Sparse Incremental Aggregation in Satellite Federated Learning

Nasrin Razmi*†, Sourav Mukherjee*†, Bho Matthiesen*†, Armin Dekorsy*†, Petar Popovski‡*
*University of Bremen, Department of Communications Engineering, Germany
†Gauss-Olbers Space Technology Transfer Center, University of Bremen, Germany
‡Aalborg University, Department of Electronic Systems, Denmark
email: {razmi, mukherjee, dekorsy, matthiesen}@ant.uni-bremen.de, petarp@es.aau.dk

***Abstract*—This paper studies Federated Learning (FL) in low Earth orbit (LEO) satellite constellations, where satellites are connected via intra-orbit inter-satellite links (ISLs) to their neighboring satellites. During the FL training process, satellites in each orbit forward gradients from nearby satellites, which are eventually transferred to the parameter server (PS). To enhance the efficiency of the FL training process, satellites apply in-network aggregation, referred to as incremental aggregation. In this work, the gradient sparsification methods from [1] are applied to satellite scenarios to improve bandwidth efficiency during incremental aggregation. The numerical results highlight an increase of over $4\times$ in bandwidth efficiency as the number of satellites in the orbital plane increases.**

***Index Terms*—Satellite Constellation, Federated learning, gradient sparsification, in-network computing**

I. Introduction

Megaconstellations of low Earth orbit (LEO) satellites have become an integral part to a variety of applications including global broadband access, Earth monitoring, and space exploration missions. The massive amount of data produced by these satellites—particularly high-resolution hyperspectral images—creates significant challenges in transmitting them back to the Earth. These challenges are exacerbated by the limitations of available bandwidth and the need to meet stringent latency requirements [2], [3].

Satellite federated learning (SFL) has recently developed as an promising technology to address the above challenges [4]–[6]. Unlike traditional approaches, SFL leverages distributed machine learning by enabling satellites to collaboratively train machine learning (ML) models without exchanging raw data. In this approach, each satellite trains the model locally using its own dataset and the global model parameters received from the central parameter server (PS), and then transmits the updated model parameters to the PS for aggregation. However, due to satellite orbital motion, the PS must wait a long time to receive model updates, making conventional federated learning (FL) impractical for satellite constellations.

To alleviate the impractical delay in the model convergence, a practical asynchronous FL approach was presented in [7], where each satellite functions as a separate node, communicating with a PS located at a ground station (GS) on Earth for model aggregation. Another asynchronous FL is presented in [8] to mitigate model staleness, leading to faster model convergence. Model accuracy of [7] is further improved in [9] by leveraging the predictability of satellite connections to the PS to schedule the transmissions of model parameters.

Further, to improve the model accuracy of SFL and address the long delays, [5] proposed to aggregate the updated model parameters inside each orbital plane and then transmitting the aggregated results to the PS. To this end, the satellites forward the updated parameters using their intra-orbit inter-satellite links (ISLs). This approach allows the necessary information from all satellites within an orbit to be accessed through a single satellite that remains visible to the GS. Moreover, inter-orbit ISLs are utilized in [10]–[12] to accelerate convergence.

In [5], to improve the bandwidth efficiency for model aggregation, each satellite applies gradient sparsification methods [13], [14]. In the approach proposed in [5], each satellite transmits the most effective elements of its parameters and the corresponding indices, along with the received elements and their indices to its neighboring satellite. For smaller sparsification ratios, the index supports for effective elements of the satellites are almost uncorrelated; therefore, making the aggregation inefficient and increasing communication budget with each hop [5]. The growing communication budget may become a bottleneck in transmission through the bandwidth-limited ISLs.

In this paper, building upon our prior research [5], we apply the sparsification approaches proposed in [1] to the satellite constellations connected with intra-orbit ISLs to improve the bandwidth efficiency while maintaining efficient communication. The structure of the paper is as follows: first, we introduce the system model in Section II. Next, FL with intra-orbit ISLs and sparsified gradient transmissions are discussed in Section III and Section IV, respectively. Finally, numerical results and conclusions are presented in Section V and Section VI.

II. System Model

A. Satellite Constellations

We consider a satellite constellation with a total of P orbital planes, where each plane p contains K_p satellites. The set of all satellites in the constellation is denoted as $\mathcal{K} = \bigcup_{p=1}^{P} \mathcal{K}_p =$

This work is supported by the German Research Foundation (DFG) under Grant EXC 2077 (University Allowance).

Algorithm 1 Satellite learning procedure

procedure SATLEARNPROC($\boldsymbol{w}^n$)
  initialize $\boldsymbol{w}_k^{n,0} = \boldsymbol{w}^n, \quad i = 0, \quad$ learning rate η
  for I epochs **do** $\triangleright$ I epochs of mini-batch stochastic gradient descent
    $\tilde{\mathcal{D}}_k \leftarrow$ Randomly shuffle $\mathcal{D}_k$
    $\mathscr{B} \leftarrow$ Partition $\tilde{\mathcal{D}}_k$ into mini-batches of size B
    for each batch $\mathcal{B} \in \mathscr{B}$ **do**
      $\boldsymbol{w}_k^{n,i} \leftarrow \boldsymbol{w}_k^{n,i} - \frac{\eta}{|\mathcal{B}|} \nabla_{\boldsymbol{w}} \left(\sum_{\boldsymbol{x} \in \mathcal{B}} f(\boldsymbol{x}, \boldsymbol{w}) \right)$
    end for
    $i \leftarrow i + 1$
  end for
  return $\boldsymbol{w}_k^{n,I}$
end procedure

$\{k_{1,1}, \ldots, k_{P,K_P}\}$, where the total number of satellites is $K = \sum_{p=1}^{P} K_p$. Satellites in orbit p move at a speed determined by $v_p = \sqrt{\frac{\mu}{h_p + r_E}}\,\mathrm{m/s}$, where $\mu = 3.98 \times 10^{14}\,\mathrm{m^3/s^2}$ is the geocentric gravitational constant, h_p is the orbit altitude, and $r_E = 6371\,\mathrm{km}$ is the Earth's radius. The orbital period of the satellites is calculated as $T_p = \frac{2\pi(r_E + h_p)}{v_p}$.

B. Computation Model

Satellites in the constellation participate in training an FL algorithm using the FedAvg method [15]. To this end, each satellite k utilizes its collected data set $\mathcal{D}_k$ and trains a ML model. In the training process, satellites solve the optimization problem

$$F(\boldsymbol{w}) = \min_{\boldsymbol{w}} \sum_{k=1}^{K} \frac{D_k}{D} F_k(\boldsymbol{w}), \tag{1}$$

where $D_k = |\mathcal{D}_k|$, $D = \sum_{k=1}^{K} D_k$ and the local loss function $F_k(\boldsymbol{w})$ is defined as

$$F_k(\boldsymbol{w}) = \frac{1}{D_k} \sum_{\boldsymbol{x} \in \mathcal{D}_k} f(\boldsymbol{w}, \boldsymbol{x}), \tag{2}$$

with $f(\boldsymbol{w}, \boldsymbol{x})$ as the per-sample loss function. The training process is coordinated by the PS over N iterations to solve (1). We consider a GS functioning as the PS.

For the nth iteration, each satellite k after receiving the global model parameters $\boldsymbol{w}^n$, performs I local steps of mini-batch stochastic gradient descent to obtain $\boldsymbol{w}_k^{n,I}$, as described in Algorithm 1 [7]. Then, the satellite transmits the gradients $\boldsymbol{g}_k^n$ to the GS, where $\boldsymbol{g}_k^n$ is defined as

$$\boldsymbol{g}_k^n = \boldsymbol{w}_k^{n,I} - \boldsymbol{w}^n. \tag{3}$$

The GS then aggregates $\boldsymbol{g}_k^n$, and updates the global model parameters as

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \sum_{k=1}^{K} \frac{D_k}{D} \boldsymbol{g}_k^n, \tag{4}$$

Afterwards, the GS transmits $\boldsymbol{w}^{n+1}$ back to the satellites for the next iteration.

C. Communication Model

To enable gradient transmission, each satellite in the constellation is equipped with three communication devices: one for the communication with the GS and the other two for the intra-orbit ISLs. In each plane, each satellite connects with two of its nearest orbital neighbors, establishing a ring network. Further, the communication with the GS is only possible when the Earth does not obstruct the line-of-sight (LoS).

We model the channel between two satellites k and i as a complex Gaussian channels, where the maximum achievable data rate is $r(k,i) = B \log_2(1 + \mathrm{SNR}(k,i))$, with B representing the allocated bandwidth, and the signal-to-noise ratio (SNR) given as [16], [17]

$$\mathrm{SNR}(k,i) = \frac{P_t G_k(i) G_i(k)}{N_0 L(k,i)}. \tag{5}$$

Further, the $\mathrm{SNR}(k,i)$ is zero, if there is no LoS. Here, $G_k(i)$ denotes the average antenna gain of satellite k towards satellite i, and $N_0 = k_B T B$ is the total noise power, where the Boltzmann constant is $k_B = 1.380649 \times 10^{-23}\,\mathrm{J/K}$ and T is the receiver noise temperature. Additionally, P_t is the transmitted power at the satellite, and the free space path loss $L(k,i)$ is defined as

$$L(k,i) = \left(\frac{4\pi f_c d(k,i)}{c} \right)^2, \tag{6}$$

where f_c is the carrier frequency, c is the speed of light, and $d(k,i)$ is the distance between satellite k and i. We assume fixed-rate transmission links with the minimum rate possible in the links.

III. FEDERATED LEARNING USING INTRA-ORBIT ISLS

For FL without the intra-orbit ISLs, each satellite must connect two times to the GS during each iteration n: first, to receive the global parameters $\boldsymbol{w}^n$ and then, to transmit back the updated gradient $\boldsymbol{g}_k^n$. Both of these transmissions require LoS between GS and the satellite. However, due to the satellite's motion, LoS to the GS lasts for a limited time during each orbital period T_p. Most of the time, the LoS is obstructed by the Earth. Therefore, the GS must wait until each satellite is visible (LoS) to transmit the global parameters $\boldsymbol{w}^n$, and then wait again for the satellite to be visible to receive the updated gradient $\boldsymbol{g}_k^n$. This introduces significant delays in model training and can lead to outdated data in time-critical applications.

Long delays can be minimized by enabling neighboring satellites to communicate through intra-orbit ISLs and by strategically scheduling transmissions to leverage the predictable paths of satellite orbits. This approach, termed *incremental aggregation*, operates as follows: in each iteration n, the satellite with LoS to the GS receives the global model parameters $\boldsymbol{w}^n$. These parameters are then distributed to other satellites within the orbit via intra-orbit ISLs. Another satellite with optimal visibility relays the updated aggregated gradients $\boldsymbol{g}_p^n = \sum_{k=1}^{K_p} D_k \boldsymbol{g}_k^n$ back to the GS. Each iteration n for an

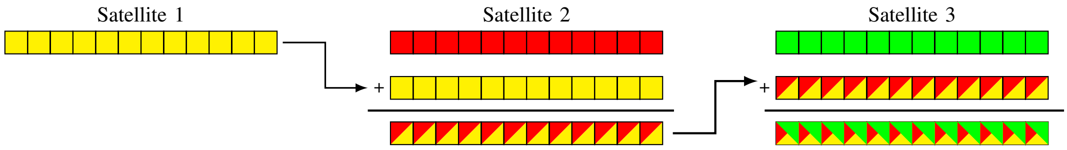

Fig. 1. The incremental aggregation method is applied across three adjacent satellites, with Satellite 1 being the farthest from the sink.

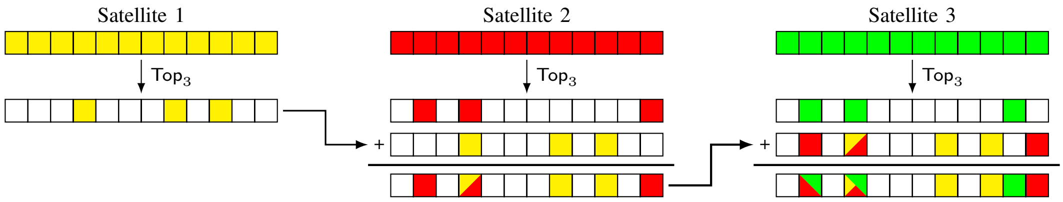

Fig. 2. The Sparse incremental aggregation method is applied across three adjacent satellites, with Satellite 1 being the farthest from the sink.

orbit is therefore structured into three distinct phases: parameter distribution, computation, and aggregation.

In the parameter distribution phase, one satellite of the orbit p which is selected for its optimal visibility to the GS, referred to as the source, receives $\boldsymbol{w}^n$ at time t. The source satellite then identifies another satellite in the same orbit as sink, which delivers the aggregated updates to the GS. The source selects sink satellite based on the computation time of each satellite, total aggregation time, and the expected visibility of the satellites. Once the source has identified the sink, it transmits $\boldsymbol{w}^n$, along with the sink's ID, to its neighboring satellites in both directions. The neighboring satellites then relay the received $\boldsymbol{w}^n$ and the ID of the sink to the next satellite in sequence, continuing until all satellites have received the packet.

After forwarding the received packet, each satellite k begins its training process as in Algorithm 1. The aggregation phase starts after training phase. In this phase, the satellites farthest from the sink initiate the process by transmitting their updated gradients to nearest neighboring satellite, toward the sink. This process continues in both directions along the ring network [5], [18].

To illustrate the incremental aggregation method with more details, we assume satellite $k-1$ is the farthest satellite from the sink. This satellite first calculates the shortest path to the sink based on the sink ID received during distribution phase. Once the shortest path is determined, the satellite transmits its gradients, scaled by its data size, as $\gamma_{k-1}^n \leftarrow D_{k-1}\boldsymbol{g}_{k-1}^n$, to a neighboring satellite k, which is chosen based on the shortest path to the sink. Note that the shortest path selected by the satellite $k-1$ involving satellite k, is also the shortest path for the satellite k towards the sink. Therefore, if the satellite k performs shortest path search, it will also come up with the same path. The satellite k then aggregates the received parameters with its own gradients as $\gamma_k^n \leftarrow D_k\boldsymbol{g}_k^n + \gamma_{k-1}^n$, and transmits this to the next satellite, $k+1$. These steps continue until the sink receives the aggregated gradients from both directions.

The key advantage of incremental aggregation is that it maintains a constant outgoing data size. Each satellite transmits data of size $n_d\omega$, where n_d denotes the gradient dimension and ω represents the storage size for a single gradient entry. In each iteration, the total data transmitted within an orbital plane p is $K_p n_d \omega$. However, when training models with large n_d in satellite mega-constellations, the data transmission size can become a bottleneck, especially with bandwidth-limited ISLs.

IV. Sparse Transmission

To improve the bandwidth efficiency of incremental aggregation, an effective approach is to compress gradients into a sparser representation. Among various compression techniques, Top_Q sparsification is a popular choice due to its strong performance. In the Top_Q approach, the gradient is converted into a sparse vector, retaining only the Q largest-magnitude elements [14]. Thus, combining Top_Q sparsification with incremental aggregation can significantly reduce the overall communication budget, making it particularly advantageous for satellite constellations.

A. Sparse Incremental Aggregation

In this approach, termed *sparse incremental aggregation (SIA)*, the kth satellite updates its gradient $\boldsymbol{g}_k^n$ by combining it with the sparsification error from the previous iteration, $\boldsymbol{e}_k^{n-1}$, as follows: $\tilde{\boldsymbol{g}}_k^n \leftarrow D_k\boldsymbol{g}_k^n + \boldsymbol{e}_k^{n-1}$. This step, known as error-feedback [1], plays a crucial role in integrating the residuals from earlier iterations, which helps accelerate convergence. Then, the Top_Q is applied to $\tilde{\boldsymbol{g}}_k^n$, resulting in $\bar{\boldsymbol{g}}_k^n \leftarrow \text{Top}_Q(\tilde{\boldsymbol{g}}_k^n)$. Further, the residual for next iteration is updated as $\boldsymbol{e}_k^n \leftarrow \tilde{\boldsymbol{g}}_k^n - \bar{\boldsymbol{g}}_k^n$.

In the aggregation phase, satellite k aggregates the incoming gradient γ_{k-1}^t with its sparsified gradient $\bar{\boldsymbol{g}}_k^n$, updating it as $\gamma_k^t \leftarrow \gamma_{k-1}^t + \bar{\boldsymbol{g}}_k^n$, before transmitting it to the next satellite $k+1$. The overall process is outlined in Algorithm 2 [1]. Further, the aggregation is performed in the common indices and otherwise values are forwarded with indices.

To illustrate, consider Fig. 2, and the process begins with satellite 1, which is farthest from the sink satellite. After

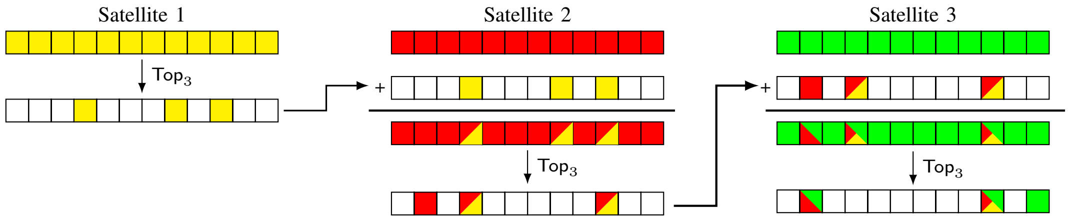

Fig. 3. The constant-length sparsification method is applied across three adjacent satellites, with Satellite 1 being the farthest from the sink.

Algorithm 2 Sparse incremental aggregation at satellite k

1: **Input** $\boldsymbol{g}_k^n$, $\boldsymbol{\gamma}_{k-1}^n$
2: Error feedback $\tilde{\boldsymbol{g}}_k^n \leftarrow D_k \boldsymbol{g}_k^n + \boldsymbol{e}_k^{n-1}$
3: Sparsification $\bar{\boldsymbol{g}}_k^n \leftarrow \text{Top}_Q(\tilde{\boldsymbol{g}}_k^n)$
4: Update error $\boldsymbol{e}_k^n \leftarrow \tilde{\boldsymbol{g}}_k^n - \bar{\boldsymbol{g}}_k^n$
5: Aggregation $\boldsymbol{\gamma}_k^n \leftarrow \boldsymbol{\gamma}_{k-1}^n + \bar{\boldsymbol{g}}_k^n$
6: **Return** $\boldsymbol{\gamma}_k^n$

Algorithm 3 Constant-length sparse incremental aggregation at satellite k

1: **Input** $\boldsymbol{g}_k^n$, $\boldsymbol{\gamma}_{k-1}^n$
2: Error feedback $\tilde{\boldsymbol{g}}_k^n \leftarrow D_k \boldsymbol{g}_k^n + e_k^{n-1}$
3: Aggregation $\tilde{\boldsymbol{\gamma}}_k^n \leftarrow \tilde{\boldsymbol{g}}_k^n + \boldsymbol{\gamma}_k^n$
4: Sparsification $\boldsymbol{\gamma}_k^n \leftarrow \text{Top}_Q(\tilde{\boldsymbol{\gamma}}_k^n)$
5: Update error $\boldsymbol{e}_k^n \leftarrow \tilde{\boldsymbol{\gamma}}_k^n - \boldsymbol{\gamma}_k^n$
6: **Return** $\boldsymbol{\gamma}_k^n$

applying the Top$_3$ operation on its error-compensated gradient, satellite 1 transmits its sparse gradient $\bar{\boldsymbol{g}}_1^n$ to satellite 2. For example, assume satellite 1 retains the values corresponding to indices $\{4, 8, 10\}$, where the indices are numbered starting from 1. Similarly, satellite 2 performs the Top$_3$ operation on its error-compensated gradient, resulting in values at indices $\{2, 4, 12\}$. Satellite 2 aggregates the values of the common indices (in this case, index 4). It then forwards this updated value of index 4 together with received values from satellite 1 in indices $\{8, 10\}$ and its own gradient values in indices $\{2, 12\}$ to the satellite 3. Satellite 3 repeats this process, combining and forwarding sparse gradients as the aggregate moves closer to the sink satellite.

In SIA, the outgoing data budget for each satellite depends on the overlap between the support of the incoming sparse aggregate and its own sparse gradient. When the supports align, the outgoing budget remains unchanged from that of the previous satellite, allowing for the benefits of incremental aggregation. However, as noted in [1], [5], after applying the Top$_Q$ operation with a low Q, the gradient supports become nearly uncorrelated. As a result, SIA primarily involves forwarding both the sparse gradient from the previous satellite and the satellite's own sparse gradient, with aggregation occurring in only a few indices. This leads to a growing data budget as the aggregate moves toward the sink satellite, ultimately reducing the efficiency of incremental aggregation. Additionally, due to the variability in transmission budget requirements across clients, predetermining a fixed budget becomes challenging.

B. Constant-Length Sparse Incremental Aggregation

To leverage the benefits of incremental aggregation and stabilize the increasing budget in SIA, an intuitive approach is to apply Top$_Q$ after combining each satellite's gradient with the incoming values from the previous satellite. In this setup, the satellite aggregates only values at shared indices, while retaining its own gradient values at other indices. This ensures a fixed transmission budget of Q parameters per satellite, with a budget of $(\omega + \lceil \log_2 n_d \rceil)Q$. Accordingly, this approach is termed *constant-length sparse incremental aggregation (CL-SIA)*.

Here, at each nth iteration, satellite k begins by updating its gradient to account for errors from the previous iteration, resulting in an error-compensated gradient: $\tilde{\boldsymbol{g}}_k^n \leftarrow D_k \boldsymbol{g}_k^n + \boldsymbol{e}_k^{n-1}$. Next, satellite k aggregates this error-compensated gradient with the sparse aggregate $\boldsymbol{\gamma}_{k-1}^n$ received from the preceding satellite $k-1$, yielding $\tilde{\boldsymbol{\gamma}}_k^n \leftarrow \boldsymbol{\gamma}_{k-1}^n + \tilde{\boldsymbol{g}}_k^n$. The Top$_Q$ is then applied to this aggregate, resulting in $\boldsymbol{\gamma}_k^n \leftarrow \text{Top}_Q(\tilde{\boldsymbol{\gamma}}_k^n)$. This sparse $\boldsymbol{\gamma}_k^n$ is then forwarded to the next satellite $k+1$, along with the associated indices. Finally, satellite k updates its sparsification error $\boldsymbol{e}_k^n \leftarrow \tilde{\boldsymbol{\gamma}}_k^n - \boldsymbol{\gamma}_k^n$, as outlined in Algorithm 3 [1].

To illustrate, consider the Fig. 3, which begins with satellite 1, that is farthest from the sink satellite. Satellite 1 first applies the Top$_3$ operation on its error-compensated gradient, retaining only the most significant components. It then transmits this sparse gradient to satellite 2. For instance, suppose satellite 1 retains values corresponding to the indices $\{4, 8, 10\}$. Upon receiving this sparse gradient, satellite 2 aggregates with its own error-compensated gradient, combining values at common indices. It then applies the Top$_Q$ operation on this aggregate to produce a sparse result, with selected indices, say, $\{2, 4, 12\}$. Therefore, utilizing benefits of the incremental aggregation. Similarly, satellite 3, upon receiving the aggregate from satellite 2 at indices $\{2, 4, 12\}$, performs the same steps to produce its own sparse aggregate, as the aggregate moves closer to the sink satellite.

Note that, the CL-SIA leverages incremental aggregation by maintaining a consistent budget at each satellite. However, as aggregate values accumulate across satellites, combining multiple previous gradients, it's possible that the actual Top$_Q$ values of a satellite's own gradient might not be retained in the final Top$_Q$ operation after aggregation. This is more likely to

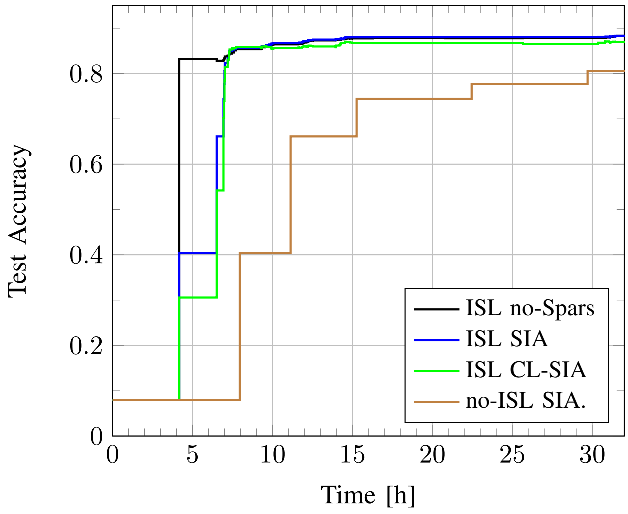

Fig. 4. Accuracy with respect to time with sparsification ratio $q = 0.01$ for a constellation with $K = 40$ satellites, where the GS is located in Bremen, Germany. Note that SIA and CL-SIA stand for sparse incremental aggregation and constant-length sparse incremental aggregation, respectively.

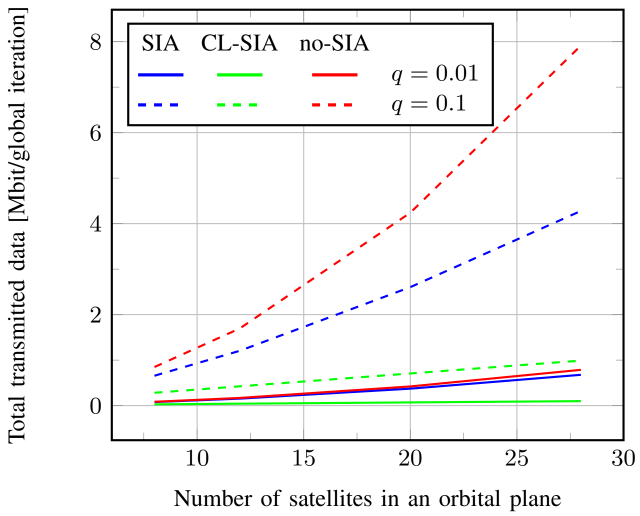

Fig. 5. Total transmitted data with respect to the number of the satellites in an orbital plane, with the sparsification ratios of $q = 0.01$ and $q = 0.1$. Note that SIA and CL-SIA stand for sparse incremental aggregation and constant-length sparse incremental aggregation, respectively.

happen as the aggregate moves toward the sink satellite. That leads to higher individual sparsification error at the satellites, which are closer to the sink satellite, potentially increasing the overall error and degrading convergence.

V. Numerical Evaluation

We present the performance of the proposed sparse incremental aggregation on a Walker star constellation with $K = 40$ satellites located in $P = 5$ evenly spaced orbital planes at inclination $85°$ at an altitude $h_p = 2000\mathrm{km}$. Each orbital plane consists of 8 equidistant satellites. The PS located in Bremen, Germany.

For communication between satellites or between the GS and the satellites, we set the transmission power on $P_t = 40\,\mathrm{dBm}$, and both the transmitter and receiver antennas gains on 32.13 dBi. Communication occurs over a channel with a bandwidth of $B = 500\,\mathrm{MHz}$. The carrier frequency is set to $f_c = 20\,\mathrm{GHz}$, and the receiver noise temperature is $T = 354\,\mathrm{K}$ [19].

We evaluate the ML performance using a logistic regression model with $n_d = 7850$ trainable parameters on the MNIST dataset, which consists of grayscale images of handwritten digits ranging from 0 to 9 [20]. The data samples are distributed evenly and randomly across the satellites.

Fig. 4 shows the test accuracy over wall clock time for incremental aggregation (IA) with and without sparsification, as well as with and without ISLs. In the case without ISLs (no-ISL SIA), each satellite, in every iteration, sparsifies its gradients after training and transmits them to the GS during its visits. In this regard, the GS must wait to receive the sparsified gradient parameters individually from each satellite. The sparsification ratio is set on $q = 0.01$. As it is seen, both SIA and CL-SIA, when combined with the ISL algorithm, achieve higher accuracy in a shorter time compared to the scenario without ISLs. This improvement occurs because, in the absence of intra-orbit ISLs, the GS must wait to receive or transmit parameters from all satellites. Moreover, as observed, the case without sparsification achieves higher accuracy initially for several hours. However, after that period, the performance with (ISL SIA) and without (ISL no-Spars) sparsification converges. A noteworthy point is that SIA shows slightly better performance than CL-SIA, which is attributed to the transmission of more data.

Figure 5 illustrates the total transmission data required to collect the gradients within a single orbital plane for both the SIA and CL-SIA algorithms. We evaluate the scenario with varying the number of satellites from 8 to 28 in the orbit, using sparsification ratios of $q = 0.01$ and $q = 0.1$. With the CL-SIA algorithm, we observe a substantial reduction in data load as the amount of transmitted data per hop remains fixed at its minimum possible value. In contrast, with the SIA algorithm, as the probability of having non-zero and non-overlapping entries increases after each hop, the communication load becomes significantly higher. Incorporating IA in each hop leads to a notable reduction in the communication load.

VI. Conclusions

We considered satellites equipped with intra-orbit ISLs. Using these ISLs, satellites collectively forward model parameters within each orbit, either transmitting them to or receiving them from the GS. The relaying of model parameters to the GS relies on in-network aggregation. The system's performance is further improved by implementing advanced sparsification algorithms in aggregation step to optimize bandwidth usage. By utilizing the algorithms proposed in [1], we could significantly reduce the communication load for satellite constellations.

References

[1] S. Mukherjee, N. Razmi, A. Dekorsy, P. Popovski, and B. Matthiesen, "Sparse incremental aggregation in multi-hop federated learning," in *2024*

IEEE 25th Int. Workshop on Signal Process. Adv. in Wireless Commun. (SPAWC), 2024, pp. 41–45.

[2] G. Giuffrida *et al.*, "Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images," *Remote Sensing*, vol. 12, no. 14, p. 2205, 2020.

[3] I. Leyva-Mayorga *et al.*, "LEO small-satellite constellations for 5G and beyond-5G communications," *IEEE Access*, vol. 8, pp. 184 955–184 964, 2020.

[4] B. Matthiesen, N. Razmi, I. Leyva-Mayorga, A. Dekorsy, and P. Popovski, "Federated learning in satellite constellations," *IEEE Netw.*, May 2023.

[5] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "On-board federated learning for satellite clusters with inter-satellite links," *IEEE Trans. Commun.*, Jan. 2024.

[6] H. Chen, M. Xiao, and Z. Pang, "Satellite-based computing networks with federated learning," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 78–84, 2022.

[7] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "Ground-assisted federated learning in LEO satellite constellations," *IEEE Wireless Commun. Lett.*, vol. 11, no. 4, pp. 717–721, Apr. 2022.

[8] L. Wu and J. Zhang, "FedGSM: Efficient federated learning for LEO constellations with gradient staleness mitigation," in *2023 IEEE 24th Int. Workshop on Signal Process. Adv. in Wireless Commun. (SPAWC)*, 2023, pp. 356–360.

[9] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "Scheduling for ground-assisted federated learning in LEO satellite constellations," in *Eur. Signal Process. Conf. (EUSIPCO)*, Belgrade, Serbia, Aug. 2022.

[10] Y. Shi *et al.*, "Satellite federated edge learning: Architecture design and convergence analysis," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 10, pp. 15 212–15 229, 2024.

[11] F. Zhou, Z. Wang, Y. Shi, and Y. Zhou, "Decentralized satellite federated learning via intra- and inter-orbit communications," in *2024 IEEE Int. Conf. on Commun. Workshops (ICC)*, 2024, pp. 786–791.

[12] Z. Zhai *et al.*, "FedLEO: An offloading-assisted decentralized federated learning framework for low earth orbit satellite networks," *IEEE Trans. on Mobile Comput.*, vol. 23, no. 5, pp. 5260–5279, 2024.

[13] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Conf. Empir. Methods Nat. Lang. Process.*, Sep. 2017.

[14] D. Alistarh *et al.*, "The convergence of sparsified gradient methods," in *Conf. Adv. Neural Inf. Process. Syst.*, Dec. 2018, p. 5976–5986.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artif. Intell. Statist. (AISTATS)*, Fort Lauderdale, FL, Apr. 2017.

[16] I. Leyva-Mayorga, B. Soret, and P. Popovski, "Inter-plane inter-satellite connectivity in dense LEO constellations," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3430–3443, 2021.

[17] L. J. Ippolito Jr, *Satellite Communications Systems Engineering*. John Wiley & Sons, 2017.

[18] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "On-board federated learning for dense LEO constellations," in *IEEE Int. Conf. Commun.*, Seoul, Korea, May 2022.

[19] I. Leyva-Mayorga *et al.*, "NGSO constellation design for global connectivity," in *Non-Geostationary Satellite Communications Systems*, E. Lagunas, S. Chatzinotas, K. An, and B. F. Beidas, Eds. Hertfordshire, UK: IET, Dec. 2022, ch. 9, pp. 189–236.

[20] Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST database of handwritten digits.