2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 978-1-6654-6483-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PIMRC56721.2023.10293916

Cross-Domain Federated Computation Offloading for Age of Information Minimization in Satellite-Airborne-Terrestrial Networks

X. Xia*, H. H. Esmat*, K. Dyer*, B. Lorenzo*, L. Guo+
*Dept. Electrical and Computer Engineering, University of Massachusetts Amherst, USA
+Dept. Electrical and Computer Engineering, Clemson University, USA
*{xiaohaoxia, habdelhafez, kpdyer, blorenzo}@umass.edu, +linkeg@clemson.edu

***Abstract*—Satellite-Airborne-Terrestrial Networks (SATNs) are expected to provide communication and edge-computing services for a plethora of IoT applications. However, preserving the freshness of information is challenging since it requires timely data collection, bandwidth, and offloading decisions across different administrative domains. In this paper, we aim to optimize the age of information (AoI) and energy consumption tradeoff when serving multiple traffic classes in SATNs. A cross-domain federated computation offloading algorithm (Fed-SATEC-Off) is presented in which different service providers (SPs) collaborate to allocate the bandwidth while unmanned aerial vehicles (UAVs) and satellites make decisions to collect, relay, and offload the computing tasks. Given the requirements of each traffic class, the optimum collaborative strategies between SPs, UAVs, and satellites are obtained together with the computation offloading topology. Our algorithm is based on multi-agent actor-critic and incorporates federated learning to improve the convergence of the learning process. The numerical results show that Fed-SATEC-Off reduces the AoI by factor 4 and achieves faster convergence than existing approaches.**

***Index Terms*—Age of Information (AoI), satellite-airborne-terrestrial network, heterogeneous IoT, multi-agent deep reinforcement learning, mobile edge computing.**

I. Introduction

The sixth generation of wireless networks (6G) [1], [2] is expected to provide services for advanced applications with massive use of artificial intelligence [3], [4] such as autonomous driving, industrial Internet of Things (IIoT), telemedicine, and the metaverse. These applications require edge computing resources to process data for IoT devices with limited capabilities. In parallel, aerial access networks consisting of unmanned aerial vehicles (UAVs), balloons, and small-size Low Earth Orbit (LEO) satellites have been deployed by several private and public entities, such as SpaceX Starlink [5]. These platforms are equipped with edge computing and their integration with the terrestrial network is promising to expand communication, computing, caching, and sensing at a global scale.

Preserving the freshness of information is crucial in many IoT applications [6]. The age of information (AoI) has been used as a measure of data freshness [7], which requires a timely allocation of communication and computing resources to extract useful features from data. The AoI and energy efficiency tradeoffs have been researched actively in terrestrial and airborne networks. In [7], [8], they focus on optimizing UAV trajectory to reduce the communication latency and energy consumption of IoT devices and show significant performance improvements compared to fixed servers. Zhu et al. [7] utilized both multi-agent reinforcement learning (MARL) and federated learning (FL) techniques to develop policies for trajectory planning and resource allocation of UAVs and resource allocation. However, integrating SATNs and their AoI and energy efficiency tradeoff analysis is a more challenging problem due to their multi-domain nature, which require bandwidth, computing, and relaying decisions among different operator networks.

Gao et al. [9] decompose the AoI minimization problem into the terrestrial and satellite networks. They propose an algorithm for base station-user selection and transmission power optimization in the terrestrial network and a coalition formation game among satellites. Zhang et al. [10] jointly optimize computation decisions and resource allocation to minimize energy consumption by computing locally, at the satellite or cloud using a Deep Deterministic Policy Gradient (DDPG) algorithm, which is centralized. Some works studied the AoI in SATNs for specific applications [11], [12]. Wang et al. [11] minimize the AoI of the status updates for vehicular networks. Therefore, how to make edge computing decisions to enable IoT data collection and computing across different domains with heterogeneous computing and caching capabilities is especially relevant in SATNs, given the different traffic requirements in IoT applications.

In this paper, we present a cross-domain federated computation offloading algorithm (Fed-SATEC-Off) in which different service providers (SPs), UAVs, and satellites collaborate to optimize the AoI and energy efficiency tradeoff. The algorithm is distributed, and each SP decides on the bandwidth allocation to UAVs and satellites for data collection, relaying, and computing. Our approach significantly reduces the AoI and achieves faster convergence than existing schemes.

This work is partially supported by the US National Science Foundation under Grant CNS-2008309 and CNS-2008049.

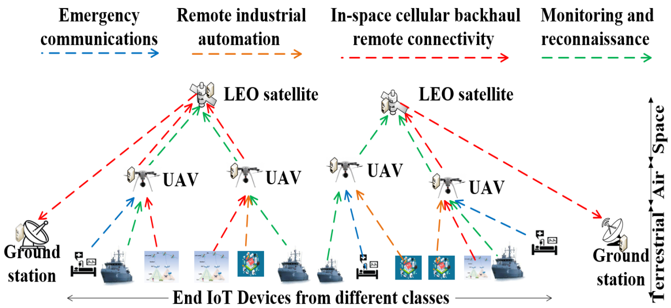

Fig. 1: Computation offloading in SATNs with heterogeneous traffic classes.

II. System Model

A. Satellite-Airborne-Terrestrial Network

We consider a SATN architecture, as shown in Fig. 1, in which terrestrial, airborne, and satellite domains are operated by a different service provider (SP). SPs collaborate to serve heterogeneous traffic demands from IoT devices denoted by k_c where c is the class index, $c = \{1, ..., C\}$ and $k_c \in \mathcal{K}_c = \{1, ..., K_c\}$. Examples of service classes include emergency communications (EC), in-space cellular backhaul (ICB), remote industrial automation (RIA), and monitoring and reconnaissance (MAR), which have different AoI requirements. We consider that the airborne and satellite SPs deploy a set $\mathcal{I}$ of unmanned aerial vehicles (UAVs), and a set $\mathcal{X}$ of LEO satellites (LEOs), respectively. Each UAV $i \in \mathcal{I}$ collects data from IoT devices and performs local computing or offloads it to an LEO satellite $x \in \mathcal{X}$. Similarly, an LEO satellite $x \in \mathcal{X}$ collects data from UAVs and performs local computing or offloads it to a ground station j. Finally, the ground stations execute the computing tasks received from the LEO satellites.

B. Communication Model

Each SP_z in domain $z \in \{1, 2, 3\}$ has a total bandwidth W_z, where $z = 1$ is the terrestrial domain, $z = 2$ is the airborne domain, and $z = 3$ is the satellite domain. Each channel will be allocated a bandwidth ratio $w_{k_c}(t)$, $w_i^c(t)$, and $w_x^c(t)$ through frequency division for communication transmission of IoT devices, UAVs, and satellites, respectively. Our communication model is divided into time slots. Thus, the link capacity $R_{k_c,i}$ between device k_c and UAV i, $R_{i,x}^c$ between UAV i and LEO x, and the link capacity $R_{x,j}^c$ between LEO x and GS j are

$$\begin{aligned} R_{k_c,i}(t) &= w_{k_c}(t) W_1 log_2(1 + P_{k_c} \cdot h_{k_c,i}/\xi_i) \\ R_{i,x}^c(t) &= w_i^c(t) W_2 log_2(1 + P_i \cdot h_{i,x}/\xi_x) \\ R_{x,j}^c(t) &= w_x^c(t) W_3 log_2(1 + P_x \cdot h_{x,j}/\xi_j) \end{aligned} \tag{1}$$

where W_z is the bandwidth of channels in domain z, P_{k_c}, P_i, and P_x are the transmission power of IoT device k_c, UAV i, and satellite x, respectively. ξ_i, ξ_x, and ξ_j are the Gaussian noise power at UAV i, satellite x and GS_j. $h_{k_c,i}$, $h_{i,x}$, and $h_{x,j}$ are the propagation gain between IoT device and UAV, between UAV and LEO, and between LEO and GS, respectively.

C. Data Collection, Execution, and Offloading

Each IoT device k_c produces data packets of size d_{k_c} independently with an elapsed time ψ_{k_c}. The data generation rate at k_c and the data arrival probability are λ_{k_c} and χ_{k_c}, respectively. Each device generates packets that are stored locally until collected by a UAV. The UAVs adopt a random mobility model and collect data from IoT devices that fall within their collection range denoted by r_i on a channel that is available. Each UAV i has a buffer to store collected data packets with a maximum size of B_i. The UAVs cache the collected data packets in their buffer for later processing or offloading. The decision to collect data by UAV i is represented by $\textbf{collect}_i^c(t) = [col_{1,k_c}^c(t), \cdots, col_{q,k_c}^c(t), \cdots, col_{B_i,k_c}^c(t)]$, where $col_{q,k_c}^c(t) \in \{0, 1\}$ represents a binary decision. At each time t, the UAVs acquire data packets from IoT devices k_c.

$$\sum_{q=1}^{B_i} col_{q,k_c}^c(t) = 1 \tag{2}$$

If the UAV decides to compute the data packet locally, it will select the data packet from the buffer for execution. Similarly, when LEO decides to perform the local computation on a packet, LEO selects a packet from the buffer for execution. At each time t, the execution decisions of UAV and LEO are denoted as $\textbf{execution}_i^c(t) = \left[exe_{1,i}^c(t), \cdots, exe_{q,i}^c(t), \cdots,\ exe_{B_i,i}^c(t)\right]$, $\textbf{execution}_x^c(t) = \left[exe_{1,x}^c(t), \cdots, exe_{q,x}^c(t), \cdots,\ exe_{B_x,x}^c(t)\right]$, and

$$\begin{aligned} \textstyle\sum_{q=1}^{B_i} exe_{q,i}^c(t) = 1 \\ \textstyle\sum_{q=1}^{B_x} exe_{q,x}^c(t) = 1 \end{aligned} \tag{3}$$

where $exe_{q,i}^c(t), exe_{q,x}^c(t) \in \{0, 1\}$. The computation time required to execute a packet of size d_{k_c} in UAV i and LEO x is

$$\begin{aligned} \tau_{k_c}(t) = d_{k_c}(t)/f_i \\ \tau_{k_c}(t) = d_{k_c}(t)/f_x \end{aligned} \tag{4}$$

where f_i and f_x is the CPU frequency at UAV i and satellite x, respectively. Based on network prediction, when UAV i decides to do offloading, it will select a data packet in the buffer for offloading and transmit it to the nearest LEO. We assume that each LEO has a sufficient observation range to receive the data transmission from the UAV. Similarly, if LEO satellite x decides to offload a packet it will select it in its buffer for offloading and transmit it to the GS j. We denote their offloading decisions as $\textbf{offload}_{i,x}^c(t) = \left[off_{1,x}^c(t),\ ...,\ off_{B_i,X}^c(t)\right]$ and $\textbf{offload}_{x,j}^c(t) = \left[off_{1,j}^c(t),\ ...,\ off_{B_x,J}^c(t)\right]$ with

$$\begin{aligned} \textstyle\sum_{q=1}^{B_i} off_{q,x}^c(t) = 1 \\ \textstyle\sum_{q=1}^{B_x} off_{q,j}^c(t) = 1 \end{aligned} \tag{5}$$

Therefore, UAV i will collect data from IoT devices, and then execute, or offload it to LEO satellites. LEO will collect data from UAVs, then execute, or offload it to the GSs.

III. Problem Formulation

The freshness of the executed data is measured by the AoI. The calculation of AoI involves determining the difference between the current time t and the generation time t_{g,k_c} of the latest data packet executed from IoT device k_c at any of the devices responsible for performing computing on the received data.

$$A_{k_c}(t) = t - t_{g,k_c} \tag{6}$$

The AoI increases at a constant rate until a new packet is received. The objective is to minimize the average AoI and energy consumption. The energy cost $cost_{k_c}$ of performing a task by device k_c on a UAV i, LEO x, or GS j is calculated as follows:

$$cost_{k_c} = d_{k_c}\Bigg(\gamma_{k_c,i}E_i^{exe} + (1-\gamma_{k_c,i})\Big(E_{ix}^{off} + \alpha_{k_c,x}E_x^{exe} + (1-\alpha_{k_c,x})(E_{xj}^{off} + \beta_{k_c,x}E_j^{exe})\Big)\Bigg) \tag{7}$$

which is proportional to the energy cost to execute the data of IoT device k_c of size d_{k_c} and to the energy consumed to offload the data. E_i^{exe}, E_x^{exe}, and E_j^{exe} is the energy needed to execute one bit of data by UAV i, LEO x, and GS j, respectively. Note that if the UAV i does not execute the data of IoT device k_c, it will be offloaded to LEO and E_{ix}^{off} is the energy cost of offloading one bit of data from UAV i to LEO x. Similarly, if the LEO x does not execute the data received from UAVs, it will be offloaded to GS, and E_{xj}^{off} is the energy cost of offloading one bit of data from LEO i to GS j. $\gamma_{k_c,i}, \alpha_{k_c,i}, \beta_{k_c,i}$ are binary indicators in which $\gamma_{k_c,i} = 1$,$\alpha_{k_c,i} = 1$ or $\beta_{k_c,i} = 1$ if the computing task of IoT device is performed by UAV i, LEO x or GS j, respectively; 0 otherwise. The data of IoT device is not split between devices, so $\gamma_{k_c,i} + \alpha_{k_c,x} + \beta_{k_c,j} \le 1$. Based on the previous definitions, the overall AoI and energy cost minimization problem is

$$\min_{\boldsymbol{\delta},\boldsymbol{\vartheta},\boldsymbol{\gamma},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}} \sum_{c=1}^{C}\sum_{k_c=1}^{K_c}\sum_{i=1}^{I}\sum_{x=1}^{X}\sum_{j=1}^{J} \{\zeta_1 A_{k_c}(t) + \varrho\zeta_2 cost_{k_c}(t)\}/K_c \tag{8}$$

subject to

$$(1) - (5)$$

$$\sum\nolimits_{k_c=1}^{K_c} \vartheta_{ik_c} \le Y, \forall i \in \mathcal{I} \tag{8.a}$$

$$\sum\nolimits_{i\in\mathcal{I}} \vartheta_{ik_c} \le 1, \forall k_c \in \mathcal{K}_c \tag{8.b}$$

$$\sum\nolimits_{i=1}^{I} \delta_{xi} \le L, \forall x \in \mathcal{X} \tag{8.c}$$

$$\sum\nolimits_{x\in\mathcal{X}} \delta_{xi} \le 1, \forall i \in \mathcal{I} \tag{8.d}$$

$$\sum\nolimits_{k_c\in\mathcal{K}_c} w_{k_c} \le W_1 \tag{8.e}$$

$$\sum\nolimits_{i\in\mathcal{I}} w_i \le W_2 \tag{8.f}$$

$$\sum\nolimits_{x\in\mathcal{X}} w_x \le W_3 \tag{8.g}$$

$$\gamma_{k_c,i} + \alpha_{k_c,x} + \beta_{k_c,j} \le 1, \forall k_c \in \mathcal{K}_c, c \in \mathcal{C}, i \in \mathcal{I}, x \in \mathcal{X}, j \in \mathcal{J} \tag{8.h}$$

$$\vartheta_{ik_c}, \delta_{xi}, \gamma_{k_c,i}, \alpha_{k_c,x}, \beta_{k_c,j} = \{0,1\}, \forall k_c, i, x, j \tag{8.k}$$

$$A_{k_c}(t) \le A_c^{max}, \forall k, c \tag{8.l}$$

where a scaling factor ϱ is used along with weighting factors ζ_1 and ζ_2 to represent the importance of AoI and cost, respectively. The association status between IoT devices and UAVs is represented by a binary variable ϑ_{ik_c}, where i denotes UAVs and k_c denotes IoT devices. Constraint (8.a) and (8.b) limit the number of IoT devices that can be associated with each UAV and Y is a quota that specifies the maximum number of IoT devices that can be supported by a UAV. Constraint (8.b) ensures that each IoT device is associated with only one UAV at a time. The association status between UAVs and LEOs is represented by a binary variable δ_{xi}, where x denotes LEOs and i denotes UAVs. Constraint (8.c) and (8.d) limit the number of UAVs that can be associated with each LEO, and L is the maximum number of UAVs that can be supported by a LEO. Constraint (8.d) ensures that each UAV is associated with only one LEO at a time. The available bandwidth in different service providers is constrained by (8.e)-(8.g) for IoT devices, UAVs and LEO satellites, respectively. The constraint (8.e) limits the amount of bandwidth allocated to IoT devices to be no more than the available bandwidth W_1 in SP_1. The constraint (8.f) limits the amount of bandwidth allocated to UAVs to be no more than the available bandwidth W_2 in SP_2. The constraint (8.g) limits the amount of bandwidth allocated to LEOs to be no more than the available bandwidth W_3 in SP_3. The constraint (8.l) ensures that the achieved AoI for each IoT device does not exceed its maximum AoI. The optimum solution cannot be achieved by solving the previous optimization problem at each time instant, due to the complex dynamics and coupling of UAVs, satellites, and SPs in our system, which makes the problem NP-hard. Therefore, we propose a new approach to solve the optimization problem by formulating it as a Markov Decision Process (MDP) and applying an iterative online algorithm.

IV. Fed-SATEC-Off Framework

We reformulate the optimization problem in (8), and model it as a 3-level Markov Decision Process (MDP) to enable interaction between the edge agents (i.e., UAVs and LEO satellites) and the SPs. A cross-domain Federated SAT Edge Computing Offloading algorithm (Fed-SATEC-Off) is developed based on the multi-agent actor-critic learning framework [7] to facilitate cooperation between the UAVs, LEO satellites, and SPs. Then, to leverage the knowledge of edge agents and reduce the exchange of communication messages, we adopt federated learning. Thus, edge agents share their actor net parameters periodically and conduct federated updating.

A. MDP Formulation

1) States: The state of each UAV agent $\mathbf{s}_i(t)$ includes observations of the IoT devices, the buffer state of the UAV, and the allocated bandwidth proportion. The state of each satellite agent $\mathbf{s}_x(t)$ includes observations of the UAV (all information about the UAV), the buffer state of the satellite, and the allocated bandwidth. On the other hand, the state of each SP $\mathbf{s}_z(t)$ includes observations of the edge agents.

2) Actions: The action of each UAV i includes executing the computing tasks of IoT devices locally or offloading them to the nearest satellite. The action of UAV i is expressed as $\mathbf{a}_i^c(t) = [\mathbf{execute}_i^c(t), \mathbf{offload}_{i,x}^c(t)]$. Likewise, the actions of LEOs are executing the tasks received from UAVs locally or offloading them to GS. The action of LEO satellite x is expressed as $\mathbf{a}_x^c(t) = [\mathbf{execute}_x^c(t), \mathbf{offload}_{x,j}^c(t)]$. The action of the SPs consists of allocating a fraction of the bandwidth for their devices' transmissions. Thus, the action of SP_1 is $\mathbf{a}_{z=1}^c(t) = [\boldsymbol{w}_{k_c}(t)]$ which is used by device k_c to offload to an UAV. The action of SP_2 is $\mathbf{a}_{z=2}^c(t) = [\boldsymbol{w}_i(t)]$ which is used by UAV i to offload to LEO. Finally, the SP_3 allocates $\mathbf{a}_{z=3}^c(t) = [\boldsymbol{w}_x(t)]$ to offload to a GS.

3) Penalty: Due to the cooperation among agents to minimize AoI and cost, we define a global penalty shared by all agents. The current penalty in the t-th slot is represented as $p_g(t) = \zeta_1\overline{\mathrm{A(t)}} + \varrho\zeta_2\overline{\mathrm{cost(t)}}$, $\forall\ g = 1,..,M+X+Z$. Also, we define $P_g(t) = \sum_{l=0}^{T}\rho^l p_g(t+l)$ as the long-term penalty, where T denotes the duration of the time window, while ρ, which is bounded between 0 and 1, represents the rate at which the penalty decreases.

4) Transition Policies: We denote the interactions and state transitions for the different agents as $\mathcal{T}(\{\boldsymbol{s}_i(t+1)\}, \{\boldsymbol{s}_x(t+1)\}, \{\boldsymbol{s}_z(t+1)\} | \{\boldsymbol{s}_i(t)\}, \{\boldsymbol{s}_x(t)\}, \{\boldsymbol{s}_z(t)\}, \{\boldsymbol{a}_i(t)\}, \{\boldsymbol{a}_x(t)\}, \{\boldsymbol{a}_z(t)\})$.

B. Fed-SATEC-Off Algorithm

We adopt a multi-agent actor-critic framework (MAAC) to facilitate collaboration among edge agents and SPs to allocate the necessary bandwidth to collect, execute and offload the data from different classes of IoT devices. This framework includes dual neural networks per agent with primary actor/critic nets and target actor/critic nets. Figure 2 illustrates that every agent (UAV, LEO, or SP) interacts with the environment to identify the optimal action that results in the lowest system penalty P_g. To address the issue of approximated Q values being unstable, we utilize an experience reply buffer (with capacity B) and target networks. Since UAVs and LEOs have the same type of actions, we use the same network architecture to implement them. Edge agents (UAV and LEO) use a multiple input-output neural network to learn different actions based on state, including observations of IoT sensors, offload channel status, and buffer status. At the same time, we use multilayer perceptrons (MLP) to realize offloading scheduling and data local execution on UAV and LEO. The state of SPs contains the status information of all UAVs and LEOs, and the output is the bandwidth allocation for each edge agent.

To encourage the agent to perform random exploration, we implement a ϵ-greedy policy with probability ϵ. For agent g in the MAAC network, the parameters of the primary actor-network $\mathcal{A}_g$ and the target actor-network $\mathcal{A}'_g$ are $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}'_g$, respectively. Similarly, the parameters of the primary critic-network $\mathcal{C}_g$ and target critic-network $\mathcal{C}'_g$ are $\boldsymbol{\phi}_g$ and $\boldsymbol{\phi}'_g$, respectively. We use the primary networks to update the parameters of target actor and critic networks in every period T_u as

$$\begin{aligned}\boldsymbol{\theta}'_g &= \tau\boldsymbol{\theta}'_g + (1-\tau)\boldsymbol{\theta}_g \\ \boldsymbol{\phi}'_g &= \tau\boldsymbol{\phi}'_g + (1-\tau)\boldsymbol{\phi}_g\end{aligned} \tag{9}$$

The mixing weight, denoted by τ, ranges from 0 to 1. The actor and critic networks have distinct learning rates, $\eta_{\mathcal{A}}$ and $\eta_{\mathcal{C}}$, respectively. The critic networks are trained by minimizing the loss function's mean squared error

$$l_{\mathcal{C}g}(\boldsymbol{\phi}_g) := E[\,||\mathcal{C}_g(\boldsymbol{s}_g, \boldsymbol{a}_g; \boldsymbol{\phi}_g) - \hat{y}_g||^2] \tag{10}$$

where $\hat{y}_g = p_g + \rho\mathcal{C}'_g(\boldsymbol{s}'_g, \boldsymbol{a}'_g, \boldsymbol{\phi}'_g)$ is the estimated long-time Q value, and p_g is the penalty of each agent g. Thus, we define the loss function of the actor networks as follows

$$l_{\mathcal{A}g}(\boldsymbol{\theta}_g) := \mathcal{C}_g(\boldsymbol{s}_g, \mathcal{A}_g(\boldsymbol{s}_g; \boldsymbol{\theta}_g); \boldsymbol{\phi}_g) \tag{11}$$

Each SP computes the federated learning update strategy for the devices in its domain and keeps parameters with weight ω, and mixes the parameters from other SPs, as

Algorithm 1 Federated Satellite-Airborne-Terrestrial Edge Computing Offloading Algorithm

Initialize: The parameters of primary and target networks $(\boldsymbol{\theta}_{UAV})_i$, $(\boldsymbol{\theta}_{LEO})_x$, $(\boldsymbol{\theta}_{SP})_z$, $(\boldsymbol{\theta}_{UAV})'_i \leftarrow (\boldsymbol{\theta}_{UAV})_i$, $(\boldsymbol{\theta}_{LEO})'_x \leftarrow (\boldsymbol{\theta}_{LEO})_x$, $(\boldsymbol{\theta}_{SP})'_z \leftarrow (\boldsymbol{\theta}_{SP})_z$, $(\boldsymbol{\phi}_{UAV})_i$, $(\boldsymbol{\phi}_{LEO})_x$, $(\boldsymbol{\phi}_{SP})_z$, $(\boldsymbol{\phi}_{UAV})'_i \leftarrow (\boldsymbol{\phi}_{UAV})_i$, $(\boldsymbol{\phi}_{LEO})'_x \leftarrow (\boldsymbol{\phi}_{LEO})_x$, $(\boldsymbol{\phi}_{SP})'_z \leftarrow (\boldsymbol{\phi}_{SP})_z$,
for each $t = 1, 2, ...\text{Epoc}h_{max}$ **do**
  Generate random $\nu \in [0, 1]$;
  for each agent g **do**
    if $\nu < \epsilon$ or $|\boldsymbol{\mathcal{B}}[g]| < B$ **then**
      Select random actions $\boldsymbol{a}_g(t)$;
    else
      Ensemble local observation and states: $\boldsymbol{s}_g(t)$;
      Set actions: $\boldsymbol{a}_g(t) = \mathcal{A}_g(\boldsymbol{s}_g(t); \boldsymbol{\theta}_g)$
    end if
    Get next state $\boldsymbol{s'}(t+1)$ and obtain $p(t)$;
    Store $\{\boldsymbol{s}, \boldsymbol{a}, p, \boldsymbol{s'}\}$ into $\boldsymbol{\mathcal{B}}$;
    if $|\boldsymbol{\mathcal{B}}[g]| \geq B$ **then**
      Sample $\{\boldsymbol{s}_g, \boldsymbol{a}_g, p_g, \boldsymbol{s}'_g\}$ from $\boldsymbol{\mathcal{B}}[g]$;
      Predict $\boldsymbol{a}'_g = \mathcal{A}'_g(\boldsymbol{s}'_g; \boldsymbol{\theta'}_g)$ and Q-value;
      Calculate actor and critic losses using (10) & (11);
      Update the actor and critic weights:
        $\boldsymbol{\phi}_g^{t+1} \leftarrow \boldsymbol{\phi}_g^t - \eta_{\mathcal{C}} \bigtriangledown_\phi \tilde{l}_{\mathcal{C}g}(\boldsymbol{\phi}_g^t)$
        $\boldsymbol{\theta}_g^{t+1} \leftarrow \boldsymbol{\theta}_g^t - \eta_{\mathcal{A}} \bigtriangledown_\theta \tilde{l}_{\mathcal{A}g}(\boldsymbol{\theta}_g^t)$
    end if
    Every T_u steps, update target networks' weights by (9);
    Every E_f steps, update SP-federated according to (12);
  end for
end for

$$\boldsymbol{\theta}_{SP}^{t+1} = \boldsymbol{\theta}_{SP}^t.\boldsymbol{\Omega} \tag{12}$$

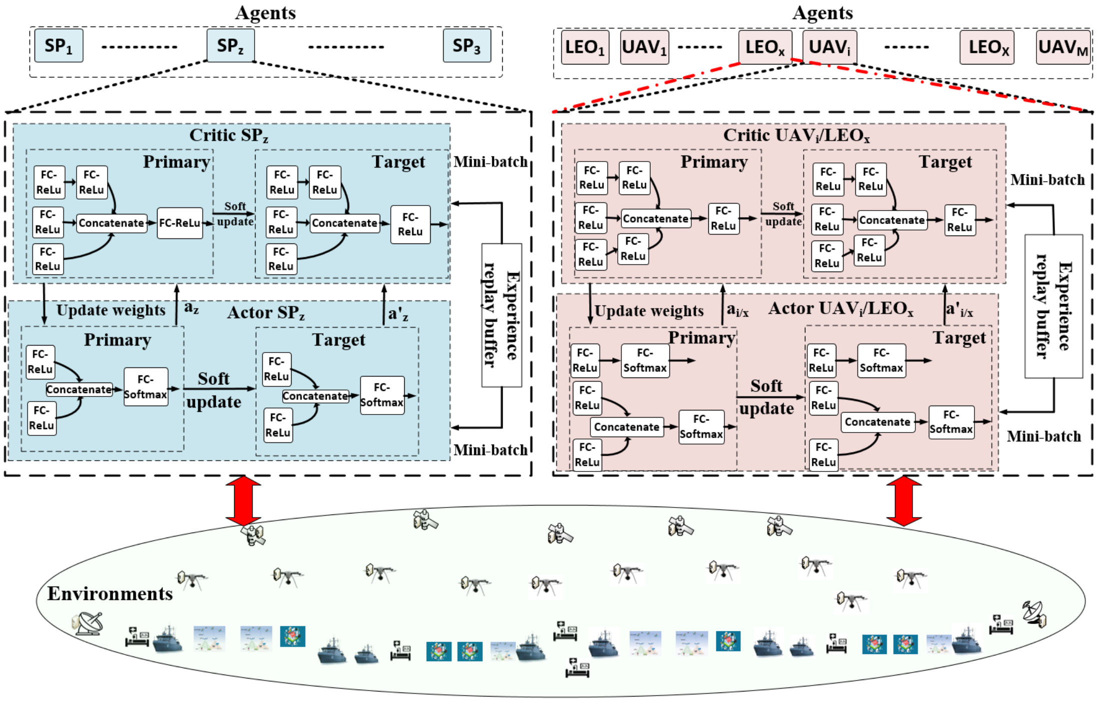

Fig. 2: Fed-SATEC-Off Collaborative Learning Architecture

where $\boldsymbol{\theta}_{SP}^t = [\boldsymbol{\theta}_1^t, .., \boldsymbol{\theta}_z^t, .., \boldsymbol{\theta}_Z^t]$ is the vector of all SP actor networks at the t-th learning epoch, and $\boldsymbol{\Omega}$ is the federated learning update matrix of each SP. In Algorithm 1, there is the corresponding federated update where the agent learns and updates the optimal policy while the system works continuously. Every E_f epochs, all SPs share their network parameters by performing joint updates.

The federated learning matrix $\boldsymbol{\Omega}$ is obtained by using weight ω for each SPs own parameters and mixing the parameters of other SPs when learning as

$$\boldsymbol{\Omega} = \begin{bmatrix} \omega & \frac{1-\omega}{N-1} & \cdots & \frac{1-\omega}{N-1} \\ \frac{1-\omega}{N-1} & \omega & \cdots & \frac{1-\omega}{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-\omega}{N-1} & \frac{1-\omega}{N-1} & \cdots & \omega \end{bmatrix} \tag{13}$$

where ω is used as the SP federated learning factor. SPs keep their own actor network with weight ω and exchange the parameters of other SPs' actor networks with weight $(1-\omega)$. Based on the federated update, the SPs share their actor network parameters and update them without exchanging their data information. Through federated learning, all SPs can share the network parameters for cooperative model training, which will help SPs to find the optimal policy and complete the network training faster. At the same time, it also helps SPs optimize the bandwidth allocation and, thus, assist the edge agents in completing data transmission more efficiently, further improving the communication capability of the system.

Table I Main parameter settings for simulations

Parameter	Value	Parameter	Value
χ_{k_c}	0.3	P_{k_c}, P_i, P_x	0.4W
r_{imove}	6	ρ, τ, ϵ	0.85, 0.8, 0.2
B_i, ζ_1	5, 0.5	B_x, ζ_2	5, 0.5
r_i	40	T_u, E_f	8
B, ϱ	16, 0.0008	$\eta_{\mathcal{A}}$, $\eta_{\mathcal{C}}$	10^{-3}, 2 $\times 10^{-3}$
W_1, W_2, W_3	200MHZ	u, b	12, 0.135
Map size	100 X 100	ξ_i, ξ_x, ξ_j	10^{-22} W
$\lambda_{k_{EC}}$	300b/slot	$\lambda_{k_{RIA}}$	500b/slot
$\lambda_{k_{MAR}}$	1000b/slot	$\lambda_{k_{ICB}}$	800b/slot
k_B	1.38×10^{-23}J/K	$dis_{k_c,i}$	2000 m
φ_{LoS}	1	φ_{NLoS}	20
f_c, G	$2.4GHz$, 31.6	$dis_{i/j,x}$	160 km

V. Numerical Results

We have performed extensive simulations to illustrate the performance of our Fed-SATEC-Off algorithm, which is based on MADDPG (decentralized), and compared it with the conventional DDPG (centralized) [10]. The simulations were conducted in Python and Tensorflow. The simulation settings are summarized in Table I unless otherwise stated. We consider that transmissions are implemented in 2.4GHz frequency band. We have placed 80 IoT devices per traffic class and 5 UAVs on a 100×100 map. Given the map size, we have assumed one LEO can receive packet transmissions from all UAVs. Four typical traffic classes for satellite networks are studied:

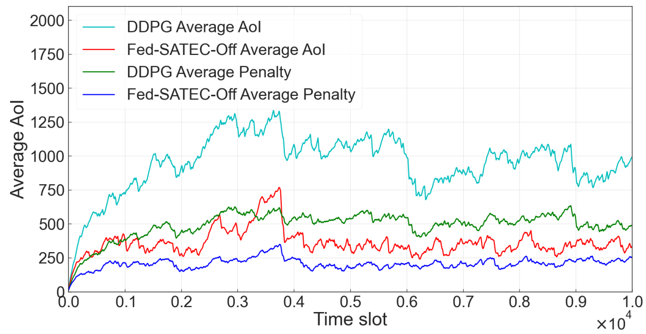

Fig. 3: Average AoI and Penalty.

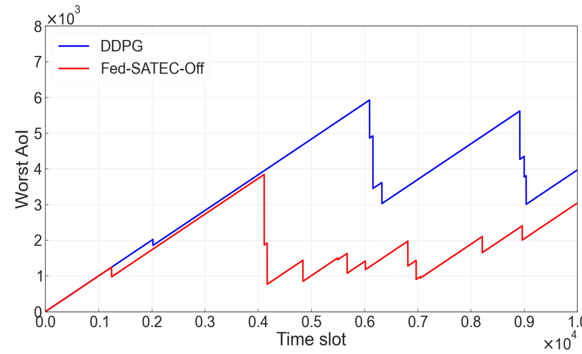

Fig. 4: Worst AoI vs no. epochs.

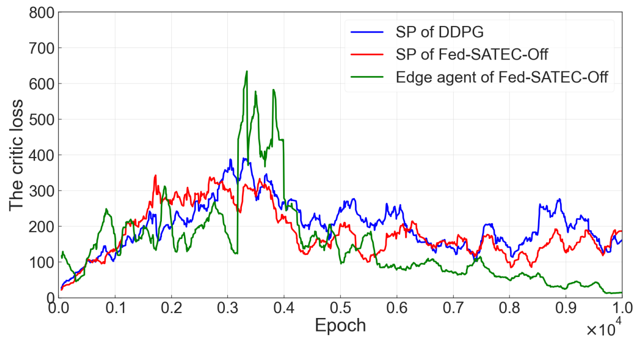

Fig. 5: Critic loss function for different agents.

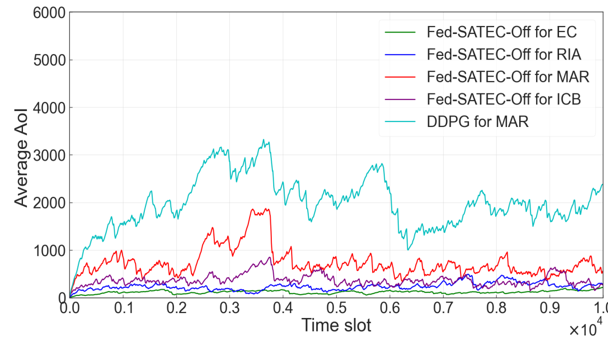

Fig. 6: AoI for different classes in Fed-SATEC-Off.

Emergency communications (EC), Remote industrial automation (RIA), Monitoring and reconnaissance (MAR), and In-space cellular backhaul (ICB).

The physical layer mode in each domain in SATEN is described below and the parameters are given in Table I. The channel gain between UAV i and LEO satellite x [13] is $h_{i,x} = \frac{G_i G_x \upsilon}{4\pi \mathrm{f_c} dis_{i,x} K_B}$ where G_i and G_x are the transmitter (UAV i) antenna gain and the receiver (LEO x) antenna gain, respectively; $dis_{i,x}$ is the distance between UAV i and LEO x, f_c is the carrier frequency, K_B is the Boltzmann's constant and υ is the speed of the light. The channel gain between LEO x and GS j [14] is $h_{x,j} = \frac{G_x G_j \upsilon}{4\pi \mathrm{f_c} dis_{x,j} K_B}$, and the channel gain between IoT device k_c and UAV i [15] is $h_{k_c,i} = 10^{-(L_{k_c,i}/10)}$, where $L_{k_c,i}$=$\mathrm{P}_{LoS}(t) \times \mathrm{PL}_{LoS}(t) + P_{NLoS}(t) \times \mathrm{PL}_{NLoS}(t)$, where $P_{LoS}(t) = \left(\frac{1}{1+ue^{(-b(-12+\sin^{-1}((100-\mathrm{u})/dis_{k_c,i})\ ))}}\right)$ is the probability of LoS, $PL_{LoS}(t) = \left(20\log(f_c) + 20\log\left(\frac{4\pi}{L}\right) + 20\log(dis_{k_c,i}(t)) + \varphi_{LoS}\right)$ is the LoS pathloss, $PL_{NLoS}(t) = \left(20\log f_c + 20\log\left(\frac{4\pi}{L}\right) + 20\log(dis_{k_c,i}(t)) + \varphi_{NLoS}\right)$ is the NLoS pathloss, $P_{NLoS}(t) = 1 - P_{LoS}(t)$. u and b are constants, which depend on the environment.

In Fig. 3, we present the average AoI and the penalty for Fed-SATEC-Off and DDPG. We can see that our algorithm reduces the average AoI by factor 3 compared to traditional DDPG. Similar improvement is obtained in terms of penalty. Although both algorithms achieved convergence after 1000 epochs, as shown in Fig. 5, our algorithm selects more efficiently the class of IoT devices to collect data and performs better allocation of the communication and computing resources. This translates into faster completion of the offloading tasks. In addition, our algorithm finds the optimum interaction strategy between edge agents and SPs, which results in the minimum penalty.

The training loss of the critic network for different agents in DDPG and Fed-SATEC-Off is presented in Fig.5. We observe that although the convergence trend of the critic network training loss for the center agent (a global SP) in DDPG and any SP in Fed-SATEC-Off is the same, both start to converge at 5000 epochs, the converged loss function value for Fed-SATEC-Off is consistently smaller and more stable than that of DDPG. This indicates that the network predicts the action taken in the current state more accurately. It also means that our Fed-SATEC-Off can predict the target more accurately and choose the optimal solution faster. At the same time, for the edge agents (UAVs/satellites) in Fed-SATEC-Off, we can also see that the training loss starts to converge after 4000 epochs. All of these indicate that our Fed-SATEC-Off has better learning performance.

The worst AoI is shown in Fig. 4. It indicates the maximum AoI achieved among all IoT devices at a particular time. Our Fed-SATEC-Off performs the best at all times. We can see that the AoI grows linearly in time. As the system learns to select and execute the packets with the largest AoI, there is a saw-tooth decline in the worst AoI. By comparing the falling time, we can see that Fed-SATEC-Off will adjust the strategy faster to solve the worst case due to the interactive strategies between agents.

In Fig. 6, we show the average AoI of the four traffic classes using Fed-SATEC-Off. We can see that EC has the smallest AoI among these four types of communications since it has the most stringent AoI requirements. We have observed that most of EC traffic is executed at the UAVs. On the other hand, MAR and ICB tolerate a higher AoI, which results in executing the tasks at LEOs and GS. In addition, MAR has the highest data generation rate of the four types, which causes the system to require more resources to process this type of data. We have observed that when comparing the performance of MAR in DDPG and Fed-SATEC-Off, the AoI is four times lower in Fed-SATEC-Off. This is because when there is more traffic demand finding the best allocation of resources is more critical, and thus the superiority of our scheme is more evident. With the other traffic classes, the improvement was about 3 times as in the overall average AoI, and it has been omitted for clarity of presentation.

VI. Conclusion

In this paper, we presented a cross-domain collaborative framework to optimize the AoI and energy efficiency trade-off in SATNs under different IoT traffic requirements. Our approach facilitates timely data collection, bandwidth, and offloading decisions across different administrative domains in a distributed manner. A cross-domain federated computation offloading algorithm (Fed-SATEC-Off) is developed based on a multi-agent actor-critic and incorporates federated learning to improve the convergence of the learning process. Our results show that Fed-SATEC-Off reduces the AoI by factor 4 and achieves faster convergence compared to existing approaches.

References

[1] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6g: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[2] H. H. Esmat, B. Lorenzo, and W. Shi, "Towards resilient network slicing for satellite-terrestrial edge computing iot," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022.

[4] S. G. Glisic and B. Lorenzo, *Artificial Intelligence and Quantum Computing for Advanced Wireless Networks*. John Wiley & Sons, 2022.

[5] https://economictimes.indiatimes.com/magazines/panache/elon-musks-30-bn-investment-plan-for-starlink-the-satellite-internet-unit-by-space/articleshow/83983229.cms?from=mdr.

[6] H. H. Esmat, X. Xia, B. Lorenzo, and L. Guo, "Multi-protocol aware federated matching for architecture design in heterogeneous iot," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 6277–6282.

[7] Z. Zhu, S. Wan, P. Fan, and K. B. Letaief, "Federated multiagent actor–critic learning for age sensitive mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1053–1067, 2022.

[8] H. Lee, S. Eom, J. Park, and I. Lee, "Uav-aided secure communications with cooperative jamming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9385–9392, 2018.

[9] Z. Gao, A. Liu, C. Han, and X. Liang, "Non-orthogonal multiple access-based average age of information minimization in leo satellite-terrestrial integrated networks," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1793–1805, 2022.

[10] H. Zhang, R. Liu, A. Kaushik, and X. Gao, "Satellite edge computing with collaborative computation offloading: An intelligent deep deterministic policy gradient approach," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[11] Y. Wang, S. Wu, J. Jiao, W. Wu, Y. Wang, and Q. Zhang, "Age-optimal transmission policy with harq for freshness-critical vehicular status updates in space–air–ground-integrated networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5719–5729, 2022.

[12] Y. Cai, S. Wu, J. Luo, J. Jiao, N. Zhang, and Q. Zhang, "Age-oriented access control in geo/leo heterogeneous network for marine iort: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 919–24 932, 2022.

[13] Z. Jia, M. Sheng, J. Li, D. Niyato, and Z. Han, "Leo-satellite-assisted uav: Joint trajectory and data collection for internet of remote things in 6g aerial access networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9814–9826, 2021.

[14] X. Wen, Y. Ruan, Y. Li, and R. Zhang, "Cognitive region design for overlay cognitive satellite terrestrial networks," *IEEE Communications Letters*, vol. 25, no. 1, pp. 244–248, 2021.

[15] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular uav-to-x communications: Design and optimization for multi-uav networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1346–1359, 2019.