Hierarchical Scheduling for Federated Learning Optimization over Deterministic Satellite Networks

Hao Deng*, Hao Yu*, Qize Guo†, Tarik Taleb†
*Hangzhou International Innovation Institute, Beihang University, Hangzhou, China
†Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum, Germany
Email: {denghaoAI, haoyu1}@buaa.edu.cn, {qize.guo, tarik.taleb}@rub.de

***Abstract*—Federated learning (FL) enables collaborative model training in low Earth orbit (LEO) satellite networks without exchanging raw sensing data. However, satellite mobility, limited onboard resources, and time-sensitive model-parameter transmission make FL scheduling challenging. This paper studies FL completion time minimization in a cyclic queuing and forwarding (CQF)-enabled deterministic satellite network by jointly formulating aggregation node selection, routing, and CQF offset scheduling. To address this problem, we propose a hierarchical Graph Attention Transformer-Proximal Policy Optimization (GAT-PPO) framework, where a high-level agent selects aggregation nodes and a low-level agent performs packet-level routing and offset control. Graph attention networks are used to encode dynamic satellite topology, and PPO is adopted for policy training. Simulation results show that the proposed method reduces FL completion time and maintains a high task success rate compared with OSPF and greedy heuristic schemes.**

***Index Terms*—Deterministic Satellite Network, Federated Training, Cyclic Queuing Forwarding, Remote Sensing**

I. Introduction

SATELLITE computing has been increasingly investigated in both academia and industry due to its potential to provide ubiquitous computing and data processing capabilities in space [1]–[5]. With the continuous improvement of onboard computing capabilities, data collected by satellites can be directly processed in orbit rather than being transmitted entirely to ground stations. This paradigm can significantly reduce bandwidth consumption and transmission latency, enabling latency-sensitive applications such as disaster monitoring and real-time environmental observation.

Many emerging satellite applications require collaborative intelligence among distributed satellites while limiting raw-data exchange. Federated learning (FL) provides a promising solution by allowing satellites to train local models with their own sensing data and exchange only model updates [6]. For example, in remote sensing applications, satellites can independently train disaster detection models using locally collected images and collaboratively obtain a global model through parameter aggregation. Such a paradigm reduces the communication overhead of transmitting large-volume remote sensing data while preserving data privacy.

However, efficient FL over satellite networks requires timely and reliable transmission of model updates throughout multiple training rounds. Satellite mobility, time-varying inter-satellite links (ISLs), limited onboard resources, and concurrent FL traffic make conventional best-effort transmission insufficient for providing predictable communication performance. Deterministic networking techniques, such as Time-Sensitive Networking (TSN) and Cyclic Queuing and Forwarding (CQF), provide bounded forwarding behavior and therefore offer a potential solution for time-sensitive FL traffic [7]–[9].

In this paper, we investigate FL completion time minimization over CQF-enabled deterministic LEO satellite networks. The problem involves two tightly coupled decision levels: selecting suitable aggregation nodes for FL tasks and scheduling model-update flows through dynamic satellite networks. We therefore jointly optimize aggregation node selection, routing, and CQF offset scheduling, and propose a hierarchical Graph Attention Transformer-Proximal Policy Optimization (GAT-PPO) framework. The high-level agent selects aggregation nodes based on global network and computing states, while the low-level agent jointly determines packet-level routing and CQF offsets. The main contributions of this paper are three-fold: 1) We formulate the joint aggregation node selection, routing, and CQF offset scheduling problem for minimizing FL completion time in dynamic deterministic LEO satellite networks. 2) We propose a hierarchical GAT-PPO framework that separates task-level aggregation scheduling from packet-level deterministic forwarding while coordinating the two decision levels. 3) Simulation results demonstrate that the proposed method reduces FL completion time and maintains a high task success rate compared with OSPF and greedy heuristic schemes.

II. Related Work

Satellite federated learning has recently received increasing attention. Ground-assisted FL scheduling was investigated in [1], while satellite federated edge learning architectures and their convergence properties were studied in [2]. FedSN [3] considered heterogeneous LEO satellite networks, and FedSpace [10] developed an FL framework across satellites and ground stations. More general architectures and applications of FL in satellite constellations were discussed in [4], [5]. These studies mainly focus on FL architecture, participant coordination, convergence, or communication efficiency. In contrast, our work explicitly considers the deterministic transmission of FL model updates and jointly couples aggregation-node selection with packet-level routing and scheduling.

Deterministic networking has also been investigated for time-sensitive satellite communications. Wang *et al.* [7] stud-

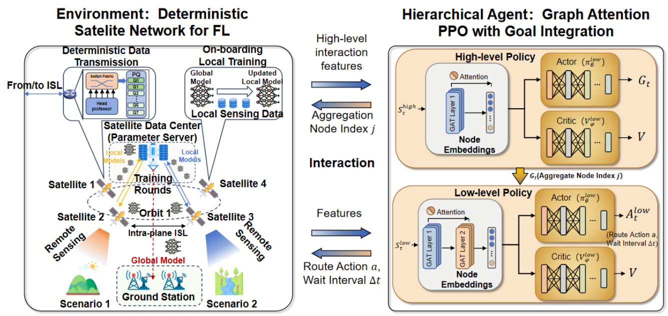

Fig. 1: Federated Model Training Over Deterministic Satellite Network for Remote Sensing Application

ied deterministic flow scheduling in LEO satellite networks, while CPF [9] explored the extension of time-sensitive networking mechanisms to large-scale LEO constellations. TSN mechanisms and their capability of providing predictable communication performance have also been extensively studied in terrestrial and wireless environments [8], [11]. These works primarily optimize deterministic communication flows independently of distributed learning tasks.

Different from the above studies, we consider the interaction between FL computation and deterministic communication. Specifically, the aggregation-node decision determines both the location of model aggregation and the communication endpoints of model-update flows, while routing and CQF offsets determine their transmission delays. Accordingly, our hierarchical GAT-PPO framework jointly optimizes aggregation node selection, routing, and CQF offset scheduling toward the end-to-end FL completion-time objective rather than optimizing FL coordination or deterministic networking separately.

III. System Model and Problem Formulation

We consider a representative scenario in which a multi-orbit Low Earth orbit (LEO) satellite constellation collaboratively performs model training while maintaining deterministic communication, as shown in Fig.1. In this section, we present the system model for federated learning (FL) over a CQF-enabled deterministic LEO satellite networks, and finally formulate the joint aggregation node selection and flow scheduling problem.

A. Dynamic LEO Satellite Network Model

We consider a LEO satellite constellation consisting of O orbital planes, where each orbital plane contains S satellites. Each satellite is equipped with onboard computing resources and inter-satellite communication capability. Similar to a typical Walker constellation, a satellite maintains intra-plane Inter-Satellite Links (ISLs) with neighboring satellites in the same orbital plane and inter-plane ISLs with satellites in adjacent orbital planes. Due to orbital motion, the availability and propagation delay of ISLs vary over time. The system time is divided into a sequence of discrete time slots. During each time slot, the network topology is assumed to remain unchanged. At time slot t, the satellite network is modeled as a dynamic graph $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t)$, where $\mathcal{E}^t$ denotes the set of available ISLs. Each link $e_{i,j}^t \in \mathcal{E}^t$ is characterized by its available bandwidth, propagation delay, physical distance, and link availability. Each satellite $v_i \in \mathcal{V}$ has limited onboard computing capability, memory, buffer capacity, and energy budget.

B. CQF-Enabled Deterministic Forwarding Model

The satellite contains an onboard TSN switch with input buffers, a packet header processor, a switching fabric, and transmission buffers associated with outgoing ISLs. Regular traffic is stored in best-effort queues, while time-sensitive FL traffic is mapped to CQF queues. Through alternating receive-transmit cycles, CQF provides predictable forwarding behavior and bounded delay for model-parameter flows.

In this paper, local model updates, aggregation results, and FL control packets are regarded as time-sensitive flows. These packets require bounded delay, because delayed model updates may block the aggregation process and increase the overall FL completion time. Let Δ denote the CQF cycle length. For each outgoing interface, two CQF queues are maintained. At each cycle, one queue receives packets while the other transmits packets. Their roles are exchanged in the next cycle. This cyclic receive-transmit mechanism provides bounded queuing behavior and avoids unbounded waiting for time-sensitive packets.

For a scheduled packet, the single-hop delay over link $e_{i,j}^t$ consists of propagation delay, transmission delay, packet processing delay, and CQF-induced waiting delay:

$$D_{k,i,j}^{\text{hop}} = D_{i,j}^{\text{prop}} + D_{k,i,j}^{\text{trans}} + D_i^{\text{proc}} + D_{k,i,j}^{\text{CQF}}. \tag{1}$$

Here, $D_{i,j}^{\text{prop}}$ is determined by the physical distance between satellites, $D_{k,i,j}^{\text{trans}}$ depends on packet size and link bandwidth, D_i^{proc} is the packet processing delay, and $D_{k,i,j}^{\text{CQF}}$ is the waiting delay caused by CQF offset decision (in terms of the number of CQF cycles.)

For a deterministic flow f_k routed along path $P_k = \{v_k^0, v_k^1, \ldots, v_k^{H_k}\}$, where v_k^0 and $v_k^{H_k}$ are the source and destination nodes, respectively, the end-to-end delay is given by

$$D_k^{\text{e2e}} = \sum_{h=0}^{H_k-1} D_{k,v_k^h,v_k^{h+1}}^{\text{hop}}. \tag{2}$$

The routing and CQF scheduling decisions are coupled. A selected route determines which ISLs are used, while the CQF offset determines the forwarding cycle at each hop. A feasible schedule must satisfy link availability, bandwidth limitation, finite queue capacity, CQF cycle ordering, and end-to-end delay requirements.

C. Federated Learning Task Model

We consider a set of FL tasks generated by satellite computing applications. The set of FL tasks is denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$. Each task $m \in \mathcal{M}$ is collaboratively trained by a subset of worker satellites $\mathcal{S}_m \subseteq \mathcal{V}$. Each worker satellite $v_i \in \mathcal{S}_m$ owns a local dataset $\mathcal{D}_{i,m}$. Each worker trains a local model using its onboard data and transmits only model updates through the satellite network. An FL task m is described by $\Gamma_m = \{\mathcal{S}_m, \mathcal{A}_m, R_m, E_m, L_m^u, L_m^d\}$, where $\mathcal{A}_m$ is the candidate set of aggregation nodes, R_m is the number of global training rounds, E_m is the number of local epochs per round, L_m^u is the size of the uploaded local model update, L_m^d is the size of the downlink global model.

For each FL task, one satellite is selected as the aggregation node. Let $y_{a,m} \in \{0,1\}$ denote whether satellite $v_a \in \mathcal{A}_m$ is selected as the aggregation node for task m. Exactly one aggregation node should be selected for each task.

During each global round, every worker first performs local training. The local training time of worker v_i for task m is modeled as

$$T_{i,m}^{\mathrm{loc}} = \frac{E_m \xi_m |\mathcal{D}_{i,m}|}{C_i}, \tag{3}$$

where ξ_m denotes the computing workload required to process one data sample, and C_i is the onboard computing capability of satellite v_i.

Once all required updates are received by aggregation node after local training, the aggregation node performs model aggregation. The aggregation time at node v_a is modeled as

$$T_{a,m}^{\mathrm{agg}} = \frac{\zeta_m |\mathcal{S}_m|}{C_a}, \tag{4}$$

where ζ_m denotes the computation workload required to aggregate one local update. After aggregation, the global model is distributed back to the participating worker satellites.

Therefore, in each FL round, the uplink and downlink flows need to be routed and scheduled through the CQF-enabled satellite network. For task m in round r, let $D_{i,a,m}^{r,u}$ denote the end-to-end delay of the uplink model update from worker v_i to aggregation node v_a, and let $D_{a,i,m}^{r,d}$ denote the end-to-end delay of the downlink global model from v_a to worker v_i.

The completion time of round r is determined by the slowest local training and uplink transmission process, the aggregation time, and the slowest downlink transmission process:

$$T_m^r = \max_{v_i \in \mathcal{S}_m} \left(T_{i,m}^{\mathrm{loc}} + D_{i,a,m}^{r,u}\right) + T_{a,m}^{\mathrm{agg}} + \max_{v_i \in \mathcal{S}_m} D_{a,i,m}^{r,d}. \tag{5}$$

The total completion time of task m is $T_m^{\mathrm{total}} = \sum_{r=1}^{R_m} T_m^r$. This formulation captures the synchronization nature of FL. The aggregation node cannot start model aggregation until all required uplink updates are received, and a global round is not completed until all participating workers receive the updated global model.

D. Problem Formulation

The objective of this paper is to minimize the total FL completion time by jointly optimizing aggregation node selection, routing, and CQF offset scheduling. The problem can be formulated as

$$\min_{\mathbf{y},\mathbf{P},\mathbf{o}} \quad \sum_{m \in \mathcal{M}} T_m^{\mathrm{total}}, \tag{6}$$

where $\mathbf{y}$ denotes the aggregation node selection variables, $\mathbf{P}$ denotes the routing paths of FL-related deterministic flows, and $\mathbf{o}$ denotes the CQF offset decisions. The optimization is subject to the following constraints. First, each FL task must select exactly one aggregation node from its candidate set. Second, each deterministic flow must be routed through available ISLs in the current dynamic topology. Third, the traffic assigned to each link cannot exceed its bandwidth capacity, and the buffered data cannot exceed the CQF queue capacity. Fourth, the selected aggregation nodes and worker satellites must satisfy onboard computation constraints. Finally, the CQF offset decisions must follow the receive-transmit cycle ordering of CQF, and the end-to-end delay of each deterministic flow should satisfy its deadline requirement.

This problem is NP-hard because aggregation node selection, routing, and CQF offset scheduling are tightly coupled. Moreover, satellite mobility causes time-varying topology and link delay, making static optimization difficult. Therefore, we adopt a hierarchical reinforcement learning framework to solve the problem as discussed in the next section.

IV. Hierarchical GAT-PPO Algorithm

A. Overview

To reduce the complexity of the joint optimization problem, we presents a hierarchical GAT-PPO algorithm, as shown in Fig.1. The high-level agent selects the aggregation node for each FL task according to the global satellite state, while the low-level agent performs packet-level forwarding by jointly determining the next-hop satellite and the CQF offset. Both agents adopt graph attention networks (GATs) to encode the dynamic satellite topology. The high-level GAT captures global topology, computing, and queue information for aggregation node selection, while the low-level GAT extracts forwarding-related features for CQF-aware routing. The two policies are trained using proximal policy optimization (PPO).

B. Hierarchical Markov Decision Process

The proposed method formulates the scheduling problem as a hierarchical Markov decision process (MDP). At time slot t, the satellite network is represented by the dynamic graph $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t)$, and the node feature matrix is denoted by $\mathbf{X}^t$.

1) High-Level Decision: For each FL task m, the high-level agent observes the global network state and selects one aggregation node from the candidate set $\mathcal{A}_m$. The high-level action is defined as $a_m^H \in \mathcal{A}_m$. The selected aggregation node determines the destinations of uplink model-update flows and the sources of downlink global-model flows. Since the aggregation node affects both computation delay and communication delay, this decision has a direct impact on the total FL completion time.

2) Low-Level Decision: After the aggregation node is selected, the low-level agent schedules packet forwarding for FL-related deterministic flows. For each packet p, the low-level action is defined as $a_p^L = \left(a_{p,r}^L, a_{p,o}^L\right)$, where $a_{p,r}^L$ denotes the next-hop satellite and $a_{p,o}^L$ denotes the CQF offset. The next-hop action should be selected within its neighbor $\mathcal{N}^t(v_p)$

at time slot t. The offset action is selected from a discrete set: $a_{p,o}^L \in \{0, 1, \ldots, O_{\max}\}$. Thus, the low-level agent jointly controls the spatial forwarding path and the temporal CQF scheduling offset.

Algorithm 1 Hierarchical GAT-PPO Training Procedure

Require: Episodes $E_{\max}$, FL rounds R, update frequency F
Ensure: Policies π_H and π_L
Initialize high-level and low-level actor-critic networks
Initialize rollout buffers $\mathcal{B}_H$ and $\mathcal{B}_L$
for episode $e = 1$ to $E_{\max}$ **do**
  Update topology $\mathcal{G}^t$ and generate FL tasks $\mathcal{M}$
  Select aggregation node for each task using π_H
  for each FL round and each transmission phase **do**
    while active packets exist **do**
      Select next hop and CQF offset for each packet using π_L
      Execute forwarding actions and store low-level transitions
      Update queues, delays, packet states, and topology
    end while
  end for
  Compute FL completion time and high-level rewards
  Update low-level networks using PPO
  if $e \bmod F = 0$ **then**
    Update high-level networks using PPO
  end if
end for

C. GAT-Based Policy Networks

1) High-Level Policy Network: The high-level policy network uses a GAT encoder to extract topology-aware node embeddings: $\mathbf{H}_H^t = \mathrm{GAT}_H(\mathbf{X}^t, \mathcal{E}^t)$. The actor then maps the node embeddings to aggregation-node logits. Invalid nodes, including non-candidate nodes and task source nodes, are masked before the softmax operation. The high-level policy is given by

$$\pi_H(a_m^H = i | s_m^H) = \frac{\exp(z_i^H + \mu_i^H)}{\sum_{j\in\mathcal{V}} \exp(z_j^H + \mu_j^H)}, \tag{7}$$

where z_i^H is the logit of node i, and μ_i^H is a mask term. Specifically, $\mu_i^H = 0$ if node i is a feasible aggregation node, and $\mu_i^H = -\infty$ otherwise.

The critic estimates the value of the high-level state. The node embeddings are first pooled into a graph-level representation:

$$\bar{\mathbf{h}}_H^t = \frac{1}{|\mathcal{V}|} \sum_{i\in\mathcal{V}} \mathbf{h}_{H,i}^t, \tag{8}$$

and the state value is computed as $V_H(s_m^H) = f_H^{\text{critic}}(\bar{\mathbf{h}}_H^t)$.

2) Low-Level Policy Network: The low-level policy network also adopts a GAT encoder to represent the current satellite topology:$\mathbf{H}_L^t = \mathrm{GAT}_L(\mathbf{X}^t, \mathcal{E}^t)$. For each packet, the node embeddings are combined with packet-specific information, including the current node and the target node. The low-level actor contains two output heads. The routing head generates the next-hop distribution, while the offset head generates the CQF offset distribution.

The joint low-level policy is factorized as

$$\pi_L(a_p^L | s_p^L) = \pi_r(a_{p,r}^L | s_p^L)\pi_o(a_{p,o}^L | s_p^L). \tag{9}$$

The low-level critic estimates the expected return of forwarding packet p from its current position to the target:

$$V_L(s_p^L) = f_L^{\text{critic}}(s_p^L). \tag{10}$$

D. Reward Design

The reward design follows the hierarchical structure of the proposed framework. The high-level reward evaluates the overall FL task completion performance, while the low-level reward provides step-wise guidance for packet forwarding.

For FL task m, the high-level reward is defined as

$$R_m^H = \eta_s \mathbb{I}_m^{\text{succ}} - \beta T_m^{\text{total}} - \eta_f \mathbb{I}_m^{\text{fail}}, \tag{11}$$

where T_m^{total} is the total completion time of task m, $\mathbb{I}_m^{\text{succ}}$ and $\mathbb{I}_m^{\text{fail}}$ indicate whether the task succeeds or fails, respectively. The parameters η_s, β, and η_f control the success reward, delay penalty, and failure penalty.

For the low-level agent, the reward of packet p after one forwarding step is defined as

$$r_p^L = \lambda_d \left(D_p^{\text{before}} - D_p^{\text{after}}\right) - \lambda_t D_p^{\text{hop}} - \lambda_o (a_{p,o}^L)^3 + R_p^{\text{term}}, \tag{12}$$

where D_p^{before} and D_p^{after} are the distances from packet p to its target before and after forwarding, respectively. D_p^{hop} is the single-hop delay, and $(a_{p,o}^L)^3$ penalizes excessive CQF offset selection. The terminal reward R_p^{term} is positive if the packet reaches its target and negative if the packet is dropped or exceeds the maximum forwarding steps.

This reward design encourages the high-level agent to select aggregation nodes that reduce FL completion time, while guiding the low-level agent to forward packets toward their destinations with low delay and limited queue waiting.

E. Training Procedure

Both agents are trained using PPO. For a policy π_θ, the probability ratio is defined as

$$\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \tag{13}$$

The clipped PPO objective is

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min\left(\rho_t(\theta)\hat{A}_t, \mathrm{clip}\left(\rho_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{14}$$

where $\hat{A}_t$ is the advantage estimate and ϵ is the clipping parameter. The final loss contains the policy loss, value loss, and entropy regularization:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}^{\text{CLIP}}(\theta) - c_v \left(V_\phi(s_t) - \hat{R}_t\right)^2 + c_e \mathcal{H}(\pi_\theta), \tag{15}$$

where c_v and c_e are the value-loss and entropy coefficients.

The overall training procedure is summarized in Algorithm 1. At the beginning of each episode, the simulator updates the satellite topology and generates concurrent FL tasks. The high-level agent first selects an aggregation node for each task. Then, the low-level agent schedules uplink and downlink packet transmissions by selecting next-hop nodes and CQF offsets. After the FL tasks are completed or terminated, rewards are computed and both agents are updated using PPO.

V. Performance Evaluation

A. Simulation Setup

The proposed hierarchical GAT-PPO framework is evaluated using a Python-based LEO satellite network simulator–StarPerf [12]. The experiments are conducted on a Mac mini 2024 with an Apple M4 Pro chip and 64 GB memory. StarPerf 2.0.0 is used to generate the dynamic 1) Boeing; 2) OneWeb; 3) Telesat satellite topology, while PyTorch 2.10.0 and PyTorch Geometric 2.7.0 are used to implement the GAT-based PPO agents. In the simulation, satellites are modeled as graph nodes and inter-satellite links are modeled as dynamic edges. The maximum ISL distance is set to 5000 km, the link bandwidth is set to 20 Mbps, and the maximum queue capacity is set to 90 packets. The topology is updated every 60 s, and the maximum CQF offset level is set to 5 cycles. For each episode, 15–25 FL tasks are randomly generated, and each task contains 2–4 worker satellites. The number of candidate aggregation nodes is set to 10. Each task performs 3 global communication round with 3 local epochs. The model parameter size is set to 30 Kb, and both the local training complexity and aggregation complexity are set to 150 units. Both high-level and low-level agents use GAT-based actor-critic networks with a hidden dimension of 64. The high-level and low-level learning rates are set to 1×10^{-4} and 3×10^{-4}, respectively. The discount factor is 0.99, the PPO clipping parameter is 0.1, and each PPO update uses 4 epochs. The main evaluation metric is the FL task completion time, including local training, uplink transmission, aggregation, and downlink transmission delays.

B. Results and Analysis

1) Training Convergence: Fig. 2 shows the training reward curves of the proposed hierarchical GAT-PPO framework under different satellite constellations. The upper subfigure presents the low-level reward, while the lower subfigure presents the high-level reward. As shown in the figure, the low-level reward gradually increases from a large negative value and converges to a stable region close to zero. This indicates that the low-level agent progressively learns effective packet-level forwarding decisions, including next-hop selection and CQF offset control. In the early training stage, packets are frequently routed through congested links or invalid paths, leading to large delay penalties and packet-drop penalties. As training proceeds, the agent learns to reduce forwarding delay and improve packet delivery, resulting in higher low-level rewards.

The high-level reward also shows an overall increasing trend, although it fluctuates more significantly than the low-level reward. This is because the high-level agent receives sparse task-level feedback only after the completion or failure of an FL task. The selected aggregation node affects the overall uplink delay, aggregation delay, and downlink delay, making the high-level reward more sensitive to topology variation and low-level routing performance. After sufficient training episodes, the high-level reward becomes stable, demonstrating that the high-level agent learns to select aggregation nodes with better communication and computation conditions. Overall, the convergence behavior verifies that the proposed hierarchical training process can jointly optimize macro-level aggregation scheduling and micro-level deterministic forwarding.

2) Impact of Worker Number per Task: Fig. 3 evaluates the impact of the number of worker satellites per FL task. Increasing the number of workers increases the amount of uplink model updates and downlink global model transmissions. As a result, the communication burden around the aggregation node becomes heavier, and the scheduling problem becomes more challenging. The proposed method maintains a 100% success rate across different worker numbers, showing strong robustness under increasing task scale. Although its delay increases as more workers participate in each task, the growth is smoother than that of the greedy heuristic. The greedy method also preserves a high success rate, but its delay becomes significantly larger when the worker number is high. This is because greedy decisions are based on local distance and queue information, which may lead to suboptimal aggregation and routing choices under dense traffic.

Traditional OSPF shows lower delay in some high-load cases, but this result should be interpreted together with its success rate. As the number of workers increases, the success rate of OSPF drops sharply, indicating that many packets or tasks fail due to congestion and lack of deterministic scheduling control. Therefore, the apparent delay advantage of OSPF under heavy load is achieved at the cost of reliability. In contrast, the proposed method provides stable task completion while keeping the delay lower than the greedy heuristic in most high-load scenarios.

3) Impact of Concurrent FL Tasks: Fig. 4 compares the performance of the proposed method with traditional OSPF and the greedy heuristic under different numbers of concurrent FL tasks. As the number of FL tasks increases, the traffic load in the satellite network becomes heavier, resulting in longer transmission delay and more severe queue congestion.

The proposed method achieves the lowest or near-lowest delay in most cases while maintaining a stable success rate of 100%. This is because the high-level agent can distribute FL tasks among suitable aggregation nodes, and the low-level agent can adapt routing and CQF offset decisions according to dynamic queue states. In contrast, traditional OSPF only considers shortest-path routing and does not explicitly account for CQF scheduling or congestion. Therefore, although OSPF can achieve relatively low delay under light traffic load, its success rate decreases rapidly as the number of concurrent tasks increases. The greedy heuristic also maintains a high success rate, but its delay grows much faster under heavy load because it relies on local decisions and cannot fully coordinate aggregation node selection, routing, and offset scheduling.

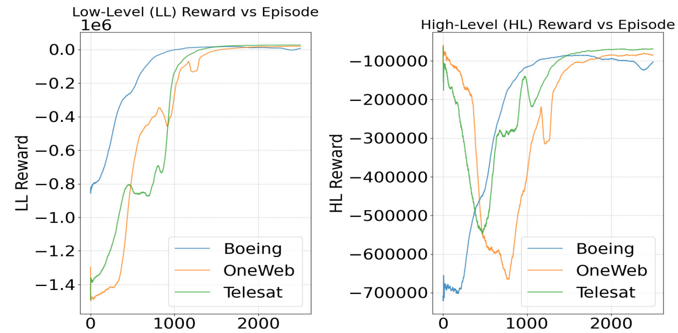

Fig. 2: a): High-Level (HL); b): Low-Level (LL) Reward versus Different Topologies: 1) Boeing; 2) OneWeb; 3) Telesat.

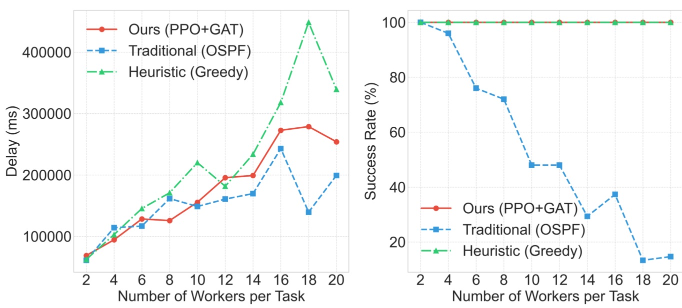

Fig. 3: a): Average Latency; b): Success Rate versus Number of Workers per Task over Boeing topology for 1): Our Solution; 2): Traditional Solution; 3): Heuristic Solution.

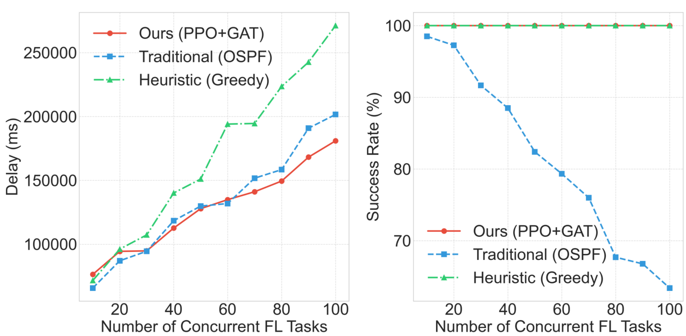

Fig. 4: a): Average Latency; b): Success Rate versus Number of Concurrent FL Tasks over Boeing topology for 1): Our Solution; 2): Traditional Solution; 3): Heuristic Solution.

VI. Conclusion

In this paper, we investigated FL completion time minimization in CQF-enabled deterministic LEO satellite networks. The problem jointly considers aggregation node selection, routing, and CQF offset scheduling under dynamic topology, limited onboard computing resources, and queue constraints. To solve this problem, we proposed a hierarchical GAT-PPO framework, where the high-level agent selects FL aggregation nodes and the low-level agent performs packet-level CQF-aware routing and offset control. Simulation results show that the proposed method achieves stable training convergence, reduces FL completion time, and maintains a high task success rate compared with OSPF and greedy heuristic schemes. These results demonstrate the effectiveness of jointly optimizing computation placement and deterministic communication scheduling for satellite federated learning.

Acknowledgment

The research work is supported by the National Natural Science Foundation of China under Grant No. 72501084, and in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2411 and the European Union's Horizon Europe research and innovation programme under the 6G-Path project (Grant No. 101139172).

References

[1] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "Scheduling for ground-assisted federated learning in leo satellite constellations," *arXiv preprint arXiv:2206.01952*, 2022. [Online]. Available: https://arxiv.org/abs/2206.01952

[2] Y. Shi, L. Zeng, J. Zhu, Y. Zhou, C. Jiang, and K. Letaief, "Satellite federated edge learning: Architecture design and convergence analysis," *arXiv preprint arXiv:2404.01875*, 2024. [Online]. Available: https://arxiv.org/abs/2404.01875

[3] Z. Lin, Z. Chen, Z. Fang, X. Chen, X. Wang, and Y. Gao, "Fedsn: A federated learning framework over heterogeneous leo satellite networks," *arXiv preprint arXiv:2311.01483*, 2023. [Online]. Available: https://arxiv.org/abs/2311.01483

[4] B. Matthiesen, "Federated learning in satellite constellations," *IEEE Network*, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1109/MNET.132.2200504

[5] H. Chen, M. Xiao, and Z. Pang, "Satellite based computing networks with federated learning," *arXiv preprint arXiv:2111.10586*, 2021. [Online]. Available: https://arxiv.org/abs/2111.10586

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[7] Z. Wang, O. He *et al.*, "Enabling deterministic flows scheduling in leo satellite networks," *IEEE Transactions on Mobile Computing*, 2025. [Online]. Available: https://www.computer.org/csdl/journal/tm/2025/05/10803087/22INwDjoPK0

[8] K. Zanbouri, M.-A. Rahim, J. John, C. Sreenan, H. Poor, and D. Pesch, "A comprehensive survey of wireless time-sensitive networking (tsn): Architecture, technologies, applications, and open issues," *arXiv preprint arXiv:2312.01204*, 2023. [Online]. Available: https://arxiv.org/abs/2312.01204

[9] F. Wang, D. Wu, W. He, Z. Li, Q. Zhang, and H. Yao, "Cpf: Bridging time-sensitive networks into large-scale leo satellite networks," in *Proc. IEEE IWCMC*, 2023.

[10] J. So, K. Hsieh, B. Arzani, S. Noghabi, S. Avestimehr, and R. Chandra, "Fedspace: An efficient federated learning framework at satellites and ground stations," *arXiv preprint arXiv:2202.01267*, 2022.

[11] A. Aijaz and S. Gufran, "Time-sensitive networking over 5g: Experimental evaluation of a hybrid 5g and tsn system with ieee 802.1qbv traffic," *arXiv preprint arXiv:2407.05989*, 2024. [Online]. Available: https://arxiv.org/abs/2407.05989

[12] Z. Lai, H. Li, and J. Li, "Starperf: Characterizing network performance for emerging mega-constellations," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, 2020, pp. 1–11.