Network for AI: Communication-Efficient Federated Learning with MST-based Scheduling and Multi-Aggregation over Optical Networks

Ruikun Wang[1,2], Jiawei Zhang[1,*], Memedhe Ibrahimi[2], Zhiqun Gu[1], Yuming Xiao[3], Francesco Musumeci[2], Massimo Tornatore[2], and Yuefeng Ji[1,*]

[1]*State Key Lab of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing, China*
[2]*Politecnico di Milano, Milano, Italy* [3]*Purple Mountain Laboratories, Nanjing, China*
*{zjw, jyf}@bupt.edu.cn

Abstract: We propose a Minimum-Spanning-Tree-based scheduling and Multi-aggregation framework (MST-M) for communication-efficient Federated Learning. Simulation results show that MST-M saves over 10% in communication costs compared to existing heuristics.

1. Introduction

Artificial Intelligence (AI) is currently revolutionizing several scientific fields, such as computer vision and natural language processing, and today optical networks are the main communication platform to interconnect the large and distributed AI models required in these fields. While in the last years a broad set of novel AI applications have emerged in optical networks [1, 2], lately there is significant shift of attention towards the opposite paradigm, i.e., how optical networks can provide effective interconnection across large and distributed AI models (a.k.a "network for AI") [3]. A prominent field of application of the "network for AI" paradigm is Federated Learning (FL), a subcategory of (typically, privacy-preserving) machine learning using decentralized model training. In traditional synchronous FL, the AI model is initialized in a central node (i.e., global model), and broadcasted to distributed nodes (i.e., local models) for local training. Then, the trained local models are uploaded to the central node for aggregation, which begins once each local model is transmitted to the central node. The above process is iterated until the global model converges. During FL training, communication costs generally include two main contributions: 1) Resource Costs (RC) that arise from communication resource consumption due to model broadcasting and uploading, and 2) Latency Costs (LC) that account for communication time, local training time and aggregation time. Since the number of local models can be large, to ensure minimal communication costs, it is crucial to develop efficient scheduling strategies. Scheduling for FL training has recently attracted research interest, but existing literature usually considers single and two-stages aggregation [4-6]. In single-aggregation, global and local models may traverse redundant paths, causing a waste of resources and high latency. In multi-aggregation, local models can also be aggregated at intermediate nodes (i.e., not only in the central node), and leveraging appropriate scheduling strategies, it is possible to effectively decrease latency and resource consumption. Therefore, to minimize communication costs (both RC and LC), it is necessary to design scheduling strategies for model communication and aggregations.

In this study, we propose a Minimum-Spanning-Tree-based scheduling and Multi-aggregation framework (MST-M) for communication-efficient FL over optical networks. We show that our proposed MST-M outperforms existing solutions based on shortest-path and first-fit (SPFF) in both single- and multi- aggregation operations.

2. Network Scenario and Communication Cost Models

Fig. 1(a) shows an example of a FL request over an optical network. Each FL request has a global model (placed in a central node) and several local models (placed in distributed nodes with local datasets). We consider that the optical network supports broadcasting and uploading of the FL models' parameters over multi-wavelength links. Each node is equipped with computing resources for training local models and for aggregation operations, i.e., aggregating local models arriving from other nodes. Fig. 1(b) shows an example of communication (i.e., broadcasting and uploading) and aggregation operations to gather parameters from three local models. We consider four ways for broadcasting (i.e., A ~ D) and uploading (i.e., E ~ H), respectively. For instance, in broadcasting, global model θ is transferred to the node hosting local model 1, and then passed from current node to other nodes hosting local models 2 and 3 (case C). The uploading process includes communication and multiple aggregation operations, e.g., local model θ_2 will be aggregated with the local model θ_1 before it is transmitted to the central node as $\theta_{1,2}$ (case F). Thus, RC is defined as the number of activated wavelengths and consumed bandwidth in broadcasting and uploading. We assume that model θ and θ_i have same size and consume same bandwidth, since each model contains the same number of trained parameters. Fig. 1(c) shows the latency model, in which broadcasting and uploading happen as in cases of C and F in

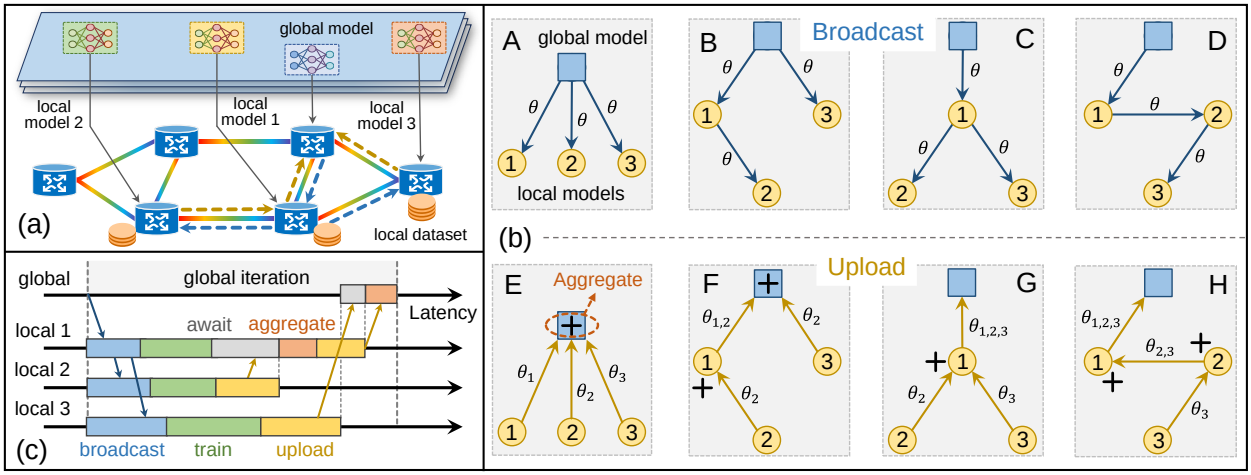

Fig. 1 (a): FL requests over optical networks; (b) Communication and aggregation models; (c) Latency models.

Fig. 1(b), respectively. The completion latency consists of communication latency for broadcasting and uploading, local training latency, aggregating latency and awaiting latency. For l^{th} local model in FL request s, communication latency ($t_{s,l}^{com}$) accounts for propagation time, transmission time, processing and queuing time; local training latency ($t_{s,l}^{train}$) depends on the number of local epochs (k_s), number of samples ($d_{s,l}$), needed computing resource for one sample (n_s) and available computing resource ($c_{s,l}$), i.e., $t_{s,l}^{train} = k_s \cdot n_s \cdot d_{s,l}/c_{s,l}$; aggregation time ($t_{s,l}^{agg}$) relies on the number of aggregating models ($a_{s,l}$), needed computing resource for one model (m_s) and available computing resource ($c_{s,l}$), i.e., $t_{s,l}^{agg} = a_{s,l} \cdot m_s/c_{s,l}$; awaiting latency ($t_{s,l}^{await}$) is caused by synchronous and awaiting operations. Thus, LC of s^{th} FL request equals $T_s = \max_l \left[t_{s,l}^{com} + t_{s,l}^{train} + t_{s,l}^{agg} + t_{s,l}^{await}\right]$. The communication cost equals $C = \alpha \cdot RC + (1-\alpha) \cdot \sum_s T_s$, where α is a coefficient to balance the importance of RC and LC.

3. MST-based Scheduling and Multi-Aggregation

We propose the novel MST-M framework to optimize FL communication costs over optical networks. The main steps of MST-M are shown in Fig. 2(a). *Step 1*: we build two auxiliary graphs for broadcasting and uploading, that we name *broadcast graph* and *upload graph*, respectively. *Step 2*: each edge $(i,\ j)$ in broadcast graph is initialized according to its communication cost with a weight equal to $w_{i,j} = \alpha \cdot R_{i,j}/R^{max} + (1-\alpha) \cdot T_{i,j}^{com}/T^{max}$, where $R_{i,j}$ and $T_{i,j}^{com}$ denote resource cost and communication time of edge $(i,\ j)$, R^{max} and T^{max} represent the maximum resource cost and communication time among all edges, respectively, and α is the same as in Section 2. *Step 3*: we find the MST based on the broadcast graph. Fig. 2(b) shows an example on how to find an MST, where we place the central node with the global model into set U, and sort distributed nodes with local models in descending order of local training times, and place distributed nodes into set V. Afterward, an MST is searched between set U and set V, where the searching order is consistent with the order in which nodes are placed in set V. The node moves from set V to set U after each searching step. The above searching steps end when set V is empty. The searched MST is the routing results in broadcasting. We use first fit scheme for wavelength assignment and subsequently assess whether the allocated paths have sufficient bandwidth. *Step 4*: upload graph is initialized with same equation in step 2, but it should be run after step 3 as this step changes RC. *Step 5*: we find an MST in upload graph, where distributed nodes with local models are ordered in descending ordering based on local training time and communication time in broadcasting.

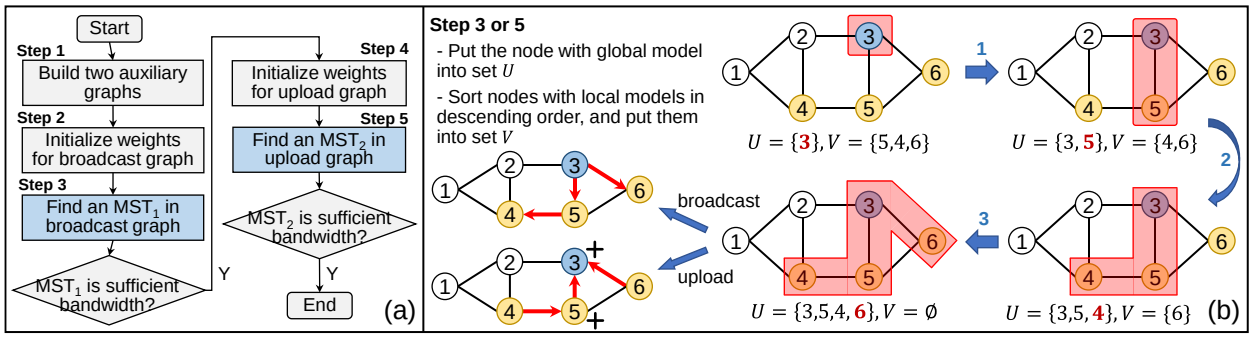

Fig. 2 (a): Flow-chart of the proposed solution; (b) Illustration of finding MST in broadcast or upload graphs.

Meanwhile, multi-aggregation operations are decided based on the searched MST. For example, as shown in Fig. 2(b), when the MST is [4→5, 5→3, 6→3], aggregation operations occur at node 5 and node 3. Meanwhile, the searched results undergo wavelength assignment and bandwidth sufficiency checks. Note that MSTs will be found in broadcasting and uploading according to its auxiliary graph, respectively. Finally, double MSTs in broadcasting and uploading are routing and aggregation results for current FL request.

4. Simulation Setup and Results

We consider an optical city network with 30 nodes [7] with link lengths ranging between 20 km and 60 km. Each link consists of 30 wavelengths (each with 25 Gbps capacity). For each node, its processing and queuing time is 10 μs, and it provides computing resources between 5 and 20 Giga operations per second (GOPS) for each global model or local model. The bandwidth for each FL request ranges between 1 Gbps and 5 Gbps, and the number of local epochs varies between 3 and 10. Moreover, the number of local models varies between 6 and 16, and for each of them the number of local samples is selected from 100 to 1000, where the locations of global model and local models are randomly selected. Meanwhile, the computing resource for training one sample ranges from 0.01 to 0.05 GOPS, while the computing resource for aggregating one model varies from 0.01 to 0.1 GOPS. In this work, we consider SPFF-based Multi-aggregation (SPFF-M) and SPFF-based Single-aggregation (SPFF-S) as benchmarks.

Fig. 3(a) shows the normalized RC and average LC under different values of α, where the number of FL requests is 40. When α is set to 0, the objective is only to minimize LC, while the objective is only to minimize RC with α is set to 1. We observe that RC and LC show opposite trends as values of α increase, since value of α affects the weights of RC and LC in initializing auxiliary graphs. Coefficient α is set to 0.5 in the following experiments. Fig. 3(b) presents the normalized RC and average LC with different number of local models, where the number of FL requests is 40. Results show that communication costs increase with the number of local models. The reason is that the number of routing paths and aggregation times increase with the number of local models. In addition, MST-M is more communication-efficient compared to baselines, as it reduces average latency by 23.59% and 16.28% compared to baselines. In Fig. 3(c), we show the normalized RC and average LC for varying number of FL requests, in which for each FL request the number of local models ranges from 6 to 16. The results show that MST-M saves 12.48% and 50.07% in RC, on average, compared to SPFF-M and SPFF-S, respectively. Besides, MST-M reduces LC by 12.45% and 14.15% compared to baselines. This is because MST-M searches globally to reduce redundant paths and latency.

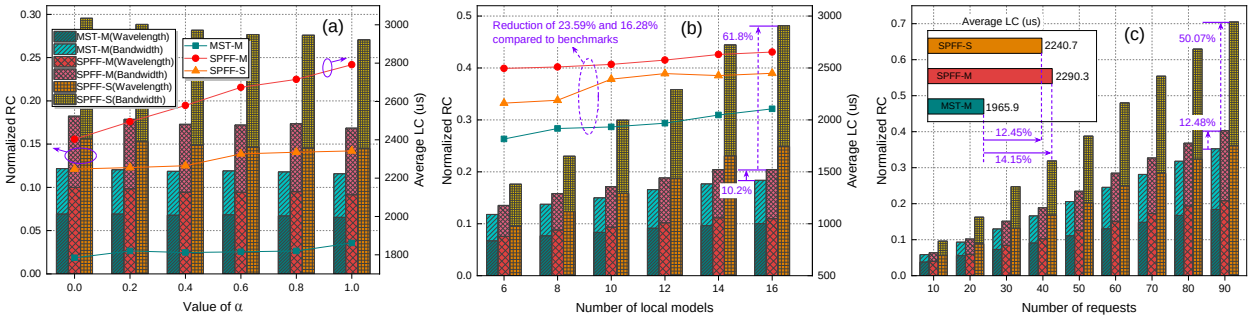

Fig. 3 (a): Normalized RC and average LC vs. values of α; (b) Normalized RC and average LC vs. number of local models; (c) Normalized RC and average LC vs. number of requests.

5. Conclusions

We devise a new Minimum-Spanning-Tree-based scheduling and Multi-aggregation framework (MST-M) for communication-efficient FL over optical networks. Simulation results show that MST-M reduces the communication costs between 10.2% and 61.8% compared to SPFF-based benchmarks under different number of local models and number of requests.

Acknowledgement: This work was supported by the NSFC project (62271078), the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"), Fund of State Key Laboratory of IPOC (BUPT) (No.IPOC2022ZT11), P.R. China., and the Fundamental Research Funds for the Central Universities (No.2023RC15).

References

[1] J. Zhang, et al, JOCN, vol. 15, no. 2, pp. A63-A73, 2023.
[2] R. Wang et al., JOCN, vol. 15, no. 7, pp. C88-C99, 2023.
[3] K. B. Letaief, et al, IEEE JSAC, vol. 40, no. 1, pp. 5-36, 2022.
[4] J. Li, et al, OFC, San Francisco, CA, USA, 2021.
[5] S. Huang, et al, IEEE IoT, vol. 10, no. 3, pp. 2056-2070, 2023.
[6] X. Zhou, et al, IEEE TVT, vol. 70, no. 6, pp. 5308-5317, 2021.
[7] Y. Xiao, et al, JLT, vol. 39, no. 17, pp. 5347-5361, 2021.