Asynchronous Federated Learning with Bidirectional Quantized Communications and Buffered Aggregation

Tomas Ortega [1] **Hamid Jafarkhani** [1]

Abstract

Asynchronous Federated Learning with Buffered Aggregation (FedBuff) is a state-of-the-art algorithm known for its efficiency and high scalability. However, it has a high communication cost, which has not been examined with quantized communications. To tackle this problem, we present a new algorithm (QAFeL), with a quantization scheme that establishes a shared "hidden" state between the server and clients to avoid the error propagation caused by direct quantization. This approach allows for high precision while significantly reducing the data transmitted during client-server interactions. We provide theoretical convergence guarantees for QAFeL and corroborate our analysis with experiments on a standard benchmark.

1. Introduction

Federated Learning (FL) is a distributed machine learning paradigm that enables training of models on decentralized data, without the need to share raw data (McMahan et al., 2017a). It has gained significant attention in recent years for its ability to mitigate privacy concerns that come with collecting sensitive information from users in a central location. Currently, FL is applied to various domains, such as natural language processing, computer vision, and healthcare (Kairouz et al., 2021b).

Many FL algorithms have been widely studied, such as FedAvg (Li et al., 2019), FedProx (Li et al., 2020), and FedSGD (McMahan et al., 2017a). These algorithms operate in a synchronous manner, i.e., all clients send updates to a central server in synchronized rounds. Since large-scale and dynamic systems are naturally asynchronous (Xu et al., 2022), there is a growing interest in studying asynchronous FL methods, where different clients can update their models and communicate with the server at different times. While asynchrony introduces additional challenges such as stale gradients and stragglers, it eliminates the need to fit clients into time slots and allows the handling of clients that are slow to respond or have limited communication capabilities (Chen et al., 2020).

Fedbuff (Nguyen et al., 2022) is an asynchronous FL algorithm that introduces a buffer on the server side to store client updates before performing a global model update. This is in contrast to previous versions of asynchronous FL, where the server sent a model update every time it received a client update, and the communication cost grew too much with the number of clients. FedBuff is also compatible with privacy-preserving technologies, has theoretical convergence guarantees under weak assumptions, and is robust to client heterogeneity. Its superiority over synchronous FL methods in terms of efficiency and fairness has been studied in (Huba et al., 2022). However, its communication cost has not been analyzed in the presence of communication compression. In this paper, we address this issue with the integration of a bidirectional quantization scheme. It is of particular interest to investigate the compound error produced by staleness and quantization, which results in a cross-error term that is not present in the separate analysis of both effects. For more related work, see Appendix A.

Contributions. As our key contributions, we

- Integrate a bidirectional quantization scheme into FedBuff to create a *"hidden" shared state* between the server and clients and *reduce communication costs*, while avoiding the error propagation caused by direct quantization. We call this algorithm Quantized Asynchronous Federated Learning (QAFeL).
- Provide a theoretical analysis of QAFeL's convergence rate. We show that quantization does not change the complexity order and prove that *FedBuff's rate can be recovered* in the limit of infinite precision quantization.
- Show that the *cross-error* term is of smaller order than the individual error from *staleness* and *quantization*.
- Validate our findings through an experimental evaluation on a standard benchmark (Caldas et al., 2018).

[1]Center for Pervasive Communications & Computing, University of California, Irvine, USA. Correspondence to: Tomas Ortega <tomaso@uci.edu>.

Workshop of Federated Learning and Analytics in Practice, of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. July 28th, 2023. This work was supported in part by the NSF Award ECCS-2207457. Copyright 2023 by the author(s).

2. System description

In both QAFeL and FedBuff, clients train asynchronously and send their local updates to the server when they have finished training. Simultaneously, the server accumulates local updates in a buffer until it has reached its maximum capacity and then produces a server model update. In FedBuff, every update is a *full-precision* model. Current machine learning models can be over tens of millions of parameters large; for example, ResNet-18 has around 11 million trainable parameters (He et al., 2016). If floating point numbers are stored in a standard 4 byte format, each client has to upload approximately 44 MB per update. Since training occurs over thousands of iterations, clients would need to upload information of the order of GB or more.

To alleviate this communication cost, QAFeL *compresses updates* using a quantizer, i.e., a lossy compressor defined as follows (Karimireddy et al., 2019; Albasyoni et al., 2020; Richtarik et al., 2021).

Definition 2.1 (Quantizer)**.** A quantizer, denoted by $Q : \mathbb{R}^d \to \mathbb{R}^d$, is a (possibly random) function that satisfies the following condition:

$$\mathbb{E}_Q \left[\|Q(x) - x\|^2 \right] \leq (1 - \delta) \|x\|^2 , \tag{1}$$

where $\delta > 0$ is a compression parameter and $\mathbb{E}_Q$ denotes the expectation with respect to the internal randomness of the quantizer Q. We use the terms quantizer and compression operator indistinctly in this work.

Using a quantizer allows us to send messages with fewer bits compared to the full model. Figure 1 sketches the system block diagram in which the server broadcasts the quantized updates. Both QAFeL and FedBuff can operate

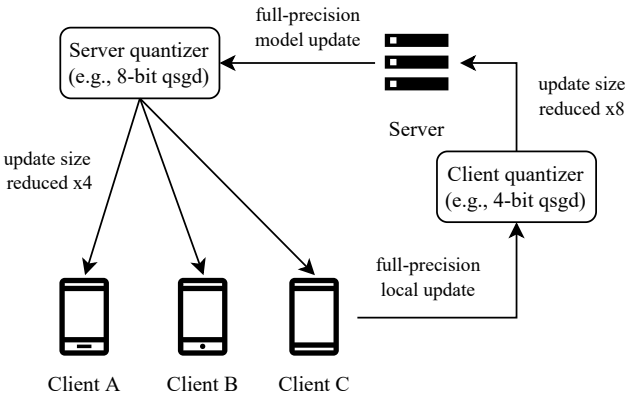

Figure 1. QAFeL system block diagram, with qsgd example quantizers. For the definition of a qsgd quantizer, see Example B.1.

in networks with or without broadcast capabilities. In this paper, we assume that both operate in the broadcast mode, that is, the server broadcasts the global update once its buffer is full. For a note regarding the non-broadcast version, see Appendix B.1.

Let us now describe how QAFeL works, and illustrate it with an execution timeline example in Figure 2, similar to the FedBuff analysis done in Figure 4 of (Huba et al., 2022). The main difference with respect to FedBuff is that

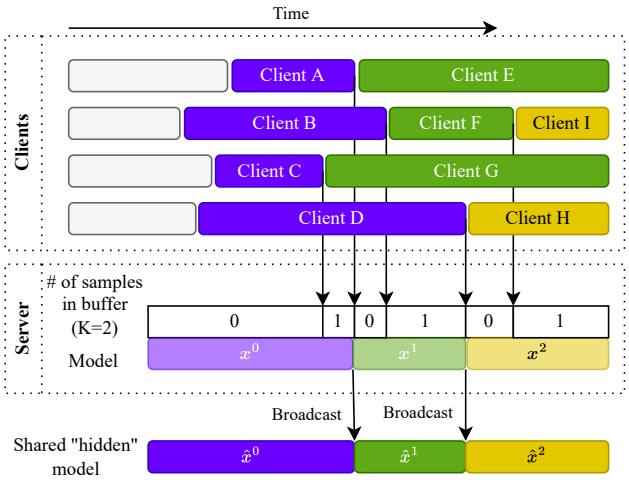

Figure 2. QAFeL example timeline, with a server with a buffer for $K = 2$ samples. Black arrows from clients to the server indicate quantized messages. Black arrows from server to the hidden model indicate a quantized broadcast message. Note that the hidden model is drawn separately from the server and the clients to indicate that it is synchronized.

QAFeL uses a "hidden" model, or hidden state, that is shared between the clients and the server. In practice, this is a model saved in the server's memory and each client, and is used as an approximation to the server model. Note that the hidden model is different from a direct quantization of the server model, and is used to avoid propagating errors. To begin training, both the server and the clients start with an initial pre-agreed upon model x^0, which is used to initiate the hidden state. The server then asynchronously samples the clients and requests them to compute a local update. A requested client will copy the current hidden state, $y_0 \leftarrow \hat{x}^t$, and perform P local updates using the equation

$$y_p \leftarrow y_{p-1} - \eta_\ell g_p(y_{p-1}), \tag{2}$$

where η_ℓ is the local learning rate and g_p is a noisy gradient. After the local updates are computed, the client sends the *quantized* difference $Q_c(y_{P-1} - y_0)$ to the server to aggregate. The server accumulates these updates in a buffer until it has K samples and then performs a global update on the model using the equation

$$x^{t+1} \leftarrow x^t + \eta_g \frac{\overline{\Delta}^t}{K}, \tag{3}$$

where $\overline{\Delta}^t$ is the sum of the local updates from the buffer. The server then updates the hidden state by computing $q^t \leftarrow Q_s(x^{t+1} - \hat{x}^t)$ and broadcasting it to the clients. The clients have a process in the background that collects q^t. Then, both the server and the clients perform the hidden state update

$$\hat{x}^{t+1} \leftarrow \hat{x}^t + q^t. \tag{4}$$

The full pseudocode for QAFeL can be found in Appendix C. Also, for a note regarding QAFeL's privacy considerations, see Appendix B.2.

3. Formulation and convergence analysis

Let us formalize the problem as a minimization of the following sum of stochastic functions:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{N} \sum_{n=1}^{N} \left(F_n(x) := \mathbb{E}_{\zeta_n}[F_n(x; \zeta_n)] \right), \tag{5}$$

where F_n is the loss function on Client n and N is the total number of clients. Each function F_n depends only on data collected locally, i.e., on Client n. Our results can be easily extended to the weighted sum case.

Let us assume that f achieves a minimum value f^*. We make the standard assumptions from the literature (Reddi et al., 2021; Li et al., 2019; Stich, 2019; Yu et al., 2019; Karimireddy et al., 2020), which are also used for FedBuff's analysis, *except* the bounded hetereogenity assumption, which is not needed in our proof.

Assumption 3.1 (Unbiased stochastic gradient)**.** We assume that for all clients, $\mathbb{E}_{\zeta_n}[g_n(x; \zeta_n))] = \nabla F_n(x), \forall x \in \mathbb{R}^d$.

Assumption 3.2 (Bounded local variance)**.** We assume that for all clients,

$$\mathbb{E}_{\zeta_n}[\|g_n(x; \zeta_n) - \nabla F_n(x)\|^2] \leq \sigma_\ell^2, \quad \forall x \in \mathbb{R}^d.$$

Assumption 3.3 (Bounded and L-smooth loss gradient)**.** We assume that each function F_n is L-smooth, that is,

$$\|\nabla F_n(x) - \nabla F_n(x')\| \leq L \|x - x'\|, \quad \forall x, x' \in \mathbb{R}^d,$$

and its variance is bounded, i.e., $\|\nabla F_n\|^2 \leq G$.

We also make one additional assumption, which is not standard for synchronous FL, but was introduced in (Nguyen et al., 2022) for the buffered asynchronous setting.

Assumption 3.4 (Bounded staleness when $K = 1$)**.** For all clients $n \in \{1, \ldots, N\}$ and for each server step t, the staleness $\tau_n(t)$, i.e., the difference in model versions between when Client n begins local training and when its updates are applied to the global model, is not greater than a maximum allowed staleness, $\tau_{\max,K}$, when the buffer size $K = 1$.

As is the case with FedBuff, it is worth noting that the upper bound on staleness depends on the buffer size, K. As the buffer size increases, the server updates less frequently, which reduces the number of server steps between when a client starts training and when its updates are applied to the global model. If assumption 3.4 is met, for any $K > 1$, the maximum delay, $\tau_{\max,K}$, is at most $\lceil \tau_{\max,1}/K \rceil$; this is proven in Appendix A of (Nguyen et al., 2022).

Proposition 3.5 (Complexity order.)**.** *Let us define the convergence rate as* $R = \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x^t)\|^2\right]$. *Under assumptions 3.1 to 3.4, with local and global learning rates satisfying Condition* (8) *in Page 12,* $\eta_\ell = \mathcal{O}\left(1/(K\sqrt{TP})\right)$ *and* $\eta_g = \mathcal{O}(K)$*, and defining* $F^* := f(x^0) - f^*$*,*

$$R_{FedBuff} = \mathcal{O}\left(\frac{F^*}{\sqrt{TP}}\right) + \mathcal{O}\left(\frac{\sigma_\ell^2 + G}{K\sqrt{TP}}\right) + \mathcal{O}\left(\frac{(\tau_{\max,1}^2 + 1)(\sigma_\ell^2 + PG)}{TK^2}\right). \tag{6}$$

Furthermore, with an unbiased client quantizer Q_c*,*

$$\begin{aligned} R_{QAFeL} \leq R_{FedBuff} &+ \mathcal{O}\left(\frac{(1-\delta_c)(\sigma_\ell^2 + G)}{K\sqrt{TP}}\right) \\ &+ \mathcal{O}\left(\frac{(2-\delta_c)(\sigma_\ell^2 + G)}{\delta_s TK}\right) \\ &+ \mathcal{O}\left(\frac{\tau_{\max,1}(1-\delta_c)(\sigma_\ell^2 + PG)}{TK^2}\right). \end{aligned} \tag{7}$$

Note that the gradient bound L is assimilated into the $\mathcal{O}(\cdot)$ notation, as done in FedBuff's analysis[1]. Proposition 3.5's proof is in Appendix F, where a version for unbiased client quantizers is also outlined. Moreover, it is also shown how a geometric partial sum in the previous expression is bounded with $1/\delta_s$, but taking the limit without this bound shows that the orders $\lim_{\delta_c, \delta_s \to 1} R_{QAFeL} = R_{FedBuff}$. In other words, QAFeL recovers FedBuff's convergence rate in the case of infinite precision quantization.

There are three error terms in (7); (i) the choice of client quantizer with an order $\mathcal{O}\left(1/\sqrt{T}\right)$; (ii) the choice of server quantizer with smaller order $\mathcal{O}(1/T)$; and (iii) the cross-error term from client quantization and staleness also with smaller order $\mathcal{O}(1/T)$. The effects of the server quantizer and the cross-error term dissipate in time faster than the effect of the client quantizer. Therefore, the choice of the client quantizer will affect the error order more than the choice of the server quantizer and also more than the staleness and quantization cross-error.

[1]The rate that appears in the original AISTATS 2022 paper has a minor error resulted from inaccuracy in Eq. (20) of that paper.

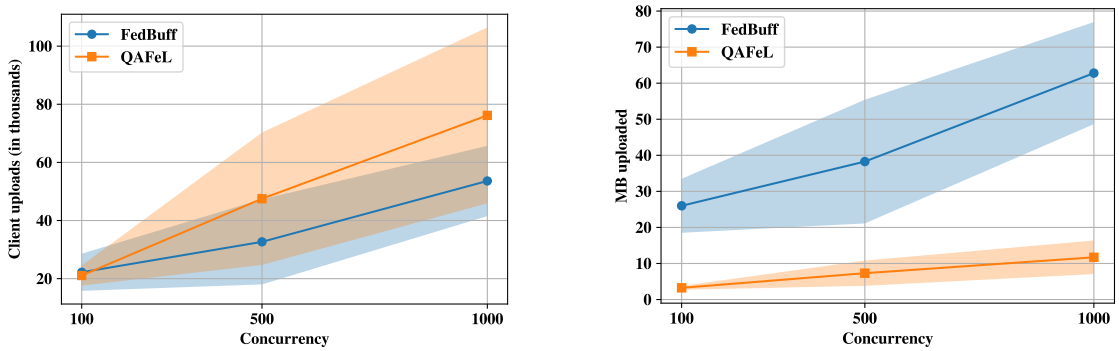

Figure 3. QAFeL and FedBuff's communication metrics to reach a validation accuracy (90%) for different concurrency values (clients training in parallel). QAFeL is using 4-bit qsgd quantization at both server and client, therefore the MB broadcasted are simply the MB uploaded divided by the buffer size, which is 10.

4. Simulation results

We conduct a series of simulations to evaluate QAFeL's performance and confirmed our theoretical derivations. Our simulations show how QAFeL's communication load is 8 times smaller than that of FedBuff while maintaining the same convergence speed.

Setup. A detailed description of the experimental setup is provided in Appendix D. The parameters are adopted from (Nguyen et al., 2022) unless they are not defined in that paper. The main traits are (i) We use the same hyper-parameters as FedBuff, (ii) we model clients arriving at a constant rate, and (iii) to simulate the time delay between a client's download and upload operation, we sample from a half-normal distribution. This distribution is selected as it provides the most accurate representation of the delay distribution observed in Meta's production FL system, see Appendix C of (Nguyen et al., 2022).

In addition to the standard metric for comparing synchronous and asynchronous FL methods (aka the number of client trips), we also present the number of bytes sent per message to illustrate QAFeL's benefits.

Dataset and model. We utilize the CelebA dataset and model inherited from (Nguyen et al., 2022), which follows the configuration of the standard LEAF benchmark (Caldas et al., 2018). The CelebA dataset (Liu et al., 2015) is a large-scale image classification dataset featuring celebrity faces with a total of 202,599 images and 10,177 annotated identities. The images in CelebA exhibit diverse variations in pose, expression, and appearance. Our task is to detect whether a celebrity is smiling or not.

Our model's high-level architecture consists of a four-layer convolutional neural network (CNN) binary classifier, with a stride of 1, a padding of 2, and a dropout rate of 0.1.

We use 4-bit qsgd quantization for both client and server – see Example B.1 for a definition of qsgd quantization and other examples. The choice of quantizer follows the results in Figure 4, where we can see QAFeL's performance for different combinations of qsgd quantizers. The results of Figure 4 are clearly consistent with the analysis presented in Section 3, as the effect of the server quantizer is less pronounced than that of the client quantizer.

Results. We perform each experiment three times and report the mean and standard deviation of the client uploads. Figure 3 illustrates how QAFeL's communication costs are lower than those of FedBuff: we decrease the MB uploaded by $5.2 - 8$ times and similarly decrease the MB broadcasted. The number of client updates is only $1 - 1.5$ times higher, but using 4-bit qsgd quantization in both client and server accounts for approximately a 8 times reduction in message size. Note that the decrease in the total MB required for upload and broadcast include the extra client updates. Additional results can be found in Appendix E.

5. Conclusion

We study the impact of bidirectional quantization on the convergence rate of buffered asynchronous FL. Using a hidden state scheme, we avoid error propagation. We show that the cross-error term is of smaller order compared to the individual terms corresponding to staleness and quantization. Empirical evaluation corroborates our analysis and shows how client quantization affects QAFeL's performance more than server quantization. The theoretical analysis presented in this work can serve as a foundation for the development of quantization schemes that ensure a specific convergence rate, for both biased and unbiased server quantizers. This approach can be used to design FL systems with bandwidth constraints, which is a common scenario.

References

Albasyoni, A., Safaryan, M., Condat, L., and Richtárik, P. Optimal gradient compression for distributed and federated learning. In *SpicyFL 2020*. arXiv, 2020. doi: 10.48550/ARXIV.2010.03246. URL https://arxiv.org/abs/2010.03246.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/6c340f25839e6acdc73414517203f5f0-Paper.pdf.

Amiri, M. M., Gunduz, D., Kulkarni, S. R., and Poor, H. V. Federated learning with quantized global model updates. In *arXiv*. arXiv, Oct 2020. doi: 10.48550/arXiv.2006.10672. URL http://arxiv.org/abs/2006.10672. arXiv:2006.10672 [cs, math].

Bonawitz, K. A., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. doi: 10.48550/ARXIV.1611.04482. URL https://arxiv.org/abs/1611.04482.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. In *NeurIPS 2019*, Dec 2018. doi: 10.48550/arXiv.1812.01097. URL https://arxiv.org/abs/1812.01097v3.

Chen, Y., Ning, Y., Slawski, M., and Rangwala, H. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 15–24, Los Alamitos, CA, USA, dec 2020. IEEE Computer Society. doi: 10.1109/BigData50022.2020.9378161. URL https://doi.ieeecomputersociety.org/10.1109/BigData50022.2020.9378161.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, volume 33, pp. 16937–16947. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html.

Gruntkowska, K., Tyurin, A., and Richtárik, P. Ef21-p and friends: Improved theoretical communication complexity for distributed optimization with bidirectional compression. In *arXiv*, Sep 2022. doi: 10.48550/arXiv.2209.15218. URL http://arxiv.org/abs/2209.15218. arXiv:2209.15218 [cs, math].

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Jun 2016. doi: 10.1109/CVPR.2016.90. URL https://ieeexplore.ieee.org/document/7780459.

Horváth, S. and Richtarik, P. A better alternative to error feedback for communication-efficient distributed learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=vYVI1CHPaQg.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4387–4398. PMLR, Nov 2020. URL https://proceedings.mlr.press/v119/hsieh20a.html.

Huba, D., Nguyen, J., Malik, K., Zhu, R., Rabbat, M., Yousefpour, A., Wu, C.-J., Zhan, H., Ustinov, P., Srinivas, H., Wang, K., Shoumikhin, A., Min, J., and Malek, M. Papaya: Practical, private, and scalable federated learning. *Proceedings of Machine Learning and Systems*, 4: 814–832, Apr 2022. URL https://proceedings.mlsys.org/paper/2022/hash/f340f1b1f65b6df5b5e3f94d95b11daf-Abstract.html.

Kairouz, P., Mcmahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5213–5225. PMLR, Jul 2021a. URL https://proceedings.mlr.press/v139/kairouz21b.html.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Nitin Bhagoji, A., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh,

A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021b. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000083. URL http://www.nowpublishers.com/article/Details/MAL-083.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes signsgd and other gradient compression schemes. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3252–3261. PMLR, May 2019. URL https://proceedings.mlr.press/v97/karimireddy19a.html.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5132–5143. PMLR, Nov 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

Karl, R., Takeshita, J., and Jung, T. Cryptonite: A framework for flexible time-series secure aggregation with non-interactive fault recovery. In Garcia-Alfaro, J., Li, S., Poovendran, R., Debar, H., and Yung, M. (eds.), *Security and Privacy in Communication Networks*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 311–331, Cham, 2021. Springer International Publishing. ISBN 978-3-030-90019-9. doi: 10.1007/978-3-030-90019-9_16. URL https://doi.org/10.1007/978-3-030-90019-9_16.

Koloskova, A., Stich, S., and Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3478–3487. PMLR, May 2019. URL https://proceedings.mlr.press/v97/koloskova19a.html.

Koloskova, A., Stich, S. U., and Jaggi, M. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. In *NeurIPS 2022*. arXiv, Jun 2022. doi: 10.48550/arXiv.2206.08307. URL http://arxiv.org/abs/2206.08307. arXiv:2206.08307 [cs, math].

Leblond, R., Pedregosa, F., and Lacoste-Julien, S. Improved asynchronous parallel optimization analysis for stochastic incremental methods. *Journal of Machine Learning Research*, 19(81):1–68, 2018. URL http://jmlr.org/papers/v19/17-650.html.

Li, L., Wang, J., and Xu, C. Flsim: An extensible and reusable simulation framework for federated learning. In Song, H. and Jiang, D. (eds.), *Simulation Tools and Techniques*, pp. 350–369, Cham, 2021. Springer International Publishing. ISBN 978-3-030-72792-5. URL https://doi.org/10.1007/978-3-030-72792-5_30.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *MLSys 2020*. arXiv, Apr 2020. URL http://arxiv.org/abs/1812.06127. arXiv:1812.06127 [cs, stat].

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *ICLR 2020*, Jul 2019. doi: 10.48550/arXiv.1907.02189. URL https://arxiv.org/abs/1907.02189v4.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society. doi: 10.1109/ICCV.2015.425. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.425.

Mania, H., Pan, X., Papailiopoulos, D., Recht, B., Ramchandran, K., and Jordan, M. I. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, Jan 2017. ISSN 1052-6234. doi: 10.1137/16M1057000. URL https://epubs.siam.org/doi/abs/10.1137/16M1057000.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, Apr 2017a. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. In *ICLR 2018*. arXiv, 2017b. doi: 10.48550/ARXIV.1710.06963. URL https://arxiv.org/abs/1710.06963.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691–706, May 2019. doi: 10.1109/SP.2019.00029. URL https://ieeexplore.ieee.org/abstract/document/8835269.

Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M., and Huba, D. Federated learning

with buffered asynchronous aggregation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 3581–3607. PMLR, May 2022. URL https://proceedings.mlr.press/v151/nguyen22b.html.

Ortega, T. and Jafarkhani, H. Gossiped and quantized online multi-kernel learning. *IEEE Signal Processing Letters*, 30:468–472, 2023. doi: 10.1109/LSP.2023.3268988. URL https://ieeexplore.ieee.org/abstract/document/10106411.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Recht, B., Re, C., Wright, S., and Niu, F. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/hash/218a0aefd1d1a4be65601cc6ddc1520e-Abstract.html.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Richtarik, P., Sokolov, I., and Fatkhullin, I. Ef21: A new, simpler, theoretically better, and practically faster error feedback. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4384–4396. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/231141b34c82aa95e48810a9d1b33a79-Abstract.html.

Stich, S., Mohtashami, A., and Jaggi, M. Critical parameters for scalable distributed learning with large batches and asynchronous updates. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 4042–4050. PMLR, Mar 2021. URL https://proceedings.mlr.press/v130/stich21a.html.

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1g2JnRcFX.

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/b440509a0106086a67bc2ea9df0a1dab-Abstract.html.

Wu, Y. and He, K. Group normalization. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pp. 3–19, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01261-8. doi: 10.1007/978-3-030-01261-8_1. URL https://doi.org/10.1007/978-3-030-01261-8.

Xie, C., Koyejo, S., and Gupta, I. Asynchronous federated optimization. In *OPT2020: 12th Annual Workshop on Optimization for Machine Learning*. arXiv, Dec 2020. doi: 10.48550/arXiv.1903.03934. URL http://arxiv.org/abs/1903.03934. arXiv:1903.03934 [cs].

Xu, C., Qu, Y., Xiang, Y., and Gao, L. Asynchronous federated learning on heterogeneous devices: A survey. In *arXiv*, Aug 2022. URL http://arxiv.org/abs/2109.04269. arXiv:2109.04269 [cs].

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7184–7193. PMLR, May 2019. URL https://proceedings.mlr.press/v97/yu19d.html.

Appendix

A. Related work

Previous work has extended the theoretical understanding of FL to the asynchronous regime (Xie et al., 2020; Recht et al., 2011; Leblond et al., 2018; Stich et al., 2021; Mania et al., 2017). Most of this work has studied the convergence behavior of asynchronous FL for the setting of homogeneously distributed data, with i.i.d client objective functions. This is well-known to be unrealistic in many instances of practical FL (Kairouz et al., 2021b). FedBuff has been proposed as a practical alternative, with buffered aggregation to handle high concurrency settings, as well as being compatible with protection against inference attacks (Nguyen et al., 2022). Sharper convergence rates compared to those of FedBuff are found in (Koloskova et al., 2022). They depend only on the average staleness, not on the maximum. Despite the better rates, this work does not accommodate buffered aggregation, which is highly desirable for scalability.

There is also extensive literature that analyzes synchronous FL with quantized communications; see, for example, (McMahan et al., 2017a), where quantization from client to server was addressed. Further studies argued the benefits of standard bidirectional quantization, which further reduces the communication cost of FL, and is of importance, particularly in wireless mediums (Amiri et al., 2020). There has also been previous work in distributed and FL with error feedback (or error control), which we use to create a "hidden" state in this work (Richtarik et al., 2021; Gruntkowska et al., 2022; Ortega & Jafarkhani, 2023; Koloskova et al., 2019). Unlike this work, the existing literature on quantized FL assumes synchronous FL.

B. Additional remarks

B.1. Non-broadcast variant.

As previously mentioned, QAFeL can be easily modified to accommodate networks without broadcast capabilities. In this case, the server must keep the hidden state updates in storage for a maximum of $C_{\max}$ updates, where $C_{\max}$ is the storage size of the model divided by the expected size of a compressed hidden state update. When the server samples an available client, the server receives the client's hidden state staleness and then transmits the necessary updates so that the hidden state is up to date. If the staleness is larger than $C_{\max}$, the server simply transmits the hidden state to the client. In both scenarios, the communication cost of QAFeL is less than or equal to that of FedBuff.

B.2. Privacy considerations.

Past research has shown that sensitive information can be recovered from the gradients of participating clients (Melis et al., 2019; Geiping et al., 2020); this is known as an interference attack. To protect against this privacy threat, FedBuff uses techniques known as secure aggregation (Karl et al., 2021; Bonawitz et al., 2016) and differential privacy (Kairouz et al., 2021a; McMahan et al., 2017b). Since our algorithm QAFeL extends FedBuff, it is also compatible with these techniques and can benefit from the same level of privacy protection. It is worth noting that our algorithm only requires the server to send perfect updates to the clients, which does not interfere with the privacy scheme of FedBuff.

B.3. Quantizer examples.

Example B.1 (Three standard quantizers). Given a vector $x \in \mathbb{R}^d$, let us define

- $\mathrm{top}_k(x)$ sends the largest k out of the d coordinates of x.
- $\mathrm{rand}_k(x)$ sends k out of the d coordinates of x, chosen at random.
- $\mathrm{qsgd}_s(x)$, given a positive integer s that sets the number of quantization levels, sends bits that represent $\|x\|$, $\mathrm{sign}(x)$, and $\xi(x,s)$,

$$\xi_i(x,s) = \begin{cases} \lceil \frac{x_i \cdot s}{\|x\|} \rceil & \text{with probability } \frac{x_i \cdot s}{\|x\|} - \lfloor \frac{x_i \cdot s}{\|x\|} \rfloor, \\ \lfloor \frac{x_i \cdot s}{\|x\|} \rfloor & \text{otherwise,} \end{cases}$$

  and the receiver can reconstruct $\frac{\|x\|}{s} \cdot \mathrm{sign}(x) \cdot \xi(x,s)$.

For top_k and rand_k, their compression parameter δ is k/d, as proven in Lemma A.1 of (Stich et al., 2018). For qsgd with s levels of quantization, $\delta = 1 - \min(\frac{2d}{s^2}, \frac{\sqrt{2d}}{s})$, see Lemma 3.1 in (Alistarh et al., 2017). Note also that top_k is the only biased quantizer out of the three. Nevertheless, there exist methods to transform top_k and general biased quantizers into unbiased ones, at the price of extra data transmission, see (Horváth & Richtarik, 2021).

One can easily see how the first two quantizers, top_k and rand_k, save data by only sending some components of the vector. For qsgd, it is important to note that $\xi_i(x, s)$ are integers that go from 0 to s. Therefore, if we only use n bits to represent these integers, we can send only n bits per coordinate instead of the full precision floating point number, which is usually 32 bits. This is called an n-bit qsgd quantizer and the number of bits per coordinate, n, automatically determines the quantization level s.

C. Full pseudocode for QAFeL

To alleviate the communication cost of asynchronous FL with buffered aggregation, we propose QAFeL, which consists of three parts: QAFeL-server, QAFeL-client, and QAFeL-client-background. QAFeL-server is stated in Algorithm 1, which runs in the server and calls QAFeL-client (Algorithm 2) when it needs an update from a client. In the background, clients are always running QAFeL-client-background, described in Algorithm 3. The highlighted texts mark the lines that differ from FedBuff. Note that Algorithm 3 is completely new.

D. Experimental details

We have utilized the scripts provided by the LEAF benchmark for non-iid client partitions, with a fixed random seed of 1549775860, to divide the users into 80% training, 10% validation, and 10% test sets, respectively. This gives us 7474, 1869, and 1869 train, validation, and test users, respectively. Each user has between 1 and 32 samples. As is standard practice for image datasets, we preprocess the train, validation, and test images. Specifically, we performed a resize and center crop operation on each image to obtain a resolution of 32×32 pixels, followed by normalizing each channel of the image to have a mean of 0.5 and a standard deviation of 0.5.

We employ a convolutional neural network (CNN) classifier as our model, which is modified from the version used in the LEAF benchmark by replacing batch normalization layers with group normalization layers (Wu & He, 2018; Hsieh et al., 2020). This modification was carried out to replicate the approach utilized in the experiments performed by FedBuff. The high-level architecture of our model consists of a four-layer CNN binary classifier, which utilizes a stride of 1, a padding of 2, and a dropout rate of 0.1.

All experiments are run 3 times, and the average is reported, along with the standard deviation. The hyperparameters we use are: client learning rate $\eta_\ell = 4.7 \cdot 10^{-6}$, server learning rate $\eta_g = 1000$, server momentum $\beta = 0.3$, and buffer size $K = 10$. Note that, as in FedBuff, we use server momentum, although the theoretical analysis does not include it. We leave the analysis with momentum for future work.

We model the arrival time and training duration of the clients as in (Nguyen et al., 2022), modeled after Meta's production FL system, with clients arriving at a constant rate, and with a training duration sampled from a half-normal distribution Y, where $Y = |X|$ and $X \sim \mathcal{N}(0, 1)$.

For experiments in Figure 3, the server uses learning rates scaled down for staleness, as done in FedBuff and (Xie et al., 2020). If client n sends an update with staleness τ_n, we scale down its weight by multiplying it by $1/\sqrt{1+\tau_n}$. To model concurrencies of 100, 500, and 1000 users we vary the constant rate at which clients arrive. The rates we select to achieve this are 125, 627, and 1253 clients per unit of time, respectively. This comes from the expected value $\sigma\sqrt{2/\pi}$ of the half-normal Y, where $Y = |X|$ and $X \sim \mathcal{N}(0, \sigma^2)$.

For the rest of experiments, we model clients arriving at 100 clients per unit of time, and no weight scaling is performed.

Our implementation is based on the FL Simulator (FLSim), which is a flexible, standalone library written in PyTorch (Li et al., 2021; Paszke et al., 2019).

Algorithm 1 QAFeL-server

input: server learning rate η_g, client learning rate η_ℓ, client SGD steps P, buffer size K, initial model x^0

$\hat{x}^0 \leftarrow x^0$ {initialize shared hidden state}
repeat
  $c \leftarrow$ sample available clients {async}
  run QAFeL-client(η_ℓ, P) on c {async}
  if receive client update **then**
    $\Delta_n \leftarrow$ received quantized update from client i
    $\overline{\Delta}^t \leftarrow \overline{\Delta}^t + \Delta_n$
    $k \leftarrow k + 1$ {number of clients in buffer}
  end if
  if $k == K$ **then**
    $\overline{\Delta}^t \leftarrow \frac{\overline{\Delta}^t}{K}$
    $x^{t+1} \leftarrow x^t + \eta_g \overline{\Delta}^t$
    Broadcast $q^t \leftarrow Q_s(x^{t+1} - \hat{x}^t)$
    $\hat{x}^{t+1} \leftarrow \hat{x}^t + q^t$ {update shared hidden state}
    $\overline{\Delta}^t \leftarrow 0, k \leftarrow 0, t \leftarrow t + 1$ {reset buffer}
  end if
until convergence

output: FL-trained global model

Algorithm 2 `QAFeL-client`

input: client learning rate η_ℓ, number of client SGD steps P

$y_0 \leftarrow \hat{x}^t$ {availability of $\hat{x}^t$ ensured by algorithm 3}
for $p = 1 : P$ **do**
  $y_p \leftarrow y_{p-1} - \eta_\ell g_p(y_{p-1})$
end for
$\Delta \leftarrow Q_c(y_0 - y_p)$ {Using an ubiased quantizer}
send Δ to server

output: client update Δ

Algorithm 3 `QAFeL-client-background`

input: initial model x^0

$\hat{x}^0 \leftarrow x^0$
repeat
  wait for quantized update q^t
  $\hat{x}^{t+1} \leftarrow \hat{x}^t + q^t$
until shutdown

output: updated FL global model

E. Additional results

In Figure 4 we observe how, given a client quantizer, quantizing with less precision at the server always results in less total bytes downloaded. On the other hand, we also observe how quantizing with less precision at the client sometimes results in more total bytes uploaded, e.g., going from 4 to 2-bit qsgd at the client, with 8-bit qsgd at the server. We also see that the number of uploads augments from 4-bit to 2-bit client qsgd, with no significant reduction in total upload bytes. This illustrates a trade-off between the amount of quantization and the speed of convergence. In other words, compressing more will ensure fewer bytes per message are sent, but more messages will have to be sent to reach the target accuracy. However, overall, QAFeL requires less total number of communication bits compared with FedBuff. The optimal level of quantization in QAFeL depends on the choice of the quantizer, as is reflected in our results.

Figure 4. Communication metrics of QAFeL and FedBuff to reach target validation accuracy (90%) with varying choices of qsgd quantizers. In all cases QAFeL saves at least $3\times$ upload cost and $2\times$ broadcast cost. Notice that in 4-bit qsgd client and server, the convergence speed is the same for both algorithms, and the upload message size goes down from 3052 MB to 428 MB, a $7\times$ decrease, with analogous decrease in download cost.

Table 1 shows the performance of QAFeL with varying levels of qsgd quantization both on the server and on the client. This is the data used to produce Figure 4. For completeness, we present results with a biased server quantizer in Table 2.

Table 1. Communications metrics of QAFeL to reach target validation accuracy (90%), with different combinations of qsgd.

Algorithm		Uploads (in thousands)	kB/upload	kB/download
FedBuff		26.1 ± 6.7	117.128	117.128
QAFeL client 8-bit qsgd,	server 8-bit qsgd	27.6 ± 4.82	29.924	29.924
	server 4-bit qsgd	29.8 ± 2.76	29.924	15.380
	server 2-bit qsgd	35.7 ± 14.5	29.924	8.108
QAFeL client 4-bit qsgd,	server 8-bit qsgd	39.7 ± 3.32	15.380	29.924
	server 4-bit qsgd	27.8 ± 10.4	15.380	15.380
	server 2-bit qsgd	37.7 ± 9.10	15.380	8.108
QAFeL client 2-bit qsgd,	server 8-bit qsgd	58.6 ± 7.16	8.108	29.924
	server 4-bit qsgd	74.6 ± 35.7	8.108	15.380
	server 2-bit qsgd	91.9 ± 34.4	8.108	8.108

F. Convergence analysis and proof

In this appendix, we discuss QAFeL's convergence analysis and provide the main theorem, a sketch of its proof that contains the main ideas, and an detailed proof along with the corollaries we use to back Proposition 3.5 in the main body. We also provide the condition on the learning rates necessary for the convergence rate to hold. For the case of biased server quantizers, we provide a looser bound on the convergence rate. Finally, we analyze the convergence rate's order of

Table 2. Communications metrics of QAFeL to reach target validation accuracy (90%). QAFeL-server uses top_k to send only the top 10% of coordinates.

Algorithm	Uploads (in thousands)	kB/upload	kB/download
FedBuff	26.1 ± 6.7	117.128	117.128
QAFeL (client 8-bit qsgd)	39.5 ± 10.2	29.924	15.404
QAFeL (client 4-bit qsgd)	42.4 ± 10.4	15.380	15.404
QAFeL (client 2-bit qsgd)	$94.8 \pm 16.7^*$	8.108	15.404

*The third iteration of 2-bit qsgd only achieved a maximum of 88.62% accuracy after 150k client uploads, so it is not used in this computation.

complexity and compare with the rate for FedBuff, i.e., the case with infinite precision. Proposition 3.5 follows from this analysis, and it is specifically a direct consequence of Corollary F.3.

To facilitate the notation, we introduce two quantities, α_P and β_P, defined as follows:

$$\alpha_P := \sum_{p=0}^{P-1} \eta_\ell^{(p)}, \quad \beta_P := \sum_{p=0}^{P-1} (\eta_\ell^{(p)})^2.$$

Conditions on the learning rates. All learning rates η_g, $\eta_\ell^{(p)}$, for all $p \in \{0, \dots, P-1\}$, must satisfy

$$\left(\frac{\alpha_P 3L^2\eta_g^2}{\delta_s} + L\eta_g\right)\left(1 + \frac{1-\delta_c}{K}\right) P\eta_\ell^{(p)} \leq 1. \tag{8}$$

For a simpler and less precise bound, if $\eta_g \leq \frac{1}{L}$ and $\eta_\ell^{(p)} \leq \min\left(\frac{K}{2P(K+1-\delta_c)}, \frac{\delta_s}{3P}\right)$, the condition is satisfied.

Theorem F.1 (Convergence of QAFeL)**.** *Choosing local learning rates $\eta_\ell^{(p)}$ and global learning rate η_g that satisfy Condition* (8)*, and unbiased server and client quantizers, the ergodic convergence rate for T iterations of QAFeL is upper bound by the following:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \frac{2\left(f(x^0) - f^*\right)}{\eta_g T \alpha_P} + 3L^2\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}} + 1\right) + 1\right)(\sigma_\ell^2 + PG)$$
$$+ \left(\frac{L\eta_g}{\alpha_P} + 3L^2\eta_g^2\sum_{t=1}^{T-1}(1-\delta_s)^t\right)\left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2 + 4G\right). \tag{9}$$

Proof sketch. In this proof, we employ L-smoothness to bound how the loss function from (5) evolves with our algorithm, i.e.,

$$f(x^{t+1}) \leq f(x^t) - \eta_g\left\langle \nabla f(x^t), \overline{\Delta}^t\right\rangle + \frac{L\eta_g^2}{2}\left\|\overline{\Delta}^t\right\|^2. \tag{10}$$

Next, we take advantage of the unbiasedness of the client quantizer to compute the expected value of the second term of the RHS of (10). Through some further manipulation, we bound the expected value with three terms, each representing a distinct source of error: staleness, quantization, and local drift. To bound each of these three terms, we utilize the remaining assumptions. We further bound the $\left\|\overline{\Delta}^t\right\|^2$ term in (10) using an inductive proof. By rearranging the terms, we are left with some loose terms. To cancel these out, we derive the condition on the learning rate, ultimately concluding the proof.

A similar proof works for a possibly biased server quantizer, stated in Corollary F.2, with the following condition on the learning rates

$$\left(\frac{12\alpha_P L^2\eta_g^2}{\delta_s^2} + L\eta_g\right)\left(1 + \frac{1-\delta_c}{K}\right) P\eta_\ell^{(p)} \leq 1. \tag{11}$$

Similarly to the unbiased case, for a simpler and less precise bound, if $\eta_g \leq \frac{1}{L}$ and $\eta_\ell^{(p)} \leq \min\left(\frac{K}{2P(K+1-\delta_c)}, \frac{\delta_s^2}{12P}\right)$, the condition is satisfied.

Corollary F.2 (Using a biased server quantizer)**.** *Choosing local learning rates $\eta_\ell^{(p)}$ and global learning rate η_g that satisfy Condition* (11)*, an unbiased client quantizer, and a* possibly biased *server quantizer, the ergodic convergence rate for T iterations of QAFeL is upper bounded by the following:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \frac{2\left(f(x^0)-f^*\right)}{\eta_g T\alpha_P} + 3L^2\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}}+1\right)+1\right)(\sigma_\ell^2+PG) + \left(\frac{L\eta_g}{\alpha_P}+\frac{12L^2\eta_g^2}{\delta_s^2}\right)\left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2+4G\right). \quad (12)$$

Complexity analysis. Under the conditions of Theorem F.1, we can choose learning rates to give the following corollary on complexity.

Corollary F.3 (Complexity order)**.** *Let us choose local and global learning rates that satisfy the condition described in* (8) *and define $F^* := f(x^0) - f^*$. By also choosing $\eta_\ell = \mathcal{O}\left(1/(K\sqrt{TP})\right)$ and $\eta_g = \mathcal{O}(K)$, for a sufficiently large T, it holds that*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \mathcal{O}\left(\frac{F^*}{\sqrt{TP}}\right) + \mathcal{O}\left(\frac{L^2(\sigma_\ell^2+PG)\left(\tau_{\max,1}^2+\tau_{\max,1}(1-\delta_c)+1\right)}{TK^2}\right) + \mathcal{O}\left(\frac{L^2(2-\delta_c)(\sigma_\ell^2+4G)}{\delta_s TK}\right) + \mathcal{O}\left(\frac{L(2-\delta_c)(\sigma_\ell^2+4G)}{K\sqrt{TP}}\right). \quad (13)$$

From the bound described in Corollary F.3, we can see the first standard term for SGD is present in our analysis. In addition, there are three terms that depend on the maximum staleness $\tau_{\max,1}$, the local variance σ_ℓ^2, the gradient bound G, and the choice of client and server quantizers. Observe that the choice of client quantizer affects the order of error in a term that decreases with $1/\sqrt{T}$, while the choice of server quantizer only affects a term that decreases with $1/T$. This means that although $\sum_{t=1}^{T-1}(1-\delta_s)^t$ can be large, it is bounded by $1/\delta_s$, and the effect of the server quantizer dissipates in time faster than the effect of the client quantizer. In conclusion, the choice of client quantizer affects the order of error more than the choice of server quantizer.

F.1. Detailed proof of the main theorem.

Let us now prove the general version of Theorem F.1, stated here for both biased and un-biased server quantizers, defining $\phi(T)$ as follows:

$$\phi(T) = \begin{cases} \frac{2}{\delta_s}\sum_{t=0}^{T-1}(1-\frac{\delta_s}{2})^{t-1} \leq \frac{4}{\delta_s^2} & \text{when } Q_s \text{ is biased,} \\ \sum_{t=1}^{T-1}(1-\delta_s)^t \leq \frac{1}{\delta_s} & \text{otherwise.} \end{cases} \quad (14)$$

Theorem F.4 (Convergence of QAFeL - General version)**.** *Choosing local learning rates $\eta_\ell^{(p)}$, and global learning rate η_g that satisfy the condition* (8)*, the ergodic convergence rate for the iterates of QAFeL is upper bound by the following:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \frac{2\left(f(x^0)-f^*\right)}{\eta_g T\alpha_P} + 3L^2\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}}+1\right)+1\right)(\sigma_\ell^2+PG) + \left(3L^2\eta_g^2\phi(T)+\frac{L\eta_g}{\alpha_P}\right)\left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2+4G\right).$$

We use the framework of (Nguyen et al., 2022), which in turn follows from (Mania et al., 2017). The summary of notation used in the proof is found in Table 3.

Let us start by formally describing the iterates of Algorithm 1,

$$x^{t+1} = x^t + \eta_g\overline{\Delta}^t = x^t + \eta_g\frac{1}{K}Q_c\left(\sum_{k\in\mathcal{S}^t}\left(-\eta_\ell\sum_{p=1}^{P}g_k(y_{k,p}^{t-\tau_k(t)})\right)\right).$$

Table 3. Summary of notation used in the proof.

$x^t, \hat{x}^t$	server, shared hidden state at time t
L	L-smoothness constant of the loss function
P, p	number, index of local steps at client
K, k	number, index of clients at the buffer
N, n	number, index of total clients
$\eta_g, \eta_\ell^{(p)}$	server, client (at step p) learning rates
Q_s, Q_c	server, client quantizers
$\overline{\Delta}^t, \Delta_k^t$	server, client k's update at time t
$\mathcal{S}^t$	set of client indices at the buffer at time t
$y_{k,p}^t$	local state at client k, during local step p at time t
$\pm$	plus and minus the same quantity, i.e., $a \pm b = a + b - b = a$

By assumption 3.3, we can use L-smoothness to compute

$$\begin{aligned} f(x^{t+1}) &\leq f(x^t) - \eta_g \left\langle \nabla f(x^t), \overline{\Delta}^t \right\rangle + \frac{L\eta_g^2}{2} \left\| \overline{\Delta}^t \right\|^2 \\ &= f(x^t) \underbrace{- \frac{\eta_g}{K} \sum_{k \in \mathcal{S}_t} \left\langle \nabla f(x^t), \Delta_k^{t-\tau_k} \right\rangle}_{T_{1,t}} + \frac{L\eta_g^2}{2} \left\| \overline{\Delta}^t \right\|^2 . \end{aligned} \tag{15}$$

Let us first use Lemma F.7, which gives us the expected value of $T_{1,t}$,

$$\begin{aligned} \mathbb{E}\left[T_{1,t}\right] = &- \frac{\eta_g}{2} \left(\sum_{p=0}^{P-1} \eta_\ell^{(p)} \right) \mathbb{E}\left[\left\| \nabla f(x^t) \right\|^2 \right] \\ &+ \sum_{p=0}^{P-1} \frac{\eta_g \eta_\ell^{(p)}}{2} \mathbb{E} \left[- \left\| \frac{1}{N} \sum_{n=1}^{N} \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2 + \underbrace{\left\| \nabla f(x^t) - \frac{1}{N} \sum_{n=1}^{N} \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2}_{T_{2,t}} \right] . \end{aligned} \tag{16}$$

Using the definition of α_P, we can plug (16) into (15), and take expectations to obtain

$$\begin{aligned} \mathbb{E}\left[f(x^{t+1})\right] \leq &\mathbb{E}\left[f(x^t)\right] - \frac{\eta_g}{2} \alpha_P \mathbb{E}\left[\left\| \nabla f(x^t) \right\|^2 \right] - \frac{\eta_g}{2} \sum_{p=0}^{P-1} \eta_\ell^{(p)} \mathbb{E} \left[\left\| \frac{1}{N} \sum_{n=1}^{N} \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2 \right] \\ &+ \frac{\eta_g}{2} \alpha_P \mathbb{E}\left[T_{2,t}\right] + \frac{L\eta_g^2}{2} \mathbb{E}\left[\left\| \overline{\Delta}^t \right\|^2 \right] . \end{aligned} \tag{17}$$

If we rearrange and sum for $t = 0, \ldots, T-1$, we obtain

$$\begin{aligned} \frac{2}{\eta_g} \mathbb{E}\left[f(x^T) - f(x^0)\right] \leq &- \alpha_P \sum_{t=0}^{T-1} \mathbb{E}\left[\left\| \nabla f(x^t) \right\|^2 \right] - \sum_{t=0}^{T-1} \sum_{p=0}^{P-1} \eta_\ell^{(p)} \mathbb{E} \left[\left\| \frac{1}{N} \sum_{n=1}^{N} \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2 \right] \\ &+ \alpha_P \sum_{t=0}^{T-1} \mathbb{E}\left[T_{2,t}\right] + L\eta_g \sum_{t=0}^{T-1} \mathbb{E}\left[\left\| \overline{\Delta}^t \right\|^2 \right] . \end{aligned} \tag{18}$$

Dividing by T and rearranging further, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \frac{2}{\eta_g T\alpha_P}\mathbb{E}\left[f(x^0)-f(x^T)\right] - \frac{1}{T\alpha_P}\sum_{t=0}^{T-1}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right] + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[T_{2,t}\right] + \frac{L\eta_g}{T\alpha_P}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right]. \tag{19}$$

Notice that the LHS of our expression is now identical to the problem statement. Let us continue to develop the RHS. Using Lemma F.8, we can bound $\sum_{t=0}^{T-1}\mathbb{E}\left[T_{2,t}\right]$ with

$$\sum_{t=0}^{T-1}\mathbb{E}\left[T_{2,t}\right] \leq 3L^2T\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}}+1\right)+1\right)(\sigma_\ell^2+PG) + 3L^2\eta_g^2\phi(T)\sum_{t=0}^{T-2}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right], \tag{20}$$

where $\phi(T)$ is a function that depends on the choice of quantizer at the server, as well as the number of iterations T, see (14).

Plugging this bound into (19), we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq{}& \frac{2}{\eta_g T\alpha_P}\mathbb{E}\left[f(x^0)-f(x^T)\right] - \frac{1}{T\alpha_P}\sum_{t=0}^{T-1}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]\\ &+ 3L^2\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}}+1\right)+1\right)(\sigma_\ell^2+PG)\\ &+\left(3L^2\eta_g^2\phi(T)+\frac{L\eta_g}{\alpha_P}\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|\right]^2.\end{aligned} \tag{21}$$

Now, we can bound the last term of the RHS with the bound derived in Appendix F.5, which states

$$\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] \leq \left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2+4G\right) + \left(1+\frac{1-\delta_c}{K}\right)P\sum_{p=0}^{P-1}(\eta_\ell^{(p)})^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right], \tag{22}$$

and we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq{}& \frac{2}{\eta_g T\alpha_P}\mathbb{E}\left[f(x^0)-f(x^T)\right]\\ &- \frac{1}{T\alpha_P}\sum_{t=0}^{T-1}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]\\ &+ 3L^2\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}}+1\right)+1\right)(\sigma_\ell^2+PG)\\ &+\left(3L^2\eta_g^2\phi(T)+\frac{L\eta_g}{\alpha_P}\right)\left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2+4G\right)\\ &+\left(3L^2\eta_g^2\phi(T)+\frac{L\eta_g}{\alpha_P}\right)\left(1+\frac{1-\delta_c}{K}\right)\frac{P}{T}\sum_{t=0}^{T-1}\sum_{p=0}^{P-1}(\eta_\ell^{(p)})^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right].\end{aligned} \tag{23}$$

Observe that we can impose that the sum of the second term and the last term of the RHS be non-positive, as long as

$$\left(3L^2\eta_g^2\phi(T)+\frac{L\eta_g}{\alpha_P}\right)\left(1+\frac{1-\delta_c}{K}\right)P\eta_\ell^{(p)} - \frac{1}{\alpha_P} \leq 0, \quad \forall p\in\{0,\dots,P-1\}. \tag{24}$$

Rearranging, we achieve the equivalent condition

$$\left(\alpha_P 3L^2\eta_g^2\phi(T)+L\eta_g\right)\left(1+\frac{1-\delta_c}{K}\right)P\eta_\ell^{(p)} \leq 1, \quad \forall p\in\{0,\dots,P-1\}. \tag{25}$$

For a simpler, less precise bound, remark that if $\eta_g \leq \frac{1}{L}$ and $\eta_\ell^{(p)} \leq \min\left(\frac{K}{2P(K+1-\delta_c)}, \frac{1}{3\phi(T)P}\right)$ the condition is satisfied. Assuming that (25) is satisfied, (23) becomes

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \frac{2}{\eta_g T\alpha_P}\mathbb{E}\left[f(x^0)-f(x^T)\right] + 3L^2\beta_P\left(\eta_g^2\tau_{\max,K}^2\left(\frac{1-\delta_c}{K\tau_{\max,K}}+1\right)+1\right)(\sigma_\ell^2+PG) + \left(3L^2\eta_g^2\phi(T)+\frac{L\eta_g}{\alpha_P}\right)\left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2+4G\right). \tag{26}$$

Finally, using f^* as a lower bound for all values of f, we conclude the proof.

F.2. Technical lemmas

Lemma F.5 (Stochastic gradient bound)**.** *For any $x \in \mathbb{R}^d$, and any learning rates $\eta_\ell^{(p)}$,*

$$\mathbb{E}\left[\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}g_k(x)\right\|^2\right] \leq \beta_P\left(\sigma_\ell^2+PG\right). \tag{27}$$

Proof. Adding and subtracting the true client gradient,

$$\mathbb{E}\left[\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}(g_k(x)\pm F_k(x))\right\|^2\right] = \sum_{p=0}^{P-1}\mathbb{E}\left[\left\|\eta_\ell^{(p)}(g_k(x)-F_k(x))\right\|^2\right] + \mathbb{E}\left[\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}F_k(x)\right\|^2\right]. \tag{28}$$

where the equality follows from the unbiasedness of g_k. Applying Cauchy-Schwarz gives us the statement of the lemma. □

Lemma F.6 (Sum of unbiasedly quantized terms bound)**.** *For any set of $n \geq 1$ vectors $\{x_n \in \mathbb{R}^d, n=1,\ldots,n\}$, and any unbiased compression operator $Q:\mathbb{R}^d \to \mathbb{R}^d$ satisfying Definition 2.1, we can ensure*

$$\mathbb{E}_Q\left\|\sum_{n=1}^n Q(x_n)\right\|^2 \leq \left\|\sum_{n=1}^n x_n\right\|^2 + (1-\delta)\sum_{n=1}^n\|x_n\|^2.$$

Proof. From the identity

$$\|a+b\|^2 = \|a\|^2+\|b\|^2+2\langle a,b\rangle,$$

we can deduce

$$\mathbb{E}_Q\left\|\sum_{n=1}^n Q(x_n)\right\|^2 = \mathbb{E}_Q\left\|\sum_{n=1}^n (Q(x_n)\pm x_n)\right\|^2 = \mathbb{E}_Q\left\|\sum_{n=1}^n (Q(x_n)-x_n)\right\|^2 + \left\|\sum_{n=1}^n x_n\right\|^2 + 2\mathbb{E}_Q\langle\sum_{n=1}^n (Q(x_n)-x_n), \sum_{n=1}^n x_n\rangle = \mathbb{E}_Q\left\|\sum_{n=1}^n (Q(x_n)-x_n)\right\|^2 + \left\|\sum_{n=1}^n x_n\right\|^2, \tag{29}$$

where the last equality follows from the unbiasedness of Q. Similarly, since $Q(x_n)-x_n$ are independent with respect to the randomness of Q, and Q is unbiased,

$$\mathbb{E}_Q\left\|\sum_{n=1}^n (Q(x_n)-x_n)\right\|^2 = \mathbb{E}_Q\sum_{n=1}^n\|Q(x_n)-x_n\|^2 \leq (1-\delta)\sum_{n=1}^n\|x_n\|^2. \tag{30}$$

Joining the past two expressions gives us the statement of the lemma. □

Lemma F.7 (Expected value of $T_{1,t}$)**.** *For the iterates of QAFeL, and defining $T_{1,t}$ as*

$$T_{1,t} := -\frac{\eta_g}{K}\sum_{k\in\mathcal{S}_t}\left\langle\nabla f(x^t), \Delta_k^{t-\tau_k}\right\rangle,$$

it holds that

$$\mathbb{E}\left[T_{1,t}\right] = -\frac{\eta_g}{2}\left(\sum_{p=0}^{P-1}\eta_\ell^{(p)}\right)\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] + \sum_{p=0}^{P-1}\frac{\eta_g\eta_\ell^{(p)}}{2}\mathbb{E}\left[-\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2 + \underbrace{\left\|\nabla f(x^t) - \frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2}_{T_{2,t}}\right]. \tag{31}$$

Proof. First, let us use the definition of $\Delta_k^{t-\tau_k}$, which is the client k's update sent to the server at time $t-\tau_k$. We can take the expectation of $T_{1,t}$ to obtain

$$\mathbb{E}\left[T_{1,t}\right] = \mathbb{E}\left[-\frac{\eta_g}{K}\sum_{k\in\mathcal{S}_t}\left\langle\nabla f(x^t), Q_c\left(\sum_{p=0}^{P-1}\eta_\ell^{(p)}g_k(y_{k,p}^{t-\tau_k})\right)\right\rangle\right] = \mathbb{E}\left[-\frac{\eta_g}{K}\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\left\langle\nabla f(x^t), g_k(y_{k,p}^{t-\tau_k})\right\rangle\right], \tag{32}$$

because the client quantizer Q_c is unbiased, and its internal randomness is independent of all other variables. Now, following the same logic as in (Nguyen et al., 2022) we use the conditional expectation

$$\mathbb{E}\left[\cdot\right] := \mathbb{E}_{\mathcal{H}}\mathbb{E}_{n\sim[N]}\mathbb{E}_{g_n|n,\mathcal{H}}[\cdot],$$

where $\mathcal{H}$ corresponds to the randomness of the history of the iterates, and $g_n \mid n, \mathcal{H}$ corresponds to the randomness of the stochastic gradient g_n, conditioned to the fact that we chose client n, and we have the iterates defined by $\mathcal{H}$. We have used $n \sim [N]$ to indicate that the client n is sampled randomly and uniformly over $1, \ldots, N$ at each time t. The expectation with respect to the quantization randomness is implicit in this notation. Using assumption 3.1, we can ensure that the stochastic gradient is unbiased, and we obtain

$$\begin{aligned}\mathbb{E}\left[T_{1,t}\right] &= -\,\mathbb{E}\left[\frac{\eta_g}{K}\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\left\langle\nabla f(x^t), g_k(y_{k,p}^{t-\tau_k})\right\rangle\right]\\ &= -\,\eta_g\mathbb{E}_{\mathcal{H}}\left[\frac{1}{N}\sum_{n=1}^{N}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\mathbb{E}_{g_n|n\sim[N]}\left\langle\nabla f(x^t), g_n(y_{n,p}^{t-\tau_n})\right\rangle\right]\\ &= -\,\eta_g\mathbb{E}_{\mathcal{H}}\left[\sum_{p=0}^{P-1}\eta_\ell^{(p)}\left\langle\nabla f(x^t), \frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\rangle\right].\end{aligned} \tag{33}$$

Notice that we have removed the dependency on the client selection, so we can simply write

$$\mathbb{E}\left[T_{1,t}\right] = -\eta_g\mathbb{E}\left[\sum_{p=0}^{P-1}\eta_\ell^{(p)}\left\langle\nabla f(x^t), \frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\rangle\right]. \tag{34}$$

Finally, from the identity

$$\langle a, b\rangle = \frac{1}{2}(\|a\|^2 + \|b\|^2 - \|a-b\|^2),$$

we can develop the previous inner product and obtain

$$\mathbb{E}\left[T_{1,t}\right] = -\frac{\eta_g}{2}\left(\sum_{p=0}^{P-1}\eta_\ell^{(p)}\right)\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] + \sum_{p=0}^{P-1}\frac{\eta_g\eta_\ell^{(p)}}{2}\mathbb{E}\left[-\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2 + \left\|\nabla f(x^t) - \frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]. \tag{35}$$

□

Lemma F.8 (Expected value of $T_{2,t}$)**.** *For the iterates of QAFeL, and defining $T_{2,t}$ as*

$$T_{2,t} := \left\| \nabla f(x^t) - \frac{1}{N}\sum_{n=1}^{N} \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2,$$

it holds that, for any $T > 0$,

$$\sum_{t=0}^{T-1} \mathbb{E}\left[T_{2,t}\right] \leq 3L^2 T \beta_P \left(\eta_g^2 \tau_{\max,K}^2 \left(\frac{1-\delta_c}{K\tau_{\max,K}} + 1 \right) + 1 \right) (\sigma_\ell^2 + PG) + 3L^2 \eta_g^2 \phi(T) \sum_{t=0}^{T-2} \mathbb{E}\left[\left\| \overline{\Delta}^t \right\|^2 \right]. \tag{36}$$

Proof. Let us start expanding the expected value of $T_{2,t}$, and applying the Cauchy–Schwarz inequality

$$\mathbb{E}\left[T_{2,t}\right] = \mathbb{E}\left[\left\| \frac{1}{N}\sum_{n=1}^{N} \nabla F_n(x^t) - \frac{1}{N}\sum_{n=1}^{N} \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2 \right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left[\left\| \nabla F_n(x^t) - \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2 \right]. \tag{37}$$

We apply the L-smoothness assumption from assumption 3.3 to each term in the previous sum, and telescope them with

$$\mathbb{E}\left[\left\| \nabla F_n(x^t) - \nabla F_n(y_{n,p}^{t-\tau_n}) \right\|^2 \right] \leq L^2 \mathbb{E}\left[\left\| x^t - y_{n,p}^{t-\tau_n} \right\|^2 \right] = L^2 \mathbb{E}\left[\left\| x^t \pm x^{t-\tau_n} \pm \hat{x}^{t-\tau_n} - y_{n,p}^{t-\tau_n} \right\|^2 \right]. \tag{38}$$

Again, by Cauchy–Schwarz, we obtain the following decomposition

$$\mathbb{E}\left[T_{2,t}\right] \leq \frac{3L^2}{N}\sum_{n=1}^{N} \mathbb{E}\left[\underbrace{\left\| x^t - x^{t-\tau_n} \right\|^2}_{\text{staleness}} + \underbrace{\left\| x^{t-\tau_n} - \hat{x}^{t-\tau_n} \right\|^2}_{\text{quantization}} + \underbrace{\left\| \hat{x}^{t-\tau_n} - y_{n,p}^{t-\tau_n} \right\|^2}_{\text{local drift}} \right]. \tag{39}$$

In Appendix F.3, we derive the bounds for these three terms. We can use the three derived bounds in (43), (45) and (47), to sum for $t = 0, \ldots, T-1$ and obtain

$$\sum_{t=0}^{T-1} \mathbb{E}\left[T_{2,t}\right] \leq 3L^2 T \beta_P \left(\eta_g^2 \tau_{\max,K}^2 (\sigma_\ell^2 + PG) \left(\frac{1-\delta_c}{K\tau_{\max,K}} + 1 \right) + (\sigma_\ell^2 + PG) \right) + 3L^2 \eta_g^2 \phi(T) \sum_{t=0}^{T-2} \mathbb{E}\left[\left\| \overline{\Delta}^t \right\|^2 \right]. \tag{40}$$

□

F.3. Bounding the effects of staleness, quantization, and local drift

Staleness. The staleness term gives us the error inferred by the τ_n steps of difference between the current server model and the model that the client is training with at the time it sends the update. We can start by using the definition of x^t,

$$\left\| x^t - x^{t-\tau_n} \right\|^2 = \left\| \sum_{\rho=t-\tau_n}^{t-1} \overline{\Delta}^\rho \right\|^2 = \left\| \sum_{\rho=t-\tau_n}^{t-1} \frac{\eta_g}{K} \sum_{j_\rho \in \mathcal{S}_\rho} \Delta_{j_\rho}^\rho \right\|^2 = \frac{\eta_g^2}{K^2} \left\| \sum_{\rho=t-\tau_n}^{t-1} \sum_{j_\rho \in \mathcal{S}_\rho} Q_c \left(\sum_{p'=0}^{P-1} \eta_\ell^{(p')} g_{j_\rho}(y_{j_\rho,p'}^\rho) \right) \right\|^2. \tag{41}$$

We can then use the unbiasedness of Q_c and apply Lemma F.6, so

$$\begin{aligned} \mathbb{E}\left[\left\| x^t - x^{t-\tau_n} \right\|^2 \right] \leq \frac{\eta_g^2}{K^2} \mathbb{E}\left[\left\| \sum_{\rho=t-\tau_n}^{t-1} \sum_{j_\rho \in \mathcal{S}_\rho} \sum_{p'=0}^{P-1} \eta_\ell^{(p')} g_{j_\rho}(y_{j_\rho,p'}^\rho) \right\|^2 \right] \\ + (1-\delta_c) \frac{\eta_g^2}{K^2} \mathbb{E}\left[\sum_{\rho=t-\tau_n}^{t-1} \sum_{j_\rho \in \mathcal{S}_\rho} \left\| \sum_{p'=0}^{P-1} \eta_\ell^{(p')} g_{j_\rho}(y_{j_\rho,p'}^\rho) \right\|^2 \right]. \end{aligned} \tag{42}$$

We can now apply Cauchy-Schwarz, the definition of β_P, and Lemma F.5 to obtain

$$\begin{aligned} \mathbb{E}\left[\left\| x^t - x^{t-\tau_n} \right\|^2 \right] &\leq \frac{\eta_g^2}{K^2} \tau_{\max,K}^2 K^2 \beta_P (\sigma_\ell^2 + PG) + (1-\delta_c) \frac{\eta_g^2}{K^2} \tau_{\max,K} K \beta_P (\sigma_\ell^2 + PG) \\ &= \eta_g^2 \tau_{\max,K}^2 \beta_P (\sigma_\ell^2 + PG) \left(\frac{1-\delta_c}{K\tau_{\max,K}} + 1 \right). \end{aligned} \tag{43}$$

Local drift. The local drift term tells us the norm of the difference between the p-th local step and the initial hidden state with which the local training started.

$$\mathbb{E}\left[\left\|\hat{x}^{t-\tau_n} - y_{n,p}^{t-\tau_n}\right\|^2\right] = \mathbb{E}\left[\left\|y_{i,0}^{t-\tau_n} - y_{n,p}^{t-\tau_n}\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{p'=0}^{p-1} \eta_\ell^{(p')} g_n(y_{n,p'}^{t-\tau_n})\right\|^2\right]. \tag{44}$$

Applying the definition of β_P and Lemma F.5 yields

$$\mathbb{E}\left[\left\|\hat{x}^{t-\tau_n} - y_{n,p}^{t-\tau_n}\right\|^2\right] \leq \beta_P(\sigma_\ell^2 + PG). \tag{45}$$

Quantization. The quantization term tells us how much the server state and the hidden state differ at time $t - \tau_n$. This term requires more work, and we divide it into two cases, depending on whether the server quantizer Q_s is biased or not. Both these cases are discussed in Appendix F.4. We then add the expectation of the quantization terms for $t = 0, \ldots, T-1$, as this simplifies the notation with the definition of $\phi(T)$ we introduced in (14). For clarity, we shall restate it here:

$$\phi(T) = \begin{cases} \frac{2}{\delta_s}\sum_{t=0}^{T-1}(1-\frac{\delta_s}{2})^{t-1} \leq \frac{4}{\delta_s^2} & \text{when } Q_s \text{ is biased,} \\ \sum_{t=1}^{T-1}(1-\delta_s)^t \leq \frac{1}{\delta_s} & \text{otherwise.} \end{cases} \tag{46}$$

The bounds made in both cases are simple geometric sums. This leads us to the following bound on the quantization term

$$\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|x^{t-\tau_n} - \hat{x}^{t-\tau_n}\right\|^2\right] \leq \eta_g^2\phi(T)\sum_{t=0}^{T-2}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right]. \tag{47}$$

The unbiased case is a direct consequence of Corollary F.10, and the biased case is proven in Corollary F.12. As we can see, an unbiased server quantizer can help us prove a better bound on convergence.

F.4. Quantization lemmas

Lemma F.9 (Hidden state bound with an unbiased quantizer)**.** *Given an unbiased quantizer Q_s, for the iterations of QAFeL, it holds that*

$$\mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] \leq \eta_g^2\sum_{s=0}^{t-1}(1-\delta_s)^{t-s}\mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right]. \tag{48}$$

Proof. We prove the statement by induction on t. For $t = 0$, $x^t = \hat{x}^t$, so the statement holds. We assume the lemma is true for t, and prove for $t+1$. First, by definition of $\hat{x}^{t+1}$,

$$\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^{t+1}\right\|^2\right] = \mathbb{E}\left[\left\|x^{t+1} - \hat{x}^t - Q_s(x^{t+1} - \hat{x}^t)\right\|^2\right] \leq (1-\delta_s)\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^t\right\|^2\right]. \tag{49}$$

Then, by the definition of x^{t+1},

$$\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^t\right\|^2\right] = \mathbb{E}\left[\left\|x^t + \eta_g\overline{\Delta}^t - \hat{x}^t\right\|^2\right] = \mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] + \eta_g^2\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right], \tag{50}$$

where the last equality follows from the unbiasedness of Q_s. Therefore, plugging (50) into (49), we obtain

$$\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^{t+1}\right\|^2\right] \leq (1-\delta_s)\left(\mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] + \eta_g^2\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right]\right), \tag{51}$$

and using induction to substitute the first term concludes the proof,

$$\begin{aligned}\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^{t+1}\right\|^2\right] &\leq (1-\delta_s)\left(\eta_g^2\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] + \eta_g^2\sum_{s=0}^{t-1}(1-\delta_s)^{t-s}\mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right]\right) \\ &= \eta_g^2\sum_{s=0}^{t}(1-\delta_s)^{t+1-s}\mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right].\end{aligned} \tag{52}$$

□

Corollary F.10 (Sum of terms from Lemma F.9)**.** *From Lemma F.9, summing for $t = 0, \dots, T-1$, it follows that*

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] \leq \eta_g^2 \sum_{s=1}^{T-1} (1-\delta_s)^s \sum_{t=0}^{T-2} \mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right]. \tag{53}$$

Proof.

$$\begin{aligned}\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] &\leq \sum_{t=0}^{T-1}\sum_{s=0}^{t-1} (1-\delta_s)^{t-s} \eta_g^2 \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] = \sum_{s=0}^{T-2}\sum_{t=s+1}^{T-1} (1-\delta_s)^{t-s} \eta_g^2 \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] \\ &= \eta_g^2 \sum_{s=0}^{T-2} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] \sum_{t=s+1}^{T-1} (1-\delta_s)^{t-s} \leq \eta_g^2 \sum_{s=0}^{T-2} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] \sum_{t=1}^{T-1} (1-\delta_s)^t.\end{aligned} \tag{54}$$

□

Lemma F.11 (Hidden state bound with a general quantizer)**.** *Given any quantizer Q_s, for the iterations of QAFeL, it holds that*

$$\mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] \leq \eta_g^2 \frac{2}{\delta_s} \sum_{s=0}^{t-1} \left(1 - \frac{\delta_s}{2}\right)^{t-1-s} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right]. \tag{55}$$

Proof. The proof follows in a similar way to the unbiased case, by induction on t. For $t = 0$, $x^t = \hat{x}^t$, so the statement holds. We assume that the lemma is true for t, and prove for $t+1$. First, by definition of $\hat{x}^{t+1}$,

$$\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^{t+1}\right\|^2\right] = \mathbb{E}\left[\left\|x^{t+1} - \hat{x}^t - Q_s(x^{t+1} - \hat{x}^t)\right\|^2\right] \leq (1-\delta_s)\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^t\right\|^2\right]. \tag{56}$$

Similarly to (Koloskova et al., 2019) and (Gruntkowska et al., 2022), we now use the well-known inequality

$$\|a+b\|^2 \leq (1+\alpha)\|a\|^2 + (1+\alpha^{-1})\|b\|^2, \quad \forall \alpha > 0. \tag{57}$$

Applying the definition of x^{t+1} and the previous inequality yields

$$\begin{aligned}\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^{t+1}\right\|^2\right] &\leq (1-\delta_s)\mathbb{E}\left[\left\|x^{t+1} - \hat{x}^t\right\|^2\right] \\ &\leq (1-\delta_s)\left(1 + \frac{\delta_s}{2}\right)\mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] + (1-\delta_s)\left(1 + \frac{2}{\delta_s}\right)\eta_g^2 \mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] \\ &\leq \left(1 - \frac{\delta_s}{2}\right)\mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] + \frac{2}{\delta_s}\eta_g^2 \mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right],\end{aligned} \tag{58}$$

and using induction to substitute the first term concludes the proof. □

Corollary F.12 (Sum of terms from Lemma F.11)**.** *From Lemma F.11, summing for $t = 0, \dots, T-1$, it follows that*

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] \leq \eta_g^2 \frac{2}{\delta_s} \sum_{s=0}^{T-1} \left(1 - \frac{2}{\delta_s}\right)^s \sum_{t=0}^{T-2} \mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right]. \tag{59}$$

Proof.

$$\begin{aligned}\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|x^t - \hat{x}^t\right\|^2\right] &\leq \eta_g^2 \frac{2}{\delta_s} \sum_{t=0}^{T-1}\sum_{s=0}^{t-1} \left(1 - \frac{\delta_s}{2}\right)^{t-1-s} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] = \eta_g^2 \frac{2}{\delta_s} \sum_{s=0}^{T-2}\sum_{t=s+1}^{T-1} \left(1 - \frac{\delta_s}{2}\right)^{t-1-s} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] \\ &= \eta_g^2 \frac{2}{\delta_s} \sum_{s=0}^{T-2} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] \sum_{t=s+1}^{T-1} \left(1 - \frac{\delta_s}{2}\right)^{t-1-s} \leq \eta_g^2 \frac{2}{\delta_s} \sum_{s=0}^{T-2} \mathbb{E}\left[\left\|\overline{\Delta}^s\right\|^2\right] \sum_{t=0}^{T-1} \left(1 - \frac{\delta_s}{2}\right)^t.\end{aligned} \tag{60}$$

□

F.5. Bounding the server step

The goal of this section is to obtain the following bound on the server step

$$\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] \leq \left(\frac{2-\delta_c}{K}\right)\beta_P\left(\sigma_\ell^2+2G\right)+\left(1+\frac{1-\delta_c}{K}\right)P\sum_{p=0}^{P-1}(\eta_\ell^{(p)})^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right], \tag{61}$$

which is obtained using the same techniques as in the previous sections. We exploit three key properties: the unbiasedness of the client quantizer, the unbiasedness of the stochastic gradient, and the fact that the expected global gradient is the same as the expected local gradient (choosing clients uniforly at random). Each property is indicated when used.

Proof. First, we exploit the unbiasedness of the client quantizer, and we apply Lemma F.6 as follows

$$\begin{aligned}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] &= \mathbb{E}\left[\frac{1}{K^2}\left\|\sum_{k\in\mathcal{S}_t}Q_c\left(\sum_{p=0}^{P-1}\eta_\ell^{(p)}g_k(y_{k,p}^{t-\tau_k})\right)\right\|^2\right]\\ &\leq \frac{1}{K^2}\mathbb{E}\left[\left\|\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\eta_\ell^{(p)}g_k(y_{k,p}^{t-\tau_k})\right\|^2\right]+\frac{1-\delta_c}{K^2}\mathbb{E}\left[\sum_{k\in\mathcal{S}_t}\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}g_k(y_{k,p}^{t-\tau_k})\right\|^2\right].\end{aligned} \tag{62}$$

Then, we apply the unbiasedness of the stochastic gradient, and we add and subtract the expectation of our terms, $\pm\nabla F_k(y_{k,p}^{t-\tau_k})$, to obtain

$$\begin{aligned}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] \leq{}& \frac{1}{K^2}\mathbb{E}\left[\left\|\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\nabla F_k(y_{k,p}^{t-\tau_k})\right\|^2\right]+\frac{1-\delta_c}{K^2}\mathbb{E}\left[\sum_{k\in\mathcal{S}_t}\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}\nabla F_k(y_{k,p}^{t-\tau_k})\right\|^2\right]\\ &+\frac{1}{K}\beta_P\sigma_\ell^2+\frac{1-\delta_c}{K}\beta_P\sigma_\ell^2.\end{aligned} \tag{63}$$

Now, we use the fact that client k is sampled uniformly at random from all clients, and we can ensure

$$\mathbb{E}\left[\nabla F_k(y_{k,p}^{t-\tau_k})\right]=\mathbb{E}_{\mathcal{H}}\left[\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{F_n|n\sim[N]}\nabla F_n(y_{n,p}^{t-\tau_n})\right]=\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right]. \tag{64}$$

Thus, again applying the adding and subtracting technique, this time with $\pm\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})$,

$$\begin{aligned}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] \leq{}& \frac{1}{K^2}\mathbb{E}\left[\left\|\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]+\frac{1-\delta_c}{K^2}\mathbb{E}\left[\sum_{k\in\mathcal{S}_t}\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]\\ &+\frac{2-\delta_c}{K^2}\mathbb{E}\left[\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\left\|\eta_\ell^{(p)}\left(\nabla F_k(y_{k,p}^{t-\tau_k})-\frac{1}{N}\sum_{n=1}^{N}\nabla F_n(y_{n,p}^{t-\tau_n})\right)\right\|^2\right]\\ &+\frac{2-\delta_c}{K}\beta_P\sigma_\ell^2.\end{aligned} \tag{65}$$

Furthermore, applying the Cauchy-Schwarz and using the bounded gradient from assumption 3.3 to the third term yields

$$\begin{aligned}\mathbb{E}\left[\left\|\overline{\Delta}^t\right\|^2\right] \leq{}& \frac{1}{K^2}\mathbb{E}\left[\left\|\sum_{k\in\mathcal{S}_t}\sum_{p=0}^{P-1}\eta_\ell^{(p)}\frac{1}{N}\sum_{n=1}^{N}F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]+\frac{1-\delta_c}{K^2}\mathbb{E}\left[\sum_{k\in\mathcal{S}_t}\left\|\sum_{p=0}^{P-1}\eta_\ell^{(p)}\frac{1}{N}\sum_{n=1}^{N}F_n(y_{n,p}^{t-\tau_n})\right\|^2\right]\\ &+\frac{2-\delta_c}{K}\beta_P\left(\sigma_\ell^2+4G\right)\end{aligned} \tag{66}$$

Finally, using Cauchy-Schwarz to bound the norm of the sum gives us our desired bound. □