning (FL) solutions are suitable for training a ML model in distributed systems. In particular, decentralized FL policies target scenarios where learning operations must be implemented collaboratively, without relying on the server, and by exchanging model parameters updates rather than training data over capacity-constrained radio links. This paper proposes a real-time framework for the analysis of decentralized FL systems running on top of industrial wireless networks rooted in the popular Time Slotted Channel Hopping (TSCH) radio interface of the IEEE 802.15.4e standard. The proposed framework is suitable for neural networks trained via distributed Stochastic Gradient Descent (SGD), it quantifies the effects of model pruning, sparsification and quantization, as well as physical and link layer constraints, on FL convergence time and learning loss. The goal is to set the fundamentals for comprehensive methods and procedures supporting decentralized FL pre-deployment design. The proposed tool can be thus used to optimize the deployment of the wireless network and the ML model before its actual installation. It has been verified based on real data targeting smart robotic-assisted manufacturing.**

I. Introduction

Federated Learning (FL) [1], [2] is emerging as an effective method to train Machine Learning (ML) models in distributed systems. The ML model parameters, *i.e.* the weights and biases of Neural Network (NN) layer, are optimized collectively by devices, acting as both training data producers and local learners. Compared with conventional edge ML relying on training data exchange, FL decouples the ML stages from the need to send local training data to the server. Sending training data might not be feasible for privacy issues, or critical designs that require extremely low latency when moving large volume of data, or with intermittent/limited communication links.

Early approaches to FL, such as federated averaging [1] allowed the networked devices to build a joint model with the help of a remote Parameter Server (PS) and by implementing a distributed Stochastic Gradient Descent (SGD) algorithm [3]. However, beyond 5G technologies envisioned for next generation Industrial Internet of Things (IIoT) systems [4] are pushing toward massively dense and fully decentralized networks that do not rely upon a central server, and where the

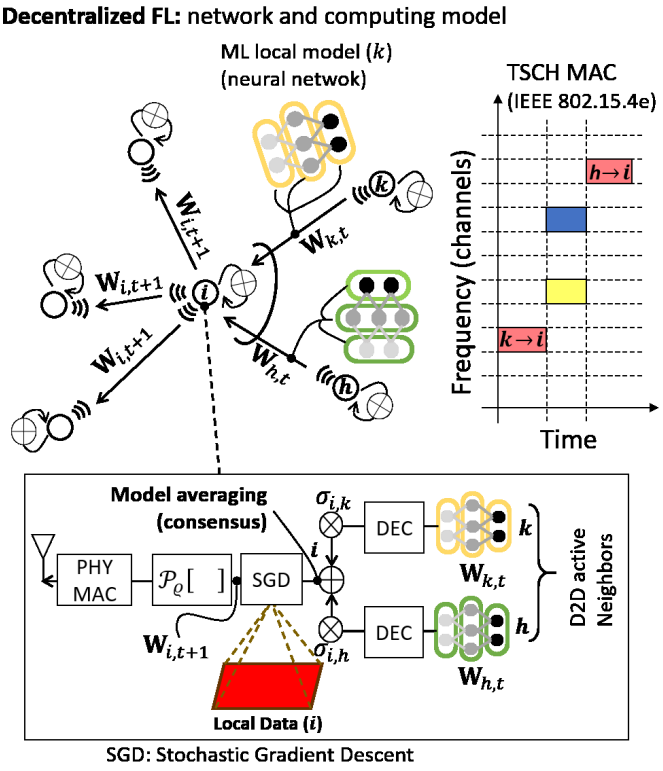

Fig. 1. Decentralized FL rooted at the Time Slotted Channel Hopping (TSCH) MAC layer of the IEEE 802.15.4e standard. Communication and computational model: model averaging, SGD on local data, ML model parameters pruning, compress and forward of parameters on each FL round.

cost of Device-to-Device (D2D) communications, in terms of energy, computational power, bandwidth and channel uses, is much lower than the cost of a long-range server connection.

Fully decentralized solutions to FL based on a distributed implementation of SGD [3] have been thus recently proposed in [5], [6], [7]. In decentralized FL, devices rely *solely* on local cooperation with neighbors, and in-network (as opposed to centralized) processing. Cooperative training of the ML model parameters, *i.e.* collected in the matrix $\mathbf{W}$, uses a distributed mesh network as backbone, typically designed for Ultra-Reliable and Low-Latency Communications (URLLC). As shown in the example of Fig.1, a device (i) receives (and

decode/reconstruct) the model updates $\mathbf{W}_{k,t}$, $\mathbf{W}_{h,t}$ from the neighbors k, h at time t over the short range radio links. Next, it upgrades the local parameters $\mathbf{W}_{i,t}$ sequentially: first, by averaging the contributions of neighbors, and then, by running SGD rounds on local data (mini)batches. The device encodes and compresses the updated local ML model parameters $\mathbf{W}_{i,t+1}$, *e.g.* using sparsity constraints, and forwards them to the neighbors using the transmission resources assigned by the Medium Access Control (MAC) layer and the scheduling policies. Finally, it waits for a new round, until a desired learning loss is obtained.

Contributions: the paper proposes a real-time simulation framework designed to analyze the performance of decentralized FL over arbitrarily complex wireless mesh network structures, limited by fading and rooted in the industry standard Time Slotted Channel Hopping (TSCH) MAC layer [17]. Existing frameworks [8], [10] limit their applicability to conventional FL relying on a PS server. On the contrary, the proposed tool real-time simulates the decentralized FL training process using federated datasets of any size as inputs and accounts for hundreds of distributed communication rounds, that involve short range D2D communications over an arbitrary graph structure. The paper considers an industrial wireless network architecture, consisting of dense, massively interacting groups of wireless field devices, namely machines with computational capabilities that move far beyond traditional sensors. In particular, in line with FL systems, every field device supports: *i)* data- or model-driven ML tools and learning of complex models, typically with 20K+ parameters; *ii)* autonomous decision making in complex and adaptive situations (*e.g.,* mobility, human-machine interfacing, time-varying environments); *iii)* URLLC networking [9] rooted in the TSCH MAC of the IEEE 802.15.4e PHY layer. The standard is designed for cable replacing in industrial processes and targets time-sensitive, energy-efficient and reliable distributed networking. However, it is subject to throughput and spectral efficiency limitations that demand for an efficient optimization of the communication resources. In industrial processes, continual training of the ML model is critical to track frequent changes of data dynamics. Considering a TSCH MAC policy [17], we analyze the key factors that rule both the learning loss and the convergence time (latency), by quantifying the time span of each FL communication round, and how this is affected by model parameters digitalization and compression, the graph structure, and the number of cooperating neighbors. The proposed framework thus targets the development of consistent design methodologies for the joint optimization of the decentralized learning operations and the distributed networking.

The paper is organized as follows. Sect. II introduces the decentralized FL paradigms based on gossip and consensus [3], [5], [6], and physical (PHY) communication design aspects, including ML model pruning, quantization and communication over wireless channels limited by fading impairments. Sect. III introduces the proposed network simulation framework and the related MAC layer adaptations for the implementation of decentralized FL on top of a TSCH compliant wireless interface. Based on an extensive analysis inside a real industrial workspace environment, in Sect. IV we study the learning loss and the convergence time over mesh networks characterized by varying (weakly to fully) connected structures and considering different decentralized FL strategies, namely gossip and consensus. The ultimate goal is to analyze how the quality of the information flow among the devices affect the learning loss and latency trade-off.

II. Decentralized FL and physical communications

The objective of FL [2] is to learn a global model $\hat{y}(\mathbf{W};\mathbf{x})$ that transforms input observations $\mathbf{x}$ into the desired outputs, *i.e.* $\hat{y} \in \{y_c\}_{c=1}^{C}$, with model parameters specified by the matrix $\mathbf{W}$. In what follows, and without leading in generality, we will focus specifically on NN models of $Q \geq 1$ layers and trained by SGD methods [3]. The model matrix $\mathbf{W} = \left[\mathbf{w}_1 \cdots \mathbf{w}_Q\right]$ collects all model parameters (*i.e.,* weights and biases) for each layer $\mathbf{w}_q = [w_{1,q}, ..., w_{d_q,q}]^T$, $q = 1, ..., Q$. Notice that layers can have different input/output dimensions (d_q) and the analysis can be extended accordingly.

In FL systems, all devices act as local learners: each k-th device has a database $\mathcal{E}_k$ of E_k (labeled) examples $(\mathbf{x}_h, y_h)$ that are used to train a local model $\mathbf{W}_{k,t}$ at some time t. Local examples $\mathcal{E}_k$ are typically collected independently and individually by the devices based on their local observations of the given phenomenon. Therefore, local samples are not representative of the full data distribution $\mathcal{E} = \bigcup_{k=1}^{K} \mathcal{E}_k$ of size $E = \sum_{k=1}^{K} E_k$. A practical example is given in Sect. IV. Federated optimization targets the iterative exchange of local models $\mathbf{W}_{k,t}$ with a PS server [9] for the cooperative training of the global model $\mathbf{W}$, such that $\forall k$ it is $\lim_{t\to\infty} \mathbf{W}_{k,t} = \mathbf{W}$. Optimization is generally applicable to any finite-sum objective/loss $L(\mathbf{W})$ of the form

$$\min_{\mathbf{W}} L(\mathbf{W}) = \min_{\mathbf{W}} \underbrace{\sum_{k=1}^{K} E_k \times L_k(\mathbf{W})}_{L(\mathbf{W})}, \tag{1}$$

where $L_k(\mathbf{W})$ is the local loss, or cost, of the k-th device, such that $\mathbf{W}_{k,t} = \underset{\mathbf{W}}{\operatorname{argmin}}\, L_k(\mathbf{W})$.

A. Gossip and consensus approaches to decentralized FL

In decentralized FL, the learning process alternates a local model update running inside the individual devices and a communication round, where the devices diffuse the model updates $\mathbf{W}_{k,t}$ over across all the network nodes [11] and following an assigned connectivity graph. In what follows, the network is modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \xi)$ with K devices (nodes or vertexes) $\mathcal{V} = \{1, 2, ..., K\}$ and edges (links) ξ. We define the neighbor set of device k as $\mathcal{N}_{k,t}$: it contains the chosen neighbors for cooperative training at round t, such that $(k, i) \in \xi, \forall i \in \mathcal{N}_{k,t}$. For this chosen neighbor set, the graph $\mathcal{G}$ has adjacency matrix $\mathbf{A} = [\sigma_{k,i}]$ where $\sigma_{k,i} > 0$ iff

$i \in \mathcal{N}_{k,t}$ and $\sigma_{k,i} = 0$ otherwise. Decentralized FL replaces the centralized fusion of model parameters implemented by the PS server with consensus [6]. Every new communication round ($t > 0$) a device k performs a *model averaging,* $\mathbf{W}_{k,t}^{(a)}$, by combining the local model $\mathbf{W}_{k,t}$ with the model update(s) $\mathbf{W}_{i,t}$ obtained from the neighbor device(s), followed by a *model adaptation*, running SGD on local training batches of data. The model update equation is written as

$$\begin{aligned}\mathbf{W}_{k,t+1} = \underbrace{\mathbf{W}_{k,t} + \epsilon_t \sum_{i \in \mathcal{N}_{k,t}} \sigma_{k,i} \left(\mathbf{W}_{i,t} - \mathbf{W}_{k,t}\right)}_{\mathbf{W}_{k,t}^{(a)}} + \\ - \mu_t \nabla \mathbf{L}_{k,t}(\mathbf{W}_{k,t}^{(a)}) \,,\end{aligned} \tag{2}$$

where ϵ_t is the *consensus step-size* while the terms $\sigma_{k,i} = E_i \,/ \left(\sum_{i \in \mathcal{N}_{k,t}} E_i\right)$ are the mixing weights. Finally, $\nabla \mathbf{L}_{k,t}(\mathbf{W}_{k,t}^{(a)}) = \nabla_{\mathbf{W}_{k,t}^{(a)}} \left[L_k(\mathbf{W}_{k,t}^{(a)})\right]$ represents the gradient of the local loss L_k observed by the k-th device w.r.t. the aggregated model $\mathbf{W}_{k,t}^{(a)}$. The SGD step size μ_t can be tuned on consecutive epochs t to improve convergence [3]. The consensus step-size ϵ_t is chosen as $\epsilon_t \in (0, 1/\Delta)$, where $\Delta = \max_k \left(\sum_{i \in \mathcal{N}_{k,t}} 1_{\sigma_{k,i} > 0}\right)$, namely the maximum degree of the graph $\mathcal{G}$ [12], being $1_{[\cdot]}$ the indicator function. Gossipgrad, or sum-weight gossip, [5] uses $\epsilon_t = \sum_{i \in \mathcal{N}_{k,t}} E_i \,/\, E$, and $\forall k, t$ a *single neighbor*, $|\mathcal{N}_{k,t}| = 1$, chosen randomly on every new communication round. Generalizing gossip, the consensus model update equation (2) exploits the cooperation with more neighbors to improve model averaging [6] on each round, at the cost of additional network resources. This trade-off is explored in Sect. IV, where a consensus update with $|\mathcal{N}_{k,t}| \leq 2$, (up to 2 neighbors chosen randomly on each round) is compared with gossip FL.

The model updates are compressed as $\widetilde{\mathbf{W}}_{k,t} = \mathcal{P}_\varrho \left[\mathbf{W}_{k,t}\right]$ by the non-linear operator $\mathcal{P}_\varrho[\cdot]$ before being forwarded to the neighbors for a new consensus round. Finally, a stopping criteria, *e.g.* on learning loss or convergence time, is then applied to end the diffusion of updates after some rounds.

B. Model pruning and communication of model parameters

Decentralized FL requires an intensive use of D2D communications for the synchronous diffusion of the model parameters. In particular, in what follows, we assume that the devices forward the model updates to the neighbors satisfying half-duplex constraints. They thus multiplex a digital representation of the model parameters into one (or more) frame(s) of B bits and transmit such frame(s) using an orthogonal, or interference free, channel of bandwidth $\mathcal{W}$ assigned by the MAC layer scheduler (described in Sect. III). Any wireless link between a pair of devices (k, i) at distance $d_{k,i}$ is impaired, at the physical layer, by frequency-flat fading with baseband complex-valued channel gain $h_{k,i} \sim \mathcal{CN}(0, 1)$, and average power $\mathrm{E}\left[|h_{k,i}|^2\right] = \vartheta_0 \left(\frac{d_0}{d_{k,i}}\right)^{\upsilon}$ that accounts for shadowing ϑ_0, path loss index υ, and the path loss terms $d_{k,i}^{-\upsilon}$ at reference distance d_0, respectively. Considering the device transmission and noise power P_T and N_0, respectively, the instantaneous Signal to Noise Ratio (SNR) for the link (k, i)

$$\gamma_{k,i} = \frac{P_T \vartheta_0}{N_0} \left(\frac{d_0}{d_{k,i}}\right)^{\upsilon} |h_{k,i}|^2 \tag{3}$$

can be used to assess the spectral efficiency of the link $B < \mathcal{W} \times \log(1 + \gamma_{k,i})$ as well as the quality of the communication channel. In particular, the link (k, i) is assigned as potential edge $(k, i) \in \xi$ of the graph $\mathcal{G}$ if $\gamma_{k,i} > \beta$ where β is the receiver-side sensitivity threshold [19].

Considering the IEEE 802.15.4e physical layer, for links whose SNR is above the sensitivity threshold $\beta = -90$ dBm, the standard supports frames of up to 127 bytes ($\frac{B}{\mathcal{W}} = 0.08$ bit/s/Hz), and Transmission Time Intervals (TTI) equal to $\mathrm{TTI} = 4$ ms, leaving about $B = 100$ bytes [18] as payload to be used by FL. Considering such small frame size, pruning and quantization of model parameters is fundamental. Typical NN models, targets of this paper, might contain > 20K parameters per layer, often extremely sparse. Effective solutions to limit the communication overhead are sparsification and distillation [13]. Model quantization in SGD algorithms has been addressed in [14] and, more recently, in [15] for application to FL.

Here, the proposed framework implements a compress and forward policy. Compression of model parameters combines a model pruning stage and deterministic quantization. Model pruning $\widetilde{\mathbf{W}}_{k,t} = \mathcal{P}_\varrho[\mathbf{W}_{k,t}]$ drops out all non-informative model parameters (*i.e.,* weights, biases or NN units) and uses the (configurable) sparsity constraint ϱ, such that $\forall w_{q,d} \in \{\mathbf{W}_{k,t}\}$ that satisfy $w_{q,d} < \varrho$, $w_{q,d}$ are set to zero. Threshold ϱ sets the compression rate and the number of ML model parameters to be digitalized. The resulting sparsified model $\widetilde{\mathbf{W}}_{k,t}$ is quantized to $b_{k,t} = b_{k,t}(\varrho)$ bits. The quantization scheme encodes the position of the non-zero model parameters and assigns a fixed number of bits (here 16 bits) to each parameter [7]. Notice that device k sending the model updates encoded by $b_{k,t}$ bits, and using frame payloads of size B, requires the multiplexing of $\left\lceil \frac{b_{k,t}}{B} \right\rceil$ frames. Model pruning and quantization therefore affect both the learning loss and the time span of each communication round.

III. Networking framework for decentralized FL

A communication round for FL, *i.e.* from time t to $t + 1$, requires the exchange of all the model parameter updates (2) using the links

$$L_t = \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_{k,t}} 1_{\sigma_{k,i} > 0} \tag{4}$$

that are chosen at time t to contribute to the cooperative training session. The duration, or time span, of each round depends on the MAC policy of the underlying wireless network. Diffusion of model parameters is rooted here at the TSCH MAC layer of the IEEE 802.15.4e [17]. Notice that the proposed approach is general enough for application to both existing and future emerging industrial wireless standards,

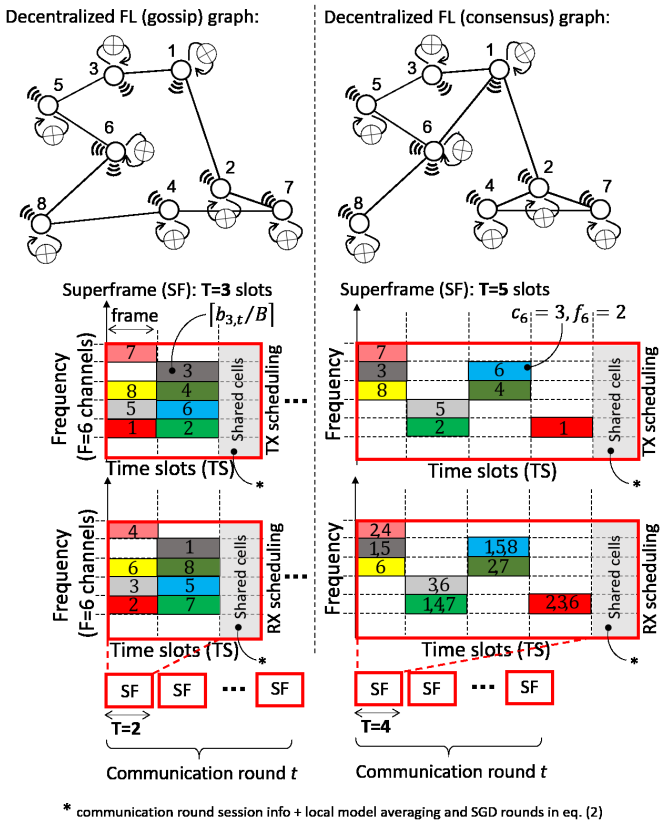

Fig. 2. TSCH MAC layer resource scheduling examples for a communication round of decentralized FL. From top to bottom: network graph, TX and RX scheduling maps. Left column: gossip FL using $\forall k, t$ one neighbor, *i.e.* $|\mathcal{N}_{k,t}| = 1$ per node, selected randomly on each round. Right column: consensus FL using $\forall k, t$ up to 3 neighbors, *i.e.* $|\mathcal{N}_{k,t}| \leq 3$, on each round. Active links, slots c_i and frequency f_i offsets, are highlighted in both cases.

all defined on top of the TSCH MAC, such as IETF IPv6 (6TiSCH) [18], WirelessHART and ISA100.11a standards [16] and recent amendments. Diffusion of model parameter updates (2) is regulated by a time-frequency scheduler that complies with the channel hopping capabilities of the radio devices and assigns transmission resources based on an (input) graph $\mathcal{G}$. Notice that scheduler runs on the Host, or server, station that acts as network manager and is in charge of distributing the synchronization information [19].

A. TSCH MAC and communication round scheduling

The time slotted channel hopping mode was originally addressed in the 2015 revision of the IEEE 802.15.4 standard. TSCH splits time in multiple Time Slots (TS) and Super-Frames (SFs) that repeat over time. For each TS, a set of F frequencies (IEEE 802.15.4 standard orthogonal channels) can be assigned for simultaneous transmission of multiple devices. A SF generally consists of a number T of consecutive TS and identifies a transmission session where slots and frequencies can be reserved to individual devices, or shared to broadcast control, synchronization or transmission session information. For each SF, a time-frequency scheduler assigns a transmission resource, namely one (or more) TS(s) and frequenc(ies), or cells, to an individual device i to communicate with its neighbor(s) according to the graph $\mathcal{G}$. Each cell is identified by the slot offset $c_i \in 1, ..., T$ and the channel offset $f_i \in 1, ..., F$ coordinates, respectively.

As depicted in the Figure 2, a decentralized FL communication round spans a number of consecutive SFs, or time slots. In this example, the model parameters exchange runs on top of TSCH MAC and considers $K = 15$ networked field devices implementing gossip and consensus FL in (2). In particular, gossip FL (on the left) uses $\forall k, t$ one neighbor, or parent, per node *i.e.* $|\mathcal{N}_{k,t}| = 1$), that may change every new communication round. Consensus based FL (on the right) performs model averaging from up to 3 neighbors, *i.e.* $|\mathcal{N}_{k,t}| \leq 3$, varying on each round (if available). For an assigned graph $\mathcal{G}$ and a specified number of channels (here $F = 6$), the scheduler optimizes the number of TS (T) in each SF to provide a slot c_i and a channel f_i offset to each device i. In particular, the scheduler should guarantee that: *i)* no couple of connected devices are scheduled to transmit in the same cell (c_i,f_i) resource; *ii)* no devices can be scheduled to simultaneously transmit and receive using the same slot c_i offset (*i.e.,* device are equipped with a single radio receiver chain); *iii)* no couple of devices sending model updates to the same device are scheduled to transmit using the same slot t_i offset. It can be easily shown that the number of slots T scales with the network degree Δ and on the number of links L_t in (4). For the considered example, $T = 3$ slots are required by gossip to visit all the devices, while $T = 5$ slots are required by consensus. For k-regular networks (*i.e.,* all network devices have the same number of neighbors, or degree Δ), the total number of slots required at round t is bounded by

$$T = T(\triangle) > \frac{L_t}{\min[F, \frac{K}{\triangle+1}]} \tag{5}$$

where we highlighted the dependency of T on the network degree Δ and the active links L_t. Notice that gossip FL uses $\Delta = 1$, $L_t = K$, and forces the network to be k-regular, therefore $T > \frac{K}{\min[F, K/2]}$. Practical solutions to the TSCH scheduling problem considering more complex graph structures can adopt graph coloring schemes [7] or ad hoc approaches [20]. In what follows, we use the Minimal Scheduling Function (MSF), recently proposed as standard [21].

Based on the example of Fig. 2, every FL communication round consists in general of $\left\lceil \frac{\max_k b_{k,t}(\varrho)}{B} \right\rceil$ SFs of $T = T(\triangle)$ time slots each. The sparsity constraint ϱ rules the network load, namely the number of bits $b_{k,t} = b_{k,t}(\varrho)$ produced by the individual devices on each communication round, and thus the duration of the round. To keep track of all FL round information, additional shared time slots are used in each SF to broadcast specific information about the FL session such as: *i)* ML model training information, *e.g.* the NN model structure to be trained in the specific round and the FL stopping criteria (here a max. number of rounds); *ii)* the number of SFs for the completion of the FL communication round; *iii)* the link

related information, *i.e.* the active neighbor set and the offset assignments, notifications of packet losses (ACK/NACK) and re-transmissions (that would delay the whole FL round). Shared channels are reserved to the Host station that is in charge of the supervision of the learning process. Notice that, although decentralized FL does not require a PS server, it is assumed here a centralized coordination of the learning rounds.

B. FL and network simulation

The decentralized FL (2) tool is written in Python and uses the TensorFlow library. The virtual environment creates an arbitrary number of devices, each configured to process an assigned training dataset and exchanging model parameters $\mathbf{W}_{k,t}$ that are pruned and quantized before forwarding using as described in Sect. II. Transmission time intervals and TSCH MAC access are simulated on a Matlab environment integrated in Python while all the exchanged parameters are saved in real-time on temporary cache files. The FL software takes as input an arbitrary (user-defined) graph $\mathcal{G}$ and a maximum number of neighbors to choose for averaging $\triangle$. It allows also to configure the consensus ϵ_t, the SGD step size μ_t and the federated data distribution E_k.

Once both network structure and learning parameters have been assigned, the framework runs a real-time simulation of the decentralized FL for a configurable number of communication rounds. Each round consists of a number of SFs used by devices to exchange model parameters. For the considered ML model training scenario (see the Sect. IV), the computational time required to issue a model update, mostly due to SGD, is in the order of few seconds, and it is negligible compared to the communication overhead. The time span of each FL communication round is thus quantified based on both TSCH constraints and scheduler decisions. The simulator outputs are: *i)* the FL learning loss for each round and device; *ii)* the number of TS, and the SFs, that were necessary to diffuse the model parameters on each communication round; *iii)* the computational time for local SGD on mini-batches.

IV. A CASE STUDY INSIDE AN INDUSTRIAL ENVIRONMENT

Decentralized FL is attracting interest in autonomous industrial systems where the centralized server orchestration is typically uncommon. A relevant example is in the field of collaborative robotics where human-robot co-presence risk mitigation is obtained by accurate Human-Robot (HR) distance monitoring. The example in Fig. 3 shows an assembly line consisting of interconnected industrial manipulators. The goal is to learn a ML model for the detection (classification) of the position of the human operators sharing the workspace, namely the HR distance d and the Direction Of Arrival (DOA) θ. Model learning is supervised while labeled data can be obtained independently by each device. The operator location information (d, θ) is used as input to the robot local control loop. The robot can re-plan its activity, stop or lower its speed, if the HR distance is smaller than a protective separation distance. Although the robots have direct connection with a Programmable Logic Controller (PLC) server, the direct link must be typically reserved for the robot control loop. This motivates the use of a decentralized FL policy.

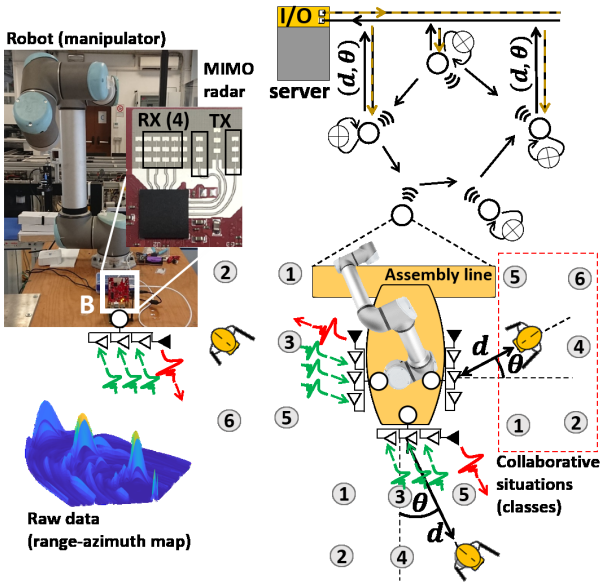

Fig. 3. Human-robot work-space (assembly line) environment, decentralized FL setup and human-robot distance (d) and direction of arrival - azimuth - (θ) classification over $C = 6$ different situations.

NN model	Layer #1 (trainable): 4 2D conv. (3×3 taps) Layer #2 (non trainable): ReLu, AvgPool (5, 5) Layer #3 (trainable): FC (4096×6) Layer #4 (non trainable): Softmax
PHY layer	$B = 100$ bytes, $\mathcal{W}$ =3 MHz TTI = 4 ms (time slot duration 10 ms) $P_T := [0 \div 5$ dBm] $\beta = -90$ dBm
TSCH	$F = 16$, T (optimized) $K = 15$ 20 slots: max SF duration

TABLE I
NN MODELS AND TRAINABLE PARAMETERS $\mathbf{W}$ FOR 6 CLASSES; PHY IEEE 802.15.4E AND TSCH MAC LAYER CONFIGURATIONS.

A. Federated data and ML model

The robotic manipulators are equipped with 3 Multiple-Input-Multiple-Output (MIMO) Frequency Modulated Continuous Wave (FMCW) radars, namely A, B, C, working in the 77 GHz band. Radars implement a Time-Division (TD) MIMO system with 2 transmit and 4 receive antennas each, and a field-of-view of 120 deg. A direct link connects the manipulators with a PLC server reserved for emergency robot stop. During the on-line workflow, the distance d and DOA θ information are classified independently by individual robots using a trained ML model and then sent to the PLC server via direct URLLC links for safety control. The ML model is here trained to classify $C = 6$ potential HR collaborative situations characterized by different HR distances and DOA ranges, corresponding to safe or unsafe conditions. These are detailed in the Fig. 3. The implemented ML model takes as input the raw range-azimuth data (after background subtraction) of size 256×63 from the radars. It consists of 2 trainable neural

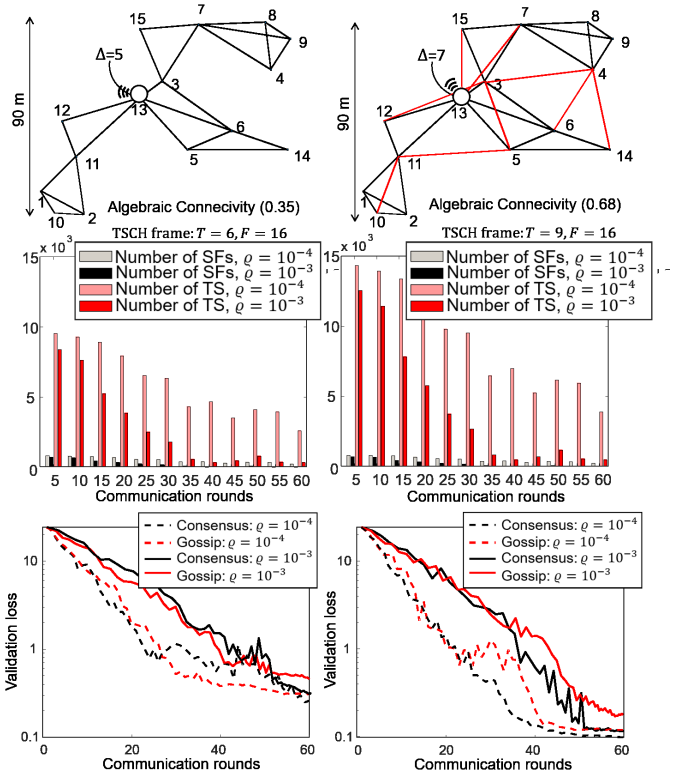

Fig. 4. Gossip FL $|\mathcal{N}_{k,t}| = 1$ vs. consensus implementing model averaging over $|\mathcal{N}_{k,t}| \leq 2$ neighbors. Effect of different sparsity constraints, namely $\varrho = 10^{-4}$ and $\varrho = 10^{-3}$, on the (total) number of TS, or the number of SF, considering two network graphs $\mathcal{G}$ with AC 0.35 and 0.68, respectively (additional links are highlighted in red). Bottom: FL validation loss versus communication rounds for both network graphs, using gossip (red), consensus (dark) and both sparsity constraints (dashed and solid lines), respectively.

Rounds	Duration	Loss	Decentralized FL
$1 \div 15$	84 s.	$12 \div 16$	Gossip ($\varrho = 10^{-3}$)
	95 s.	$7 \div 14$	Gossip ($\varrho = 10^{-4}$)
	126 s.	$13 \div 16$	Consensus ($\varrho = 10^{-3}$)
	143 s.	$6 \div 14$	Consensus ($\varrho = 10^{-4}$)
$20 \div 35$	18 s.	$5 \div 12$	Gossip ($\varrho = 10^{-3}$)
	63 s.	$0.8 \div 3$	Gossip ($\varrho = 10^{-4}$)
	27 s.	$4 \div 10$	Consensus ($\varrho = 10^{-3}$)
	95 s.	$0.25 \div 0.7$	Consensus ($\varrho = 10^{-4}$)
$40 \div 60$	8 s.	$0.15 \div 0.35$	Gossip ($\varrho = 10^{-3}$)
	41 s.	$0.1 \div 0.2$	Gossip ($\varrho = 10^{-4}$)
	12 s.	$0.1 \div 0.2$	Consensus ($\varrho = 10^{-3}$)
	53 s.	$0.07 \div 0.13$	Consensus ($\varrho = 10^{-4}$)

TABLE II
COMMUNICATION ROUND TIME SPANS (IN SECONDS) AND FL LOSS FOR GOSSIP AND CONSENSUS OVER A NETWORK WITH AC $= 0.68$ AND VARYING ϱ. PHY AND MAC PARAMETERS ARE IN TABLE I.

network layers of 25.000 parameters: details are given in Table I. D2D communications are regulated by the TSCH MAC and IEEE 802.15.4e physical layers with parameters listed in the same table. Decentralized FL uses model averaging (2) with consensus $\epsilon_t = 1/\triangle$ and SGD step size $\mu_t = 0.025$. A simplified database for testing, and some Python scripts as well, are available in the repository [22].

B. Communication overhead in TSCH networks

Learning loss of *gossip* using $|\mathcal{N}_{k,t}| = 1$ (red lines) and *consensus* using $|\mathcal{N}_{k,t}| \leq 2$ neighbors (dark lines) are compared in Fig. 4 over two networks of $K = 15$ robots and characterized by increasing number of connected components (from left to right), measured by Algebraic Connectivity (AC) [23]. For all cases, the loss is computed for 60 consecutive communication rounds, over a validation dataset and averaged over all the $K = 15$ devices. Local data is Independent and Identically Distributed (IID) among the robots using 2% of the full training dataset $\mathcal{E}$. The quality of the information flow is measured by AC, being the second smallest eigenvalue of the Laplacian of graph $\mathcal{G}$ [23]. Small AC, *i.e.* AC $= 0.35$ in the example on the left, indicates a weakly connected graph, a larger AC value, *i.e.* AC $= 0.68$ on the right, indicates the presence of a larger number of connected components. Large AC allows to reduce the learning loss as increasing the population of links that can be chosen for gossip/consensus operations. However, such benefit comes at the price of a more intensive use of the radio resources, that results in a increase of the time span of each communication round.

With respect to a TSCH MAC implementation, gossip FL uses a (sub)graph characterized by $\Delta = 1$, and $L_t = K$: it requires $T = 12$ slots per SF, of which 9 are reserved for model parameters exchange and 3 for shared cells (to exchange information about FL stages). Consensus implements model averaging over up to 2 neighbors and needs more cells, namely $T = 18$ slots per SF, of which 15 are used for parameters diffusion and 4 for shared cells. For the two cases, in the same Fig. 5, we highlighted the number of SFs, or equivalently, the *total* number of TS, that are necessary to diffuse the model parameters on each communication round. First rounds are more critical as they need more transmission resources compared to the last ones. In fact, during FL training, major changes of ML model parameters happen during initial rounds, while most of them stabilize after some rounds, thanks to model averaging. Two different sparsity constraints for model pruning are compared, namely $\varrho = 10^{-4}$ (dashed lines) that drops out the 20% of the parameters $\mathbf{W}_{k,t}$, on average, and $\varrho = 10^{-3}$ (solid lines) that drops the 30% of the parameters.

Considering the above choices, and assuming a TS duration of 10 ms, the duration of the initial (rounds $1 \div 15$), intermediate (rounds $20 \div 35$) and final (rounds $40 \div 60$) communication rounds are reported in Table II as well as the corresponding average FL losses. The sparsity constraint can be increased, *e.g.* from $\varrho = 10^{-4}$ to $\varrho = 10^{-3}$, to limit the round duration in exchange for a larger loss. For the considered case study, the use of gossip with a sparsity constraint of $\varrho = 10^{-3}$ is effective for the weakly connected graph characterized by AC $= 0.35$. For networks with larger AC, *i.e.* AC $= 0.68$, decentralized FL should implement a model averaging with more neighbors to leverage the additional links. Consensus obtains a loss of 0.1, corresponding to an human-robot distance estimation accuracy of 95%. The use of a model pruning with $\varrho = 10^{-4}$ is a reasonable choice, even if increasing the number of SFs per

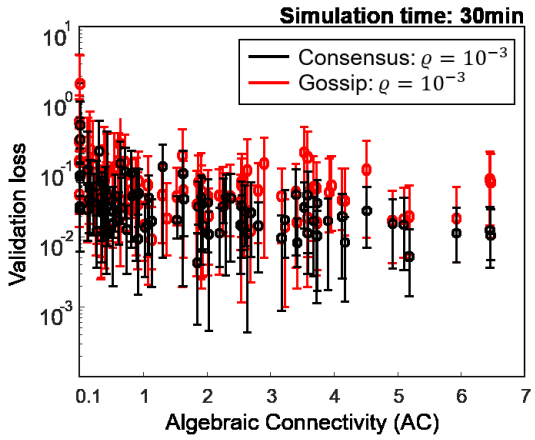

Fig. 5. Analysis of decentralized FL validation loss (max and min values) for varying network structures measured by Algebraic Connectivity (AC).

round. Although the paper focuses on a specific case study, we found that gossip FL is less sensitive to model pruning choices and it is a practical choice for the implementation of decentralized FL on weakly connected networks, characterized by few neighbors per device. Consensus is more sensitive to model pruning choices while it generally converges faster, compared to gossip, over strongly connected networks.

C. Impact of graph structure on convergence

In Fig. 5, decentralized FL loss is analyzed for varying network structures characterized by AC ranging from 0.1 to 6.5. Error bars for each network structure account for maximum and minimum loss values across $K = 15$ devices. The analysis considers 90 random graphs featuring different connectivity structures and it is relevant to highlight useful scaling laws on convergence. As done previously, we compare the learning loss of gossip $|\mathcal{N}_{k,t}| = 1$ and consensus FL $|\mathcal{N}_{k,t}| \leq 2$ observed after 30 min of simulation time in both cases. The observed loss is comparable, typically for $\mathrm{AC} \leq 0.4$. Consensus using multiple neighbors reveals to be more effective when strongly connected components emerge in the network, *i.e.* for $\mathrm{AC} \geq 1$. Deploying networks featuring large AC is always beneficial in terms of learning loss as expected; however, improvements are marginal when considering networks with large components ($\mathrm{AC} \geq 4$). It is expected that increasing the number of neighbors to be considered for consensus on each round would provide further benefits, at the price of an increased convergence time.

V. Conclusions

The paper introduced a joint learning and network simulation framework for the implementation of decentralized federated learning (FL) policies on top of a TSCH compliant wireless interface rooted at the IEEE 802.15.4e industrial standard. An extensive analysis inside a real industrial workspace environment is proposed to verify the proposed tool. The analysis focused in particular on how ML model parameter quantization, in-network processing, and connectivity affect the FL loss and the latency trade-off. Although not considered in this paper, the number of devices (K), the distribution of the federated training data (E_k), and the ML model complexity should be also accounted for to infer more general conclusions. These are left for future work.

References

[1] Konečný J., et al. "Federated optimization: Distributed machine learning for on-device intelligence," CoRR, 2016. [Online]. Available: http://arxiv.org/abs/ 1610.02527.

[2] P. Kairouz, et al., "Advances and open problems in federated learning," [Online]. Available: https://arxiv.org/abs/1912.04977.

[3] S. S. Ram, et al. "Distributed stochastic subgradient projection algorithms for convex optimization," Journal of optimization theory and applications, vol. 147, no. 3, pp. 516-545, Dec. 2010.

[4] A. Ghosh, A. Maeder, M. Baker and D. Chandramouli, "5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15," in IEEE Access, vol. 7, pp. 127639-127651, 2019.

[5] J. A Daily, et al., "Gossipgrad: Scalable deep learning using gossip communication based asynchronous gradient descent," [Online]. Available: https://arxiv.org/abs/1803.05880.

[6] S. Savazzi, et al. "Federated Learning with Cooperating Devices: A Consensus Approach for Massive IoT Networks," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4641-4654, May 2020.

[7] H. Xing, et al., "Decentralized Federated Learning via SGD over Wireless D2D Networks," Proc. IEEE 21st Int. Workshop on Signal Processing Advances in Wireless Comm. (SPAWC), 2020.

[8] M. Chen, et al. "Performance Optimization of Federated Learning over Wireless Networks," Proc. IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, pp. 1-6, 2019.

[9] M. Bennis, et al. "Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," Proc. of the IEEE, vol. 106, no. 10, pp. 1834-1853, Oct. 2018.

[10] M. M. Amiri and D. Gunduz, "Federated learning over wireless fading channels," IEEE Trans. Wireless Commun., 2020.

[11] M. Blot, et al., "Gossip training for deep learning," 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 2016. [Online]. Available: https://arxiv.org/abs/1611.09726.

[12] G. Soatti, et al. "Consensus-Based Algorithms for Distributed Network-State Estimation and Localization," IEEE Trans. on Signal and Information Processing over Networks, vol. 3, no. 2, pp. 430-444, June 2017.

[13] A. Elgabli, et al., "GADMM: Fast and Communication Efficient Framework for Distributed Machine Learning," Journal of Machine Learning Research, 2020.

[14] D. Alistarh, et al., "Qsgd: Communication-efficient sgd via gradient quantization and encoding," NIPS, 2017, pp. 1709–1720.

[15] N. Shlezinger, et al. "Federated Learning with Quantization Constraints," Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8851-8855, Barcelona, Spain, 2020.

[16] Standard IEC 62591:2010, Industrial communication networks - Wireless communication network and communication profiles - WirelessHART™, Edition 2.0, 06 Nov. 2015.

[17] H. Kurunathan, et al., "IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1989-2010, thirdquarter 2018.

[18] X. Vilajosana, et al., "IETF 6TiSCH: A Tutorial," IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 595-615, Firstquarter 2020

[19] L. Ascorti, et al., "A Wireless Cloud Network Platform for Industrial Process Automation: Critical Data Publishing and Distributed Sensing," IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 4, pp. 592-603, April 2017.

[20] S. Duquennoy, et al., "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in Proc. Conf. Embedded Netw. Sensor Syst. (SenSys), Seoul, South Korea, 2015, pp. 337–350.

[21] T. Chang, M. et al., "6TiSCH minimal scheduling function (MSF)," Internet Eng. Task Force, draft-ietf-6tisch-msf-06, Aug. 2019.

[22] S. Savazzi, "Federated Learning: mmWave MIMO radar dataset for testing, " IEEE Dataport, 2020. [Online]. Available: http://dx.doi.org/10.21227/0wmc-hq36. Accessed: May. 18, 2020.

[23] N. M. Maia de Abreu, "Old and new results on algebraic connectivity of graphs," Elsevier Linear Algebra and its Applications, vol. 423, no. 1, pp. 53-73, May 2007.