Asynchronous Decentralized Learning over Unreliable Wireless Networks

Eunjeong Jeong*, Matteo Zecchin*, and Marios Kountouris
Communication Systems Department
EURECOM, Sophia Antipolis, France
{eunjeong.jeong, matteo.zecchin, marios.kountouris}@eurecom.fr

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838891

***Abstract*—Decentralized learning enables edge users to collaboratively train models by exchanging information via device-to-device communication, yet prior works have been limited to wireless networks with fixed topologies and reliable workers. In this work, we propose an asynchronous decentralized stochastic gradient descent (DSGD) algorithm, which is robust to the inherent computation and communication failures occurring at the wireless network edge. We theoretically analyze its performance and establish a non-asymptotic convergence guarantee. Experimental results corroborate our analysis, demonstrating the benefits of asynchronicity and outdated gradient information reuse in decentralized learning over unreliable wireless networks.**

***Index Terms*—asynchronous decentralized learning, over-the-air computation, device-to-device communication.**

I. Introduction

Distributed learning algorithms empower devices in wireless networks to collaboratively optimize the model parameters by alternating between local optimization and communication phases. Leveraging the aggregated computational power available at the wireless network edge in a communication efficient [1] and privacy preserving manner [2], distributed learning is considered to be a key technology enabler for future intelligent networks. A promising paradigm, which enables collaborative learning among edge devices communicating in a peer-to-peer (server-less) manner, is decentralized learning [3]. Differently from federated learning, decentralized algorithms do not require a star topology with a central parameter server, thus being more flexible with respect to the underlying connectivity [4]. This feature renders decentralized learning particularly appealing for future wireless networks with device-to-device communication. Several decentralized learning schemes over wireless networks have been proposed and analyzed [5]–[8], highlighting the key role of over-the-air computation (AirComp) [9] for low-latency training at the edge. Prior works have mainly considered wireless networks of reliable workers communicating in a fixed topology throughout the entire training procedure. Nevertheless, these assumptions are hardly met in practical systems, in which communication links can be intermittent or blocked, and devices may become temporarily unavailable due to computation impairments or energy saving reasons. Asynchronous distributed training has been shown to mitigate the effect of stragglers (slow workers) [10]–[12]. However, harnessing the potential benefits of asynchronism in decentralized learning over unreliable wireless networks remains elusive.

In this paper, we propose an asynchronous implementation of decentralized stochastic gradient descent (DSGD) as a means to address the inherent communication and computation impairments of heterogeneous wireless networks. In particular, we study decentralized learning over a wireless network with a random time-varying communication topology, comprising unreliable devices that can become stragglers at any point of the learning process. To account for communication impairments, we propose a consensus strategy based on time-varying mixing matrices determined by the instantaneous network state. At the same time, we design the learning rates at the edge devices in such a way so as to preserve the stationary point of the original network objective in spite of the devices' heterogeneous computational capabilities. Finally, we provide a non-asymptotic convergence guarantee for the proposed algorithm, demonstrating that decentralized learning is possible even when outdated information from slow devices is used to locally train the models. Experimental results confirm our analysis and show that reusing stale gradient information can speed up convergence of asynchronous DSGD.

II. System Model

We consider a network consisting of m wireless edge devices, in which each node i is endowed with a local loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ and local parameter estimate $\theta_i \in \mathbb{R}^d$. The network objective consists in minimizing the aggregate network loss subject to a consensus constraint

$$\begin{aligned} \underset{\theta_1,\dots,\theta_m}{\text{minimize}} \; & f(\theta_1,\dots,\theta_m) := \frac{1}{m}\sum_{i=1}^{m} f_i(\theta_i) \\ \text{s.t.} \quad & \theta_1 = \theta_2 = \cdots = \theta_m. \end{aligned} \tag{1}$$

This corresponds to the distributed empirical risk minimization problem whenever f_i is a loss term over a local dataset. In the following, $f(\theta)$ denotes the network objective $f(\theta_1,\dots,\theta_m)\big|_{\theta_1=\cdots=\theta_m=\theta}$ and $\bar{\theta} = 1/m\sum_{i=1}^{m}\theta_i$. To solve (1), we consider a DSGD algorithm according to which devices alternate between a local optimization based on gradient information (computation phase) and a communication phase.

*These authors contributed equally to this work.

The work of E. Jeong is supported from a Huawei France-funded Chair on Future Wireless Networks. The work of M. Zecchin is funded by the Marie Sklodowska Curie action WINDMILL (grant No. 813999).

A. Computation model

To locally optimize the model estimate θ_i, we assume that each device can query a stochastic oracle satisfying the following properties.

Assumption 1. *At each node i, the gradient oracle $g_i(\theta)$ satisfies the following properties for all $\theta \in \mathbb{R}^d$*

- $\mathbb{E}[g_i(\theta)] = \nabla_\theta f_i(\theta)$ *(unbiasedness)*
- $\mathbb{E}\,\|g_i(\theta) - \nabla_\theta f_i(\theta)\|^2 \leq \sigma^2$ *(bounded variance)*
- $\mathbb{E}\,\|g_i(\theta)\| \leq G^2$ *(bounded magnitude).*

We admit the existence of straggling nodes and that a random subset of devices can become inactive or postpone local optimization procedures, e.g., due to computation impairments or energy constraints. As a result, devices may join the communication phase and disseminate a model that has been updated using gradient information computed using previous model estimates, or a model that has not been updated at all from the previous iteration(s). Formally, at every optimization round t, the local update rule is

$$\theta_i^{(t+\frac{1}{2})} = \begin{cases} \theta_i^{(t)}, & \text{if device } i \text{ is straggler at round } t \\ \theta_i^{(t)} - \eta_i^t g_i(\theta^{(t-\tau_i)}), & \text{otherwise} \end{cases} \tag{2}$$

where η_i^t is a local learning rate and the delay $\tau_i \geq 0$ accounts for the staleness of the gradient information at device i.

B. Communication model

The channel between any pair of device i and j follows a Rayleigh fading model. At every communication iteration t, devices can exchange information according to a connectivity graph $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$, where $\mathcal{V} = \{1, 2, \ldots, m\}$ indices the network nodes and $(i, j) \in \mathcal{E}^{(t)}$ if devices i and j can communicate during round t. We consider symmetric communication links; therefore the communication graph is undirected. While the connectivity graph is assumed to remain fixed within the optimization iteration, it may vary across optimization iterations due to deep fading, blockage, and/or synchronization failures.

III. Asynchronous Decentralized SGD

The proposed asynchronous DSGD procedure, which takes into account both computation and communication failures, is detailed in Algorithm 1.

At the beginning of each training iteration t, non straggling devices update the local estimate $\theta_i^{(t)}$ according to (2) using a potentially outdated gradient information. Subsequently, based on the current connectivity graph $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$, devices agree on a symmetric and doubly stochastic mixing matrix $W^{(t)}$ using a Metropolis-Hastings weighting scheme [13]. The weights are very simple to compute and are amenable for distributed implementation. In particular, each device requires only knowledge of the degrees of its neighbors to determine the weights on its adjacent edges.

After that, it follows a communication phase in which devices exchange the updated estimates and employ a gossip scheme based on $W^{(t)}$. To leverage AirComp capabilities, devices employ analog transmission together with the scheduling scheme proposed in [7]. Accordingly, the communication phase is divided into multiple pairs of communication slots. Each pair consists of an *AirComp slot* and a *broadcast slot* as illustrated in Fig. 1. During the AirComp slot s, the star center i receives the superposition of the signals transmitted by its neighboring devices $\mathcal{N}^{(t)}(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}^{(t)}\}$. In particular, each scheduled node $j \in \mathcal{N}^{(t)}(i)$ transmits to

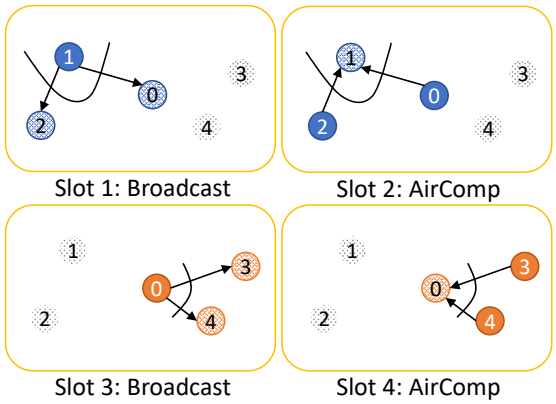

Fig. 1. An example of the timeline for one training iteration composed of alternate Broadcast and AirComp slots.

Algorithm 1 Asynchronous Decentralized SGD

Input: $\theta_i^{(0)} = \mathbf{0} \in \mathbb{R}^d$
Output: $\bar{\theta}^{(T)}$

for t in $[0, T]$ **do**
  for each non straggling devices **do**
    update local model as (2)
  end for
  Determine matrix $W^{(t)}$ based on $\mathcal{G}^{(t)}$
  for s in $[1, S_t]$ **do**
    if $s \equiv 0 \pmod 2$ **then**
      # Broadcast phase
      for each device i scheduled in slot s **do**
        Device i transmits (6)
        Each device $j \in \mathcal{N}_i^{(t)}$ receives (7)
        Each device $j \in \mathcal{N}_i^{(t)}$ estimates (8)
      end for
    else
      # AirComp Phase
      for each star center i scheduled in slot s **do**
        Each device $j \in \mathcal{N}_i^{(t)}$ transmits (6)
        Device i receives (4)
        Device i estimates (5)
      end for
    end if
  end for
  for each device **do**
    model consensus as in (9)
  end for
end for

the star center i

$$x_j^{(s,t)} = \frac{\sqrt{\gamma_i^{(s,t)}}}{h_{i,j}^{(s,t)}} w_{i,j}^{(t)} \theta_j^{(t+\frac{1}{2})} \tag{3}$$

where $h_{i,j}^{(s,t)} \in \mathbb{C}^d$ is the channel coefficient between user i and j during slot s, $\gamma_i^{(s,t)} \in \mathbb{R}$ is a power alignment coefficient, and $w_{i,j}^{(t)}$ is the (i,j) entry of the mixing matrix W. The star center i receives the aggregated signal

$$y_i^{(s,t)} = \sum_{j \in \mathcal{N}(i)} h_{i,j}^{(s,t)} x_j^{(s,t)} + z_i^{(s,t)} \tag{4}$$

where $z_i^{(s,t)} \sim \mathcal{N}(0, \sigma_w \mathbb{1}_d)$ is a noise vector, and estimates the aggregated model as

$$\hat{y}_i^{(s,t)} = \frac{y_i^{(s,t)}}{\sqrt{\gamma_i^{(s,t)}}} = \sum_{j \in \mathcal{N}(i)} w_{i,j}^{(t)} \theta_j^{(t+\frac{1}{2})} + \frac{z_i^{(s,t)}}{\sqrt{\gamma_i^{(s,t)}}}. \tag{5}$$

On the other hand, during a broadcast slot s, scheduled node i transmits using a power scaling factor $\alpha_i^{(s,t)}$ the signal

$$x_i^{(s,t)} = \sqrt{\alpha_i^{(s,t)}} \theta_i^{(t+\frac{1}{2})} \tag{6}$$

and all neighboring devices $j \in \mathcal{N}^{(t)}(i)$ receive

$$y_j^{(s,t)} = h_{j,i}^{(s,t)} x_i^{(s,t)} + z_j^{(s,t)} \tag{7}$$

and estimate the updated model as

$$\hat{y}_j^{(s,t)} = w_{j,i}^{(t)} \frac{y_j^{(s,t)}}{\sqrt{\alpha_i^{(s,t)}} h_{j,i}^{(s,t)}} = w_{j,i}^{(t)} \left(\theta_i^{(t+\frac{1}{2})} + \frac{z_j^{(s,t)}}{\sqrt{\alpha_i} h_{j,i}} \right). \tag{8}$$

At the end of the communication phase, each node i obtains the new estimate $\theta_i^{(t+1)}$ combining all received signals and using a consensus with step size $\zeta \in (0,1]$

$$\theta_i^{(t+1)} = (1-\zeta)\theta_i^{(t+\frac{1}{2})} + \zeta \left\{ \sum_{j=1}^{m} w_{i,j}^{(t)} \theta_j^{(t+\frac{1}{2})} + \tilde{n_i}^{(t)} \right\} \tag{9}$$

where $\tilde{n}_i^{(t)} \sim \mathcal{N}(0, \tilde{\sigma}_{w,i}^{(t)} \mathbb{1}_d)$ is a noise vector term that accounts for the aggregation of noise components during AirComp and broadcast transmissions at device i during communication phase t.

IV. Convergence Analysis

In this section, we study the effect of communication and computation failures on the asynchronous DGSD procedure and prove its convergence.

A. Effect of Communication Failures

Communication impairments amount for a random connectivity graph with an edge set that differs at each different optimization iteration. From an algorithmic perspective, random communication impairments result in DSGD with stochastic mixing matrices. A particular class of stochastic mixing matrices are those that satisfy the expected consensus property.

Definition 1 (Expected Consensus Rate [4])**.** *A random matrix $W \in \mathbb{R}^{m \times m}$ is said to satisfy the expected consensus with rate p if for any $X \in \mathbb{R}^{d \times m}$*

$$\mathbb{E}_W \left[\left\| WX - \bar{X} \right\|_F^2 \right] \leq (1-p) \left\| X - \bar{X} \right\|_F^2$$

where $\bar{X} = X \frac{\mathbf{1}\mathbf{1}^T}{m}$ and the expectation is w.r.t. the random matrix W.

Lemma 1. *If the event that the connectivity graph $\mathcal{G}^{(t)}$ is connected at round t has a probability $q > 0$ and the Metropolis-Hastings weighting is used to generated the mixing $W^{(t)}$, the expected consensus rate is satisfied with rate $p = q\delta > 0$, with δ being the expected consensus rate in case of a connected topology.*

Proof. See Appendix A. □

If the expected consensus is satisfied, it is then possible to establish a convergent behavior for the estimates generated by the proposed algorithm.

Lemma 2 (Consensus inequality)**.** *Under Assumption 1, after T iterations, decentralized SGD with a constant learning rate η and consensus step size ζ satisfies*

$$\sum_{i=1}^{m} \left\| \theta^{(T)} - \bar{\theta}^{(T)} \right\|_2 \leq \eta^2 \frac{12mG^2}{(p\zeta)^2} + \zeta \frac{2}{p} \sum_{i=1}^{m} \sigma_{w,i}^2$$

where $\sigma_{w,i}^2 = \max_{t=0}^{T} \mathbb{E} \left\| \tilde{n}_i^{(t)} \right\|^2$.

Proof. See Appendix B. □

Overall, communication failures amount to a reduced expected consensus rate compared to the scenario with perfect communication. At the same time, dropping users that are delayed and are unable to synchronize and perform AirComp, renders the communication protocol more flexible. For instance, in Fig. 2, we consider a network of nine nodes organized according to different topologies and show the evolution of the average spectral gap of the mixing matrix with Metropolis-Hastings weights, whenever devices not satisfying a certain delay constraint are dropped. As expected, stricter delay requirements result in sparser effective communication graphs and mixing matrices with smaller spectral gaps.

B. Effect of Computation Failures

Random computation impairments make the group of devices that effectively update the model parameter vary over time. To account for this in the analysis, we introduce a virtual learning rate that is zero in case of failed computation. Namely, the learning rate at device i during computation round t becomes

$$\tilde{\eta}_i^{(t)} = \begin{cases} 0, & \text{if } i \text{ is straggler at round } t \\ \eta_i^{(t)}, & \text{otherwise} \end{cases}$$

where $\eta_i^{(t)}$ is a specified learning rate value in case of successful computation. Furthermore, to ensure that the procedure converges to stationary points of the network objective even

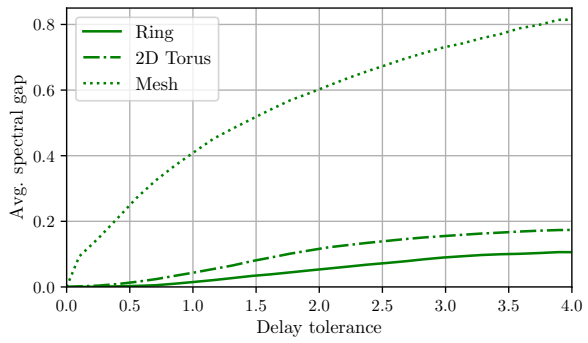

Fig. 2. Average spectral gap under different delay constraints for mesh, ring, and two-dimensional torus topologies with 9 nodes. Each link is associated to a completion time $\sim Exp(1)$ and is dropped if it exceeds the delay tolerance value.

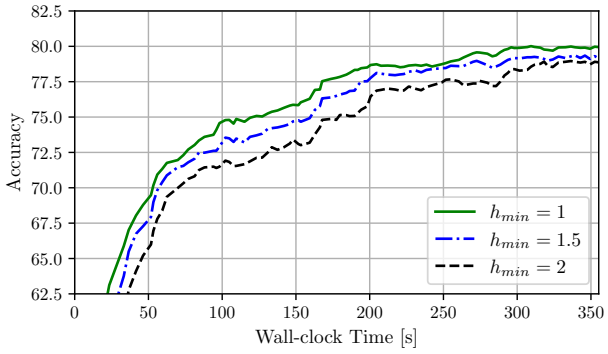

Fig. 3. Test accuracy versus time under different channel gain thresholds. Smaller thresholds result in larger average consensus rates and therefore in faster convergence.

when edge devices have different computing capabilities, the expected learning rates have to be equalized. In particular, if $\mathbb{E}[\eta_i^{(t)}] = \eta,\ \forall i$, we have that stationary points are maintained in expectation, namely

$$\sum_{i=1}^{m} \mathbb{E}[\tilde{\eta}_i^{(t)}] \nabla f_i(\theta) = 0 \implies \nabla f(\theta) = 0.$$

Finally, the existence of straggling devices introduces asynchronicity in the decentralized optimization procedure. In particular, a device i that fails at completing the gradient computation at a given optimization iteration is allowed to apply the result in a later one, without discarding the computation results. While we do not specify the delay distribution, we rather introduce the following assumption regarding the staleness of gradients.

Assumption 2. *For all iteration t, there exists a constant $\gamma \leq 1$ such that*

$$\mathbb{E}\left\| \nabla f(\bar{\theta}^{(t)}) - \frac{\sum_{i=1}^{m} \nabla f_i(\theta_i^{(t-\tau_i)})}{m} \right\|^2 \leq \gamma \mathbb{E}\left\| \nabla f(\bar{\theta}^{(t)}) \right\|^2 + L^2 \frac{\sum_{i=1}^{m} \mathbb{E}\left\| \theta_i^{(t)} - \bar{\theta}^{(t)} \right\|^2}{m}.$$

The above assumption is similar to the one in [10] with an additional consensus error term. Note that the value of γ is proportional to the staleness of the gradients and in case of perfect synchronization ($\gamma = 0$) the bound amounts to a standard consensus error term.

C. Convergence Guarantee

In this subsection, we demonstrate the convergence of the decentralized optimization procedure to a stationary point of the problem (1).

Theorem 1. *Consider a network of unreliable communicating devices in which the expected consensus rate is satisfied with constant p and each device can be a straggler with probability $\rho_i < 1$. If Assumptions 1 and 2 are satisfied, asynchronous DSGD with constant learning rate $\eta_i = \min_j(1-\rho_j)/(\sqrt{4LT}(1-\rho_i))$ and consensus rate $\zeta = 1/T^{3/8}$ satisfies the following stationary condition*

$$\frac{1}{T}\sum_{t=1}^{T} \left\| \nabla f(\bar{\theta}^{(t)}) \right\|^2 \leq \frac{8\sqrt{L}(f(\bar{\theta}^{(T)}) - f^*)}{\gamma' \rho_{min} \sqrt{T}} + \frac{3G^2 L}{T^{1/4} p^2 \gamma'} + \sqrt{\frac{L}{4T}} \frac{\sigma^2}{m\gamma' \min_j(1-\rho_j)} + \sum_{i=1}^{m} \frac{\sigma_{w,i}^2}{m\gamma'} \left(\frac{2L^2\gamma}{pT^{3/8}} + \frac{4L\sqrt{L}}{mT^{1/4}\rho_{min}} \right)$$

where $\gamma' = 1 - \gamma$, $\rho_{min} = \min_j(1-\rho_j)$ and $f^* = \min_{\theta \in \mathbb{R}^d} f(\theta)$.

Proof. See Appendix C. □

The above theorem establishes a vanishing bound on the stationarity of the returned solution, which involves quantities related to both communication and computation impairments. In particular, the constant of the slowest vanishing terms $T^{-1/4}$ contains the term p related to random connectivity, as well as γ' and ρ_{min} due to stragglers.

V. Numerical Results

The effectiveness of the proposed asynchronous DSGD scheme is assessed using a network of $m = 15$ devices that collaboratively optimize the parameters of a convolutional neural network (CNN) for image classification with Fashion-MNIST. Gradients are calculated using batches of 16 data samples and the performance is evaluated using a test set of 500 images. We model the channel gain between each device pair as Rayleigh fading and we assume a shifted exponential computation time at each device, i.e., $T_{comp} = T_{min} + Exp(\mu)$ with $T_{min} = 0.25$s and $\mu = 1$. In Fig. 3, nodes communicate only when the channel is in favorable conditions, i.e., when the channel gain exceeds a certain minimum threshold h_{min}. This allows to save energy; however, while higher threshold values result into lower average energy consumption, they also

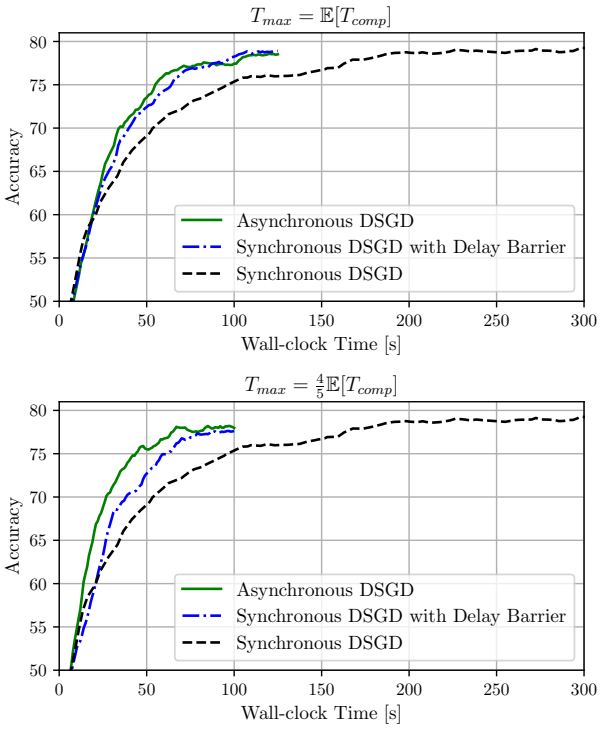

Fig. 4. Test accuracy for the asynchronous, synchronous with delay barrier, and synchronous schemes under two different values of T_{max}.

produce mixing matrices with smaller consensus rate, thus increasing the convergence time.

To study the effect of computation impairments, our proposed asynchronous learning algorithm is compared with: (i) *synchronous DSGD*, which waits for all devices to finish their computations; and (ii) *synchronous DSGD with a delay barrier* T_{max}, which discards computation from users that violate the maximum computing time. Compared to the latter, our asynchronous procedure allows for slow devices to reuse stale gradient computations during later iterations. In Fig.4, we plot the evolution of the test accuracy of the aforementioned algorithms under two different values of T_{max}. For a moderate delay constraint $T_{max} = \mathbb{E}[T_{comp}]$, asynchronous DSGD and synchronous DSGD with delay barrier perform similarly as the fraction of slow users is modest. Nonetheless, imposing a delay constraint and discarding slow devices greatly reduces the training time compared to the synchronous DSGD case. On the other hand, for a stringent delay requirement, $T_{max} = \frac{4}{5}\mathbb{E}[T_{comp}]$, reusing stale gradients turns out to be beneficial and the proposed asynchronous DSGD attains higher accuracy faster compared to the synchronous DSGD with a delay barrier.

VI. Conclusion

In this work, we have proposed and analyzed an asynchronous implementation of DSGD, which enables decentralized optimization over realistic wireless networks with unreliable communication and heterogeneous devices in terms of computation capabilities. We have studied the effect of both communication and computation failures on the training performance and proved non-asymptotic convergence guarantees for the proposed algorithm. The main takeaway is that reusing outdated gradient information from slow devices is beneficial in asynchronous decentralized learning.

Appendix

A. Proof of Lemma 1

Define the event $E^{(t)} := \{\mathcal{G}^{(t)} \text{ is connected}\}$ and its complementary event $\bar{E}^{(t)}$. Whenever the Metropolis-Hasting weights are obtained from a connected graph, the resulting mixing matrix $W^{(t)}$ has a consensus rate greater than zero. Therefore, there exists $\delta > 0$ such that

$$\mathbb{E}_{W^{(t)}|E^{(t)}} \left\| W^{(t)}X - \bar{X} \right\|_F^2 \leq (1-\delta) \left\| W^{(t)}X - \bar{X} \right\|_F^2$$

It follows that, for any $X \in \mathbb{R}^{d\times m}$

$$\begin{aligned}\mathbb{E}_{W^{(t)}} \left\| W^{(t)}X - \bar{X} \right\|_F^2 =& q\mathbb{E}_{W^{(t)}|E^{(t)}} \left\| W^{(t)}X - \bar{X} \right\|_F^2 \\ &+ (1-q)\mathbb{E}_{W^{(t)}|\bar{E}^{(t)}} \left\| X - \bar{X} \right\|_F^2 \\ \leq& q(1-\delta) \left\| W^{(t)}X - \bar{X} \right\|_F^2 \\ &+ (1-q) \left\| X - \bar{X} \right\|_F^2\end{aligned}$$

where we have lower bounded the consensus rate by zero in case of disconnected topologies. Grouping terms and having assumed $q > 0$, we obtain that the expected consensus is satisfied with rate $(1 - q\delta) > 0$.

B. Proof of Lemma 2

Similarly to [7], [14] we establish the following recursive inequality

$$\begin{aligned}\sum_{i=1}^m \mathbb{E} \left\| \theta^{(t)} - \bar{\theta}^{(t)} \right\|^2 \leq& \left(1 - \frac{p\zeta}{2}\right) \sum_{i=1}^m \mathbb{E} \left\| \theta^{(t-1)} - \bar{\theta}^{(t-1)} \right\|^2 \\ &+ \frac{\eta^2}{p\zeta} \left(6mG^2\right) + \zeta^2 \sum_{i=1}^m \mathbb{E} \left\| \tilde{n}_i^{(t)} \right\|^2.\end{aligned}$$

Defining $\sigma_{w,i}^2 = \max_{t=0}^T \mathbb{E} \left\| \tilde{n}_i^{(t)} \right\|^2$ and then solving the recursion we obtain the final expression.

C. Proof of Theorem 1

We denote stale gradients by $g_i(\tilde{\theta}_i^{(t)}) = g_i(\theta_i^{(t-\tau_i)})$. According to the update rule, at each iteration $t+1$, we have

$$\mathbb{E}[f(\bar{\theta}^{t+1})] = \mathbb{E}\left[f\left(\bar{\theta}^t - \frac{1}{m} \sum_{i=1}^m \left(\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)}) + \zeta \tilde{n}_i^{(t)} \right) \right) \right]$$

where the expectation is w.r.t. the stochastic gradients, the communication noise $\Xi^{(t)}$, and the computation and commu-

nication failures at iteration $t+1$. For an L-smooth objective function, we have

$$\mathbb{E}[f(\bar{\theta}^{(t+1)})] \leq f(\bar{\theta}^{(t)}) \underbrace{-\frac{1}{m}\sum_{i=1}^{m}\left\langle \nabla f(\bar{\theta}^{(t)}), \mathbb{E}[\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)})]\right\rangle}_{:=T_1} + \underbrace{\frac{L}{2m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)})\right\|^2}_{:=T_2} + \frac{L}{2m^2}\zeta^2\sum_{i=1}^{m}\mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2$$

where we used the fact that the communication noise has zero mean and is independent across users.

Adding and subtracting $\nabla f_i(\bar{\theta}^{(t)})$ to each summand of T_1 and since $\mathbb{E}[\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)})] = \eta \nabla f_i(\tilde{\theta}_i^{(t)})$, with $\eta = \min_j(1-\rho_j)/(\sqrt{4LT})$, we obtain

$$\begin{aligned}
T_1 =& -\eta\left\langle \nabla f(\bar{\theta}^{(t)}), \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\rangle \\
=& \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(t)}) - \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2 \\
&- \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 - \frac{\eta}{2m^2}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2 \\
\leq& \frac{\eta\gamma}{2}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 + \frac{\eta L^2}{2m}\sum_{i=1}^{m}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2 \\
&- \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 - \frac{\eta}{2m^2}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2
\end{aligned}$$

where we have used the staleness assumption. The last term can be bounded using the property of the stochastic gradient and the fact that $\tilde{\eta}_i^{(t)} \leq 1/(\sqrt{4LT}) \leq 1/(\sqrt{4L})$ as

$$\begin{aligned}
T_2 \leq& \frac{L}{2m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\tilde{\eta}_i^{(t)}[g_i(\tilde{\theta}_i^{(t)}) - \nabla f_i(\tilde{\theta}_i^{(t)})]\right\|^2 \\
&+ \frac{L}{2m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\tilde{\eta}_i^{(t)}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2 \\
\leq& \frac{\sigma^2}{8mT} + \frac{\eta}{8m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2.
\end{aligned}$$

Summing T_1 and T_2 we obtain

$$\begin{aligned}
T_1 + T_2 \leq& -\frac{\eta}{2}(1-\gamma)\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 + \frac{\sigma^2}{8mT} \\
&+ \frac{\eta L^2}{2m}\sum_{i=1}^{m}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2 \\
&- \frac{\eta}{4m^2}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2.
\end{aligned}$$

Defining $\gamma' = (1-\gamma)$, telescoping and taking expectations we obtain

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 \leq& 2\frac{f(\bar{\theta}^0) - f(\bar{\theta}^T)}{\eta T\gamma'} + \frac{\sigma^2}{4\eta\gamma' mT} \\
&+ \frac{1}{T}\sum_{t=1}^{T}\frac{L^2}{m\gamma'}\sum_{i=1}^{m}\mathbb{E}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2 \\
&+ \frac{1}{T}\sum_{t=1}^{T}\frac{L\zeta^2}{\eta m^2\gamma'}\sum_{i=1}^{m}\mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2.
\end{aligned}$$

Defining $\sigma_{w,i}^2 = \max_{t=0}^{T}\mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2$ and bounding the consensus term by Lemma 2, we obtain

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 \leq& 2\frac{f(\bar{\theta}^0) - f(\bar{\theta}^T)}{\eta T\gamma'} \\
&+ \frac{L^2}{m\gamma'}\left(\eta^2\frac{12mG^2}{(p\zeta)^2} + \zeta\frac{2}{p}\sum_{i=1}^{m}\sigma_{w,i}^2\right) \\
&+ \frac{\sigma^2}{4\eta\gamma' mT} + \frac{L\zeta^2}{\eta m^2\gamma'}\sum_{i=1}^{m}\sigma_{w,i}^2.
\end{aligned}$$

The final result is obtained setting $\eta = \frac{1}{\sqrt{4LT}}$ and $\zeta = \frac{1}{T^{3/8}}$.

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 *Artif. Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[2] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Trans. on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2012.

[3] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Autom. Control*, vol. 31, no. 9, pp. 803–812, 1986.

[4] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *Inter. Conf. on Machine Learning (ICML)*, pp. 5381–5393, 2020.

[5] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via sgd over wireless d2d networks," in *IEEE 21st Inter. Workshop on Sig. Proc. Adv. in Wirel. Commun. (SPAWC)*, 2020.

[6] E. Ozfatura, S. Rini, and D. Gündüz, "Decentralized SGD with over-the-air computation," in *IEEE Global Communications Conference*, 2020.

[7] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *arXiv preprint arXiv: 2101.12704v1*, 2021.

[8] Y. Shi, Y. Zhou, and Y. Shi, "Over-the-air decentralized federated learning," *arXiv preprint arXiv:2106.08011*, 2021.

[9] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[10] S. Dutta, J. Wang, and G. Joshi, "Slow and stale gradients can win the race," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1012–1024, 2021.

[11] G. Nadiradze, A. Sabour, P. Davies, I. Markov, S. Li, and D. Alistarh, "Decentralized SGD with asynchronous, local and quantized updates," *arXiv preprint arXiv:1910.12308*, 2019.

[12] T. Adikari and S. Draper, "Decentralized optimization with non-identical sampling in presence of stragglers," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3702–3706, IEEE, 2020.

[13] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying Metropolis weights," *Automatica*, vol. 1, 2006.

[14] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," *arXiv preprint arXiv:1907.09356*, 2019.