Multi-protocol Aware Federated Matching for Architecture Design in Heterogeneous IoT

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

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000972

***Abstract*—Enabling timely data collection in heterogeneous IoT networks under different protocols and spectrum bands (e.g., WiFi, Bluetooth, Zigbee, LoRa) is crucial to implementing large-scale IoT systems. This paper presents a federated matching framework for heterogeneous IoT networks in which an intermediate layer of multi-protocol mobile gateways (M-MGs) is deployed by different service providers (SPs) to collect and relay data from IoT objects and perform computing tasks. The aim is to develop collaborative strategies between M-MGs and SPs to minimize the average weighted sum of the age-of-information and energy consumption. A novel collaborative framework based on a 2-level multi-protocol multi-agent actor-critic (MP-MAAC) is presented, where M-MGs and SPs can learn the interactive strategies through their own observations. The M-MGs strategies include the selection of IoT objects for data collection, execution, and offloading to SPs' access points while SPs decide on the spectrum allocation. Moreover, we incorporate federated matching (Fed-Match) into the multi-agent collaborative framework to improve the convergence of the learning process. The numerical results show that our Fed-Match algorithm reduces the AoI by factor 4, collects twice more packets than existing approaches and establishes design principles for the stability of the training process.**

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

I. Introduction

With the broad integration of wireless communications and the Internet of Things (IoT) in multiple types of surrounding applications, ensuring the flexibility of IoT deployment for data freshness becomes a challenging task [1], [2]. This is especially relevant given the device heterogeneity of IoT with different types of wireless protocols used for specific applications, e.g., WiFi supports real-time high definition video surveillance in traffic control, while NB-IoT is a better choice for smart agriculture due to its wider coverage and longer battery life. In fact, applications requiring large-scale IoT deployment (e.g., industrial IoT in a warehouse) will proactively aggregate multiple sources of data for better decision making [3], where a variety of data will be sensed by heterogeneous IoT devices and collected via different wireless protocols. However, existing works on architecture design for IoT data collection narrow their models down to a specific application domain [4, 5] and similar data freshness performance is analyzed for one protocol [6]. Further, many IoT protocols work on the same wireless spectrum bands, such as WiFi, Bluetooth, ZigBee, and LoRa on 2.4GHz, which will cause severe interference if the transmission schedule is not properly coordinated. Therefore, how to enable heterogeneous IoT data collection with different data patterns, coverage, protocols, dedicated spectrum bands, and caching capabilities becomes a pressing need to guarantee the freshness of data in large-scale IoT systems.

The age of information (AoI) has been used as a measure of the data freshness [7]-[9]. Recently, AoI was introduced to evaluate the performance of IoT applications that conduct complex tasks (e.g., artificial intelligence tasks [6]), requiring processing and computing to extract useful features. Kuang et al. [7] analyzed the average AoI of local computing, edge computing, and partial offloading in which part of the task is processed locally and the remaining remotely. Song et al. [8] proposed a metric called the age of task and developed joint partial offloading and scheduling algorithms in a multiuser network. Reinforcement learning (RL), especially multi-agent reinforcement learning (MARL) and federated learning (FL) have been adopted for distributed scenarios where agents learn interactive decisions through their local observations and collaborate to achieve global optimal strategies [10]. For cases where the action space is large, policy-based methods such as multi-agent Deep Deterministic Policy Gradient (MADDPG) are used [9]. Xie et al. [10] developed a Deep RL algorithm (DRL) to design offloading and scheduling policies to minimize the AoI and energy consumption in an IoT system. Zhu et al. [9] adopted MARL and FL to learn policies for trajectory planning of unmanned aerial vehicles (UAVs) and resource allocation. Despite the existing works related to AoI in edge-enabled IoT, policies for joint collaborative data collection, offloading, and spectrum allocation to minimize the AoI and energy cost in multi-protocol IoT networks have not been studied.

In this paper, we present a framework for architecture design in heterogeneous IoT networks in which an intermediate layer of multi-protocol mobile gateways (M-MGs) is deployed by different service providers (SPs) to collect and relay data from IoT objects and perform computing tasks. The M-MGs are assumed to be equipped with multiple wireless protocols needed for performing heterogeneous IoT data collection, and they also have computation and caching capabilities. The

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

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on February 01,2025 at 03:47:30 UTC from IEEE Xplore. Restrictions apply.

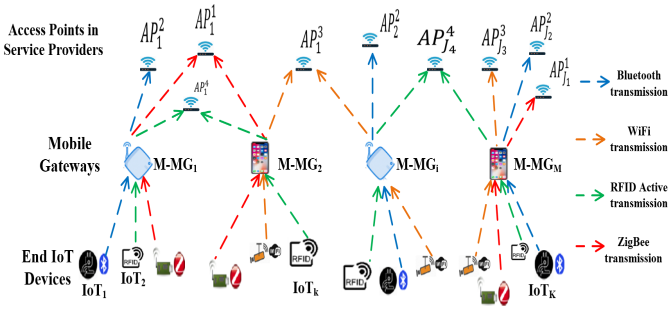

Fig. 1: Multi-protocol Federated IoT Architecture.

challenge lies in the uncertain spectrum availability, dynamic data demand, and mobility of M-MGs, which results in intermittent connectivity. To address these issues, a federated matching (Fed-Match) framework is presented to obtain collaborative strategies between M-MGs and SPs to minimize the average weighted sum of the age-of-information and energy consumption. The M-MGs strategies include the selection of IoT objects for data collection, execution, and offloading to SPs' access points while SPs perform spectrum allocation. Our approach significantly outperforms existing baselines and achieves better convergence.

II. System Model and Problem Formulation

A. Hierarchical IoT Network Architecture

We consider a hierarchical IoT network architecture, as shown in Fig. 1, operated by a set $\mathcal{Z} = \{1, ..., Z\}$ of IoT SPs denoted by $\{SP_z\}_{z=1}^{Z}$ who serve traffic demands from IoT devices denoted by $\{O_k^z\}_{k=1}^{K_z}$, $k \in \mathcal{K}_z = \{1, ..., K_z\}$ using wireless protocol z. Each SP_z owns J_z access points (APs) denoted by $\{AP_j^z\}_{j=1}^{J_z}$, $j \in \mathcal{J}_z = \{1, ..., J_z\}$, which have access to the SP's cloud computing center which stores data and implements centralized control. Moreover, there are M M-MGs denoted by $\{M - MG_i\}_{i=1}^{M}$, $i \in \mathcal{M} = \{1, ..., M\}$ that can operate in $Z_i \leq Z$ interfaces. For simplicity, we assume $Z_i = Z$. SPs incentivize M-MGs to collect data from each object O_k^z, perform local computing, and forward it to the corresponding AP_j^z. They can be commercial-off-the-shelf (COTS) gateways equipped with batteries, smartphones, and special designed software-defined radio (SDR), all of which are deployed with more than one wireless protocol in different or same wireless spectrum bands, such as WiFi, Bluetooth, ZigBee and LoRa on 2.4GHz, and have computing and caching capabilities.

B. Communication Model

M-MGs are equipped with Z interfaces to access different spectrum bands. To simplify the model, we assume M-MGs have only one radio for each interface and each M-MG can be shared by Z SPs. Each SP $z \in \mathcal{Z}$ has a total bandwidth W^z and each channel will be allocated a fraction $w_k^z(t)$ and $w_i^z(t)$ of the total bandwidth by frequency division mode for transmission between O_k^z and $M - MG_i$ and $M - MG_i$ transmission to AP_j^z, respectively. We model our communication system in a time-slotted manner. The power propagation gain from object O_k^z and M-MG i is $g_{ki} = \beta \cdot d_{ik}^{-\mathcal{X}}$, where β is an antenna-related parameter, $\mathcal{X}$ is the path loss factor, and r_{ik} is the distance between the two nodes. Thus, the link capacity c_{ki}^z from object O_k^z to $M - MG_i$ and the link capacity c_{ij}^z from $M - MG_i$ to AP_j^z are

$$\begin{aligned} c_{ki}^z(t) &= w_k^z(t) W^z log_2(1 + P_k \cdot g_{ki}/\xi_i) \\ c_{ij}^z &= w_i^z(t) W^z log_2(1 + P_i \cdot g_{ij}/\xi_j) \end{aligned} \tag{1}$$

where W^z is the bandwidth of channels on interface z, P_k is the transmission power of object O_k^z, ξ_i is the Gaussian noise power of M-MG i, P_i is the transmission power of M-MG i and ξ_j is the Gaussian noise power at AP_j^z. In our model, there is no data transmission between M-MGs. They only share states, observations, and learning parameters.

C. Data Collection, M-MG Processing and Offloading

Each IoT device O_k^z generates data independently with a data packet size d_k^z and elapsed time ψ_k^z. The data rate and arrival probability of data generation at O_k^z are λ_k and b_g^k, respectively. The generated packets are stored by each device locally until they are collected by M-MGs. We assume M-MGs move following a random mobility model [11] and collect the data from IoT devices when there are within their transmission range and there is an available channel. Each M-MG i has a collection data buffer of size $B_{i,col}^z$ and an execution data buffer of size $B_{i,exe}^z$ per interface $z \in \mathcal{Z}$. The collection data buffer caches data packets from each IoT device O_k^z which will be scheduled for local processing at M-MG i. The collection decision for M-MG i is $\textbf{collect}_i^z(t) = [col_1^z(t), \cdots, col_{B_{i,col}^z}^z(t)]$, where $col_q^z(t) \in \{0, 1\}$.

At each time t, each M-MG i collects a data packet from IoT device O_k^z on interface $z \in \mathcal{Z}$

$$\sum\nolimits_{q=1}^{B_{i,col}^z} col_q^z(t) = 1. \tag{2}$$

Similarly, each M-MG i makes a decision to execute a packet from the collection buffer per interface $z \in \mathcal{Z}$ at each time t, $\textbf{execution}_i^z(t) = \left[exe_1^z(t), \ ..., \ exe_{B_{i,col}^z}^z(t)\right]$, and

$$\sum\nolimits_{q=1}^{B_{i,col}^z} exe_q^z(t) = 1 \tag{3}$$

where $exe_q^z(t) \in \{0, 1\}$. The computation time for executing a packet of size d_k^z in M-MG i with CPU frequency f_i is

$$\tau_{ki}^z(t) = d_k^z(t)/f_i \tag{4}$$

After local execution, the data packet is stored in the execution buffer until the M-MG i offloads it to AP_j^z. The offloading decision for each packet is $\textbf{offload}_{i,j_z}(t) = \left[off_{1,j_z}(t), \ ..., \ off_{B_{i,exe,J_z}^z}(t)\right]$ with

$$\sum\nolimits_{q=1}^{B_{i,exe}^z} off_{q,j_z}(t) = 1 \tag{5}$$

Consequently, a M-MG i either collects, execute or offload data form each interface z at any time t.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on February 01,2025 at 03:47:30 UTC from IEEE Xplore. Restrictions apply.

III. Problem Formulation

The AoI is a performance metric that measures the freshness of the data at the receiver side. It is defined as the difference between the current time t and the generation time $t_{g,k}^z$ of the latest data packet from device O_k^z received at any of the corresponding APs

$$A_k^z(t) = t - t_{g,k}^z \tag{6}$$

The AoI increases linearly with time until a new packet is received. Our goal is to minimize the overall average AoI and the energy cost. The energy cost $cost_{ik}^z$ of M-MG i to serve IoT device O_k^z is

$$cost_{ik}^z = d_k^z E_i^{exe} + (d_k^z + \Delta_k) E_i^{off} \tag{7}$$

which is proportional to the energy cost to execute the data of IoT device O_k^z with size d_k^z and E_i^{exe} is the energy needed to execute one data bit by M-MG i, and proportional to the energy consumed to offload the data. Note that after the M-MG i executed the data of IoT device O_k^z, the data size increases by Δ_k, and E_i^{off} is the energy cost of offloading one data bit from M-MG i. Based on the previous definitions, the optimization problem is as follows

$$\min_{\boldsymbol{y},\boldsymbol{\beta}_z,\boldsymbol{\gamma},\boldsymbol{w}} \sum_{z=1}^{Z}\sum_{i=1}^{M}\sum_{k=1}^{K}\sum_{j_z=1}^{J_z} y_{zi}\gamma_{ik}\{\beta_{ij_z}\zeta_1 A_k^z(t) + \varrho\zeta_2 cost_{ik}^z(t)\}/K \tag{8}$$

subject to

$$(1)-(5)$$

$$\sum\nolimits_{z\in\mathcal{Z}} y_{zi} \le Z, \forall i \in \mathcal{M} \tag{8.a}$$

$$\sum\nolimits_{i\in\mathcal{M}} y_{zi} \le M, \forall z \in \mathcal{Z} \tag{8.b}$$

$$\sum\nolimits_{z\in\mathcal{Z}}\sum\nolimits_{j_z\in\mathcal{J}_z} \beta_{ij_z} \le Z \times J, \forall i \in \mathcal{M} \tag{8.c}$$

$$\sum\nolimits_{i\in\mathcal{M}} \beta_{ij_z} \le M, \forall z \in \mathcal{Z}, j_z \in \mathcal{J}_z \tag{8.d}$$

$$\sum\nolimits_{k=1}^{K} \gamma_{ik} \le Z, \forall i \in \mathcal{M} \tag{8.e}$$

$$\sum\nolimits_{i\in\mathcal{M}} \gamma_{ik} \le 1, \forall k \in \mathcal{K} \tag{8.f}$$

$$\sum\nolimits_{k\in\mathcal{K}_z} w_k^z + \sum\nolimits_{i\in\mathcal{M}} w_i^z \le W^z, \forall i \in \mathcal{M}, \forall k \in \mathcal{K} \tag{8.g}$$

$$y_{zi}, \beta_{ij_z}, \gamma_{ik} = \{0,1\}, \forall z, i, k, j_z \tag{8.h}$$

where K, M and Z are the number of IoT devices, M-MGs and SPs in the network, respectively. ζ_1 and ζ_2 represent the weighting factors for the AoI $A_k^z(t)$ and cost $cost_{ik}^z$, respectively, with $\zeta_1 + \zeta_2 = 1$, and ϱ is a scaling factor. $\boldsymbol{y} = [y_{zi}]_{(Z\times M)}$ with $y_{zi} \in \{0,1\}$ indicates that SP z is associated with M-MG i when $y_{zi} = 1$. (8.a) and (8.b) state that each M-MG can serve Z SPs and each SP can serve M M-MGs, respectively. $\boldsymbol{\beta}_z = [\beta_{ij_z}]_{(M\times J_z)}$ with $\beta_{ij_z} \in \{0,1\}$ indicates that M-MG i is associated with AP_j^z when $\beta_{ij_z} = 1$. (8.c) and (8.d) state that each M-MG can transmit to $(J \times Z)$ APs and each AP can serve M M-MGs, respectively. $\boldsymbol{\gamma} = [\gamma_{ik}]_{(M\times K)}$ where γ_{ik} is a binary variable representing the association status between the IoT device O_k^z and M-MG i. The number of IoT devices that can be associated with M-MG i is constrained as in (8.e) and (8.f) and Z is defined as a quota that represents the maximum number of IoT devices that can be supported by M-MG i. (8.f) guarantees that each IoT device can be associated with at most one M-MG at a time. $\boldsymbol{w}$ denotes the bandwidth and (8.g) constrains the allocated bandwidth to object k and M-MG i not to exceed the available bandwidth W^z.

Solving the previous optimization at every time instant will not result in the optimum solution. In fact, the dynamics and the coupling of M-MGs and SPs in our 2-level multi-protocol IoT architecture make the problem NP-hard. For these reasons, we reformulate the optimization problem as a Markov Decision Process (MDP) and solve it with a new online iterative algorithm.

IV. Multi-protocol Federated Matching Framework

We model the optimization problem in (8) as a 2-level MDP to capture the interactions between SPs and M-MGs and we adopt multi-agent reinforcement learning to solve it. To overcome the complexity of searching a large state space when the size of the network increases, policy-based methods such as Advantage Actor-Critic (A2C), DDPG, and MADDPG are used, which rely on dual neural networks to estimate the action-value function $Q(\boldsymbol{s}, \boldsymbol{a})$ [9]. Therefore, we present a 2-level multi-protocol multi-agent actor-critic (MP-MAAC) for SP and M-MG collaboration and an online Fed-Match to improve the convergence.

A. MDP

1) States: The state of each M-MG agent $\mathbf{s}_i(t)$ contains the local observations of the environment (devices info), the status of the M-MG including buffer states, allocated AP_{j_z}, allocated bandwidth to collect data from IoT devices, and allocated offloading bandwidth to relay the collected data to APs. The state of each SP agent $\mathbf{s}_z(t)$ includes the status of all M-MGs associated with this SP.

2) Actions: The M-MGs collect data as they move, execute the data locally and offload it to the corresponding AP_{j_z}, $\mathbf{a}_i^z(t) = [\mathbf{collect}_i^z(t), \mathbf{execute}_i^z(t), \mathbf{offload}_{i_{j_z}}(t)]$.

The action of the SP, $\mathbf{a}^z(t) = [\boldsymbol{w}_i^z(t), \boldsymbol{w}_k^z(t)]$, consist of allocating bandwidth to M-MGs to offload data and collect data from IoT devices, respectively.

3) Penalty: Since there is collaboration among agents to minimize AoI and cost, all agents share the global penalty. The current penalty at t-th slot for each agent is $p_g(t) = \zeta_1\overline{\mathrm{A}}(\mathrm{t}) + \varrho\zeta_2\overline{\mathrm{cost}}(\mathrm{t})$, $\forall\ g = 1,..,M+Z$. To explore the global optimization of system, we set the long-term penalty as $P_g(t) = \sum_{l=0}^{T}\rho^l p_g(t+l)$, where T is the length of the time window and $\rho \in [0,1]$ is the penalty decay.

4) Transition Policies: In our multi-protocol IoT architecture, it is difficult to obtain a formatted strategy to cover all the state transitions of IoT devices, M-MGs, SPs, and spectrum allocation. Therefore, to represent the interactions

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on February 01,2025 at 03:47:30 UTC from IEEE Xplore. Restrictions apply.

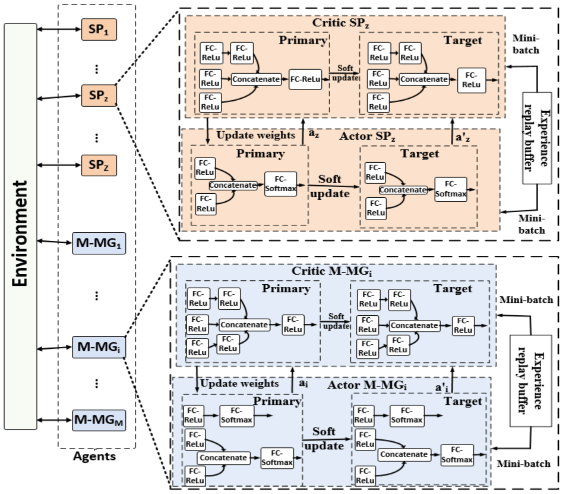

Fig. 2: Fed-Match Learning Collaboration Architecture.

among entities in the network, we use $\mathcal{T}(\{\boldsymbol{s}_i(t+1)\}, \boldsymbol{s}_z(t+1)|\{\boldsymbol{s}_i(t)\}, \boldsymbol{s}_z(t), \{\boldsymbol{a}_i(t)\}, \boldsymbol{a}_z(t))$.

B. Multi-protocol Fed-Match

We design a multi-protocol multi-agent actor-critic network (MP-MAAC) that contains the primary actor/critic networks and the target actor/critic networks to control the M-MGs and find the SPs' optimal bandwidth allocation. The MP-MAAC architecture is illustrated in Fig.2, where each agent (M-MG or SP) interacts with the environment to learn the optimal action that minimizes the system penalty P_g. The experience reply buffer (with capacity B) and target networks are used to deal with the instability issue of the approximated values of Q. For M-MG agents, a multiple input-output neural network is built to learn the different actions in the states that include the observation of IoT devices, offloading channel states, and the buffer states. We use the multilayer perceptrons (MLPs) for offloading scheduling and data execution at M-MGs. For SP agents, the state contains the buffer states of all M-MGs and the output is the bandwidth allocation. We adopt a ϵ-greedy policy to enforce random actions with probability ϵ. The parameters of primary actor-network ($\mathcal{A}_g$), primary critic-network ($\mathcal{C}_g$), target actor-network ($\mathcal{A}'_g$) and target critic-network ($\mathcal{C}'_g$) of g-th learning agent are $\boldsymbol{\theta}_g$, $\boldsymbol{\phi}_g$, $\boldsymbol{\theta}^{'}_g$ and $\boldsymbol{\phi}^{'}_g$, respectively. The parameters of target actor and critic networks are updated by the primary networks every T_u period as

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

where $\tau \in [0,1]$ is mixing weight. The learning rates of actor network and critic network are $\eta_{\mathcal{A}}$ and $\eta_{\mathcal{C}}$, respectively. The critic networks are updated by minimizing the mean squared error loss function

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

where $\hat{y}_g = p_g + \rho\mathcal{C}'_g(\boldsymbol{s}'_g, \boldsymbol{a}'_g, \boldsymbol{\phi}'_g)$ and $\hat{y}_g$ is the estimated long-time Q value, p_g is the penalty for each agent. Since we aim at minimizing the penalty, the loss function of actor networks can be written as follows

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

Algorithm 1 Multi-protocol Fed-Match Online Collaboration

```
Initialize: Hyper parameters of learning algorithms, the primary networks' parameters $(\boldsymbol{\theta}_{M-MG})_i, (\boldsymbol{\theta}_{SP})_z$, and target networks' parameters: $(\boldsymbol{\theta}_{M-MG})'_i \leftarrow (\boldsymbol{\theta}_{M-MG})_i, (\boldsymbol{\theta}_{SP})'_z \leftarrow (\boldsymbol{\theta}_{SP})_z$.
for epoch $t = 1$ to max_epoch do
    Generate $\nu \in [0,1]$ randomly;
    for epoch agent $g$ in $\{1, ...., M, .., M+Z\}$ do
        if $\nu < \epsilon$ or $|\boldsymbol{\mathcal{B}}[g]| < B$ then
            Choose actions $\boldsymbol{a}_g(t)$ randomly;
        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
    end for
    Interact with environment and obtain $p(t), \boldsymbol{s'}(t+1)$;
    Add $\{\boldsymbol{s}, \boldsymbol{a}, p, \boldsymbol{s'}\}$ into $\boldsymbol{\mathcal{B}}$;
    for epoch agent $g$ in $[1, ..., M..., M+Z]$ do
        if $|\boldsymbol{\mathcal{B}}[g]| \geq B$ then
            for each agent $g$ in $\{1, ..., M, ...., M+Z\}$ do
                Sample $\{\boldsymbol{s}_g, \boldsymbol{a}_g, p_g, \boldsymbol{s}'_g\}$ from $\boldsymbol{\mathcal{B}}[g]$;
                Predict new actions: $\boldsymbol{a}'_g = \mathcal{A}'_g(\boldsymbol{s}'_g; \boldsymbol{\theta'}_g)$ ;
                Predict new $Q$-value:
                $Q'(\boldsymbol{s}'_g, \boldsymbol{a}'_g) = \mathcal{C}'_g(\boldsymbol{s}'_g, \boldsymbol{a}'_g; \boldsymbol{\phi}'_g)$;
                Calculate $\hat{y}_g$;
                Calculate $l_{\mathcal{C}g}(\boldsymbol{\phi}_g), l_{\mathcal{A}g}(\boldsymbol{\theta}_g)$ by (10) and (11);
                Update network parameters:
                    $\boldsymbol{\phi}_g^{t+1} \leftarrow \boldsymbol{\phi}_g^t - \eta_{\mathcal{C}} \bigtriangledown_{\phi} \tilde{l}_{\mathcal{C}g}(\boldsymbol{\phi}_g^t)$
                    $\boldsymbol{\theta}_g^{t+1} \leftarrow \boldsymbol{\theta}_g^t - \eta_{\mathcal{C}} \bigtriangledown_{\theta} \tilde{l}_{\mathcal{C}g}(\boldsymbol{\theta}_g^t)$
            end for
        end if
    end for
    if $t$ mod $T_u == 1$ then
        Update target actor and critic networks using (9);
    end if
    if $t$ mod $E_f == 1$ then
        Run MG-federated updating using (12);
        Run SP-federated updating using (13);
    end if
end for
```

Under the proposed updating rule, each agent preserves the parameters with weight ω and mixes the others' parameters, which can be formulated by

$$\boldsymbol{\theta}^{t+1}_{M-MG} = \boldsymbol{\theta}^{t}_{M-MG}.\boldsymbol{\Omega}_1 \tag{12}$$

$$\boldsymbol{\theta}^{t+1}_{SP} = \boldsymbol{\theta}^{t}_{SP}.\boldsymbol{\Omega}_2 \tag{13}$$

where $\boldsymbol{\theta}^t_{M-MG} = [\boldsymbol{\theta}^t_1, .., \boldsymbol{\theta}^t_i, .., \boldsymbol{\theta}^t_M]$ and $\boldsymbol{\theta}^t_{SP} = [\boldsymbol{\theta}^t_1, .., \boldsymbol{\theta}^t_z, .., \boldsymbol{\theta}^t_Z]$ denote the vector of all M-MG actor networks and the vector of all SP actor networks at the t-th learning epoch, respectively, $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ denote the federated updating matrix of M-MG and SP, respectively. Based on the proposed learning framework, we develop the corresponding Fed-Match online collaboration as in Algorithm 1 where the

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on February 01,2025 at 03:47:30 UTC from IEEE Xplore. Restrictions apply.

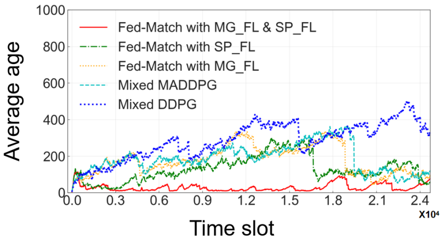

Fig. 3: Average AoI

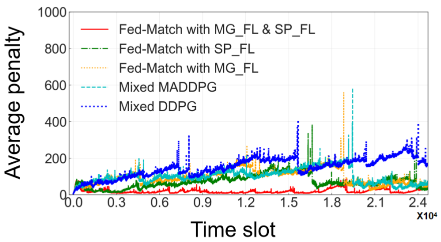

Fig. 4: Average Penalty

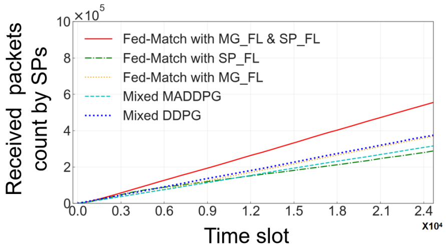

Fig. 5: Packets received vs no. epochs

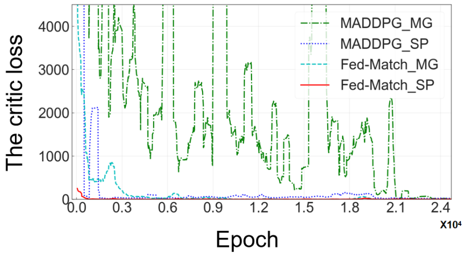

Fig. 6: Agent Critic loss

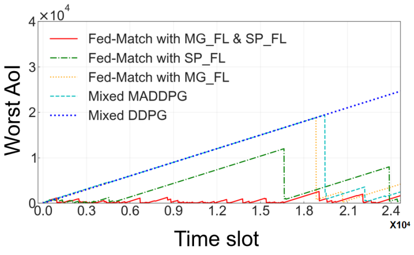

Fig. 7: Worst AoI vs no. epochs

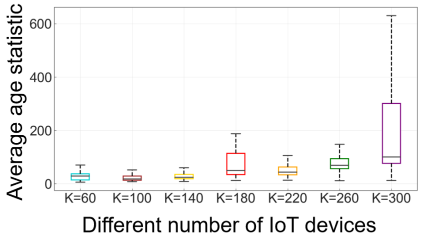

Fig. 8: Average AoI vs no. IoT devices

agents learn and update the optimal policies while the system works continuously. Every E_f learning epoch, all M-MG agents share their actor-network parameters and perform the federated updating. At the same time, all SP agents also share their actor-network parameters to conduct the federated updating. The $\mathbf{\Omega}$ is used as the federated learning factor, which means that each agent keeps the parameters with weight ω and mixes the other parameters when learning. The federated learning factors $\mathbf{\Omega}_1$ for M-MGs and $\mathbf{\Omega}_2$ for SPs are defined by using the following equation

$$\mathbf{\Omega}_n = \begin{bmatrix} \omega_n & \dfrac{1-\omega_n}{N-1} & \cdots & \dfrac{1-\omega_n}{N-1} \\ \dfrac{1-\omega_n}{N-1} & \omega_n & \cdots & \dfrac{1-\omega_n}{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{1-\omega_n}{N-1} & \dfrac{1-\omega_n}{N-1} & \cdots & \omega_n \end{bmatrix} \qquad (14)$$

where $n = 1$ and $N = M$ for M-MG federated learning and $n = 2$ and $N = Z$ for SP federated learning. The whole system acts as a cooperative model. To achieve global optimality using federated learning, agents share their actor net parameters and perform federated updating. Thus there is no need to exchange data messages and the communication cost of sharing model-level parameters can be neglected. The ω_1 is used as the M-MG federated learning factor, and during the learning period, M-MGs will retain the parameters with weights ω_1 and exchange the network parameters with weights $(1 - \omega_1)$. By exchanging network parameters among M-MGs, the learning convergence is improved, and is faster to obtain the offloading policy that results in the minimum penalty.

Likewise, each SP outputs the bandwidth ratios for each M-MG and per interface to perform bandwidth allocation. SPs also exchange model parameters. In particular, ω_2 is used as the SP federated learning factor. During the learning period, SPs will keep the parameters with weights ω_2 and exchange network parameters with weights $(1 - \omega_2)$. By doing so, the SP's policy training is accelerated to allocate the bandwidth more rationally for each M-MG's interface. This will reduce packet stagnation at M-MGs to avoid increasing the AoI. Since only the parameters of the lightweight behavioral network are transmitted, the communication efficiency of the system improves.

V. Numerical Results

We have conducted extensive simulations to illustrate the performance of our proposed architecture and Fed-Match algorithm and compare them with popular reinforcement learning algorithms, i.e., DDPG and multi-agent DDPG (MADDPG) [9], and with two modified versions of our Fed-Match with only one level of collaboration between M-MGs (Fed-Match MG-FL) and one level between SPs (Fed-Match SP-FL). Unless otherwise stated, the simulation parameters are given in Table I. We set the IoT environment on a 100 x 100 map with 100 to 300 IoT devices using Zigbee, Cellular, WiFi, and LoRa protocols (i.e., 4 SPs), and 7 M-MGs with 4 interfaces each.

As shown in Fig. 3, the average AoI with Fed-Match, which implements FL at both M-MGs and SPs, achieves the lowest AoI, fastest convergence, and lowest variance of the results. On the other hand, the average AoI with DDPG is 40 times higher and has a very slow convergence (when set up to 25000 epochs). If Fed-Match is implemented considering only FL for either SPs or M-MGs, we can see that the algorithm converges after 16500 or 19000 epochs, respectively. Therefore, as we anticipated, using our Fed-Match can effectively speed up agent learning and achieves learning stability. The penalty based on AoI and energy cost is shown in Fig. 4. We can see that Fed-Match achieves the minimum penalty, which means

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on February 01,2025 at 03:47:30 UTC from IEEE Xplore. Restrictions apply.

Table I Main parameter settings for simulations

Parameter	Value	Parameter	Value
λ_k, b_g^k	1Kb/slot, 0.3	$P^i_{tr,max}$	0.2W
r^i_{move}, r^i_{obs}	6, 60	ρ, τ, ϵ	0.85, 0.8, 0.2
$r^i_{collect-Zigbee}$	25	$B^z_{i,col}$, ζ_2	5, 0.5
$r^i_{collect-Cellular}$	40	$B^z_{i,exe}$, ζ_1	5, 0.5
$r^i_{collect-WiFi}$	60	T_u, E_f	8
$r^i_{collect-LoRa}$	80	$\eta_{\mathcal{A}}$, $\eta_{\mathcal{C}}$	1 $\times 10^{-3}$, 2 $\times 10^{-3}$
W_{Zigbee}, $W_{Cellular}$	5KHZ, 100KHZ	B, ϱ, $\mathcal{X}$	32, 0.0008, 4
W_{WiFi}, W_{LoRa}	20KHZ, 10KHZ	ξ_i, ξ_j	–174 dBm/Hz

Table II A numerical comparison on AoI with different ω_1 and ω_2.

Approach	Average age		Average penalty	
	mean	std	mean	std
Mixed DDPG	285.22	99.81	146.67	50.21
Mixed MADDPG	184.90	76.46	96.54	38.64
Fed-Match with MG-FL & SP-FL(ω_1=0.5,ω_2=0.5)	**28.60**	**20.50**	**18.40**	**10.40**
Fed-Match with MG-FL(ω_1=0.5)	176.31	76.27	92.25	38.51
Fed-Match with SP-FL(ω_2=0.5)	128.55	68.33	68.37	34.48
Fed-Match with MG-FL & SP-FL(ω_1=0.15,ω_2=0.25)	162.97	50.14	85.56	25.71
Fed-Match with MG-FL & SP-FL((ω_1=0.15,ω_2=0.5)	65.01	35.71	36.61	18.17
Fed-Match with MG-FL & SP-FL(ω_1=0.15,ω_2=0.8)	70.10	37.96	39.15	19.34
Fed-Match with MG-FL & SP-FL(ω_1=0.25,ω_2=0.25)	151.45	59.06	79.81	30.05
Fed-Match with MG-FL & SP-FL(ω_1=0.25,ω_2=0.5)	82.90	55.50	45.56	28.04
Fed-Match with MG-FL & SP-FL(ω_1=0.25,ω_2=0.8)	242.41	85.77	125.30	43.49
Fed-Match with MG-FL & SP-FL(ω_1=0.5,ω_2=0.25)	232.30	63.37	120.24	32.43
Fed-Match with MG-FL & SP-FL(ω_1=0.5,ω_2=0.8)	51.46	39.05	29.84	19.72
Fed-Match with MG-FL & SP-FL(ω_1=0.8,ω_2=0.25)	169.49	46.81	88.84	24.17
Fed-Match with MG-FL & SP-FL(ω_1=0.8,ω_2=0.5)	82.19	78.77	45.20	39.60
Fed-Match with MG-FL & SP-FL(ω_1=0.8,ω_2=0.8)	154.83	72.17	81.51	36.51

that the SPs can learn to allocate bandwidth more efficiently and the M-MGs can collect and offload data faster. Also, this shows the superiority of having interactive policies between M-MGs and SPs to minimize the penalty. In addition, Fig. 5 shows the overall number of the packets received by all APs and all SPs. Fed-Match can offload 3 times more packets than existing schemes, which indicates that the efficiency of the system is improved.

In Fig. 6, the training loss of the critic networks for M-MGs and SPs in MADDPG and Fed-Match are shown. We can observe that the critic loss for M-MGs and SPs in Fed-Match converges almost immediately, which means that the optimal strategy is found faster when we implement federated learning among M-MGs and among SPs simultaneously. Similar behavior was observed in the actor-network but the results are omitted in the interest of space. The worst AoI is shown in Fig. 7 for all algorithms. The worst AoI refers to the maximum AoI of all IoT devices at the current time. Our Fed-Match continues to be the best and its worst AoI remains stable and the lowest at any time. Our architecture design based on a two-layer federated learning considers all IoT devices and M-MGs when allocating bandwidth for data offloading to avoid data stagnation. At the same time, we can see that DDPG is unable to offload data for a long time leading to an increase in AoI. This is because DDPG requires larger neural network models with a complex structure to learn the relationship between the global input state and each agent's local policy. This complicates and slows down training. Next, we increase the number of sensors from 60 to 300 and evaluate the performance of Fed-Match. For the same available bandwidth and number of M-MGs, the average AoI is consistently low up to 260 sensors. To further reduce the AoI, more M-MGs need to be deployed.

In Table II, we compare the convergence of the Fed-Match algorithm for different values of the weights ω_1 and ω_2 of the FL in the M-MGs and the SPs, respectively. We have found that best performance is obtained when $\omega_1 = \omega_2 = 0.5$. We have also noticed that the best convergence does not translate into the best performance due to the influence of multiple random variables, i.e., the best performance is obtained when ω_1 and ω_2 lie on the interval $[1/Number\_agents, 0.5]$. This is because online collaboration benefits from the fluctuation of the gradients since the actor parameters $\boldsymbol{\theta}$ can adapt to the changes in the environment. For instance, when ω_1 is too large, each MG relies mainly on its own learning strategy and requires longer training to converge. However, when the value of ω is too small, the agents will lose insights into the performance from their own parameters.

VI. Conclusion

In this paper, we presented a multi-protocol IoT architecture design to enable timely data collection in heterogeneous IoT networks under different protocols and spectrum bands. We developed collaborative policies for data scheduling and bandwidth allocation between M-MGs and SPs to minimize the average AoI and energy consumption. The policies are based on a new federated matching framework. Our results showed a significant reduction in the AoI and better convergence, learning stability, and system efficiency than existing schemes.

References

[1] Cisco forecast. Available: https://blogs.cisco.com/networking/iot-and-the-network-what-is-the-future.

[2] B. Lorenzo, F. J. González-Castaño, L. Guo, F. Gil-Castiñeira and Y. Fang, "Autonomous Robustness Control for Fog Reinforcement in Dynamic Wireless Networks," in IEEE/ACM Transactions on Networking, vol. 29, no. 6, pp. 2522-2535, Dec. 2021.

[3] Y. Liu, M. Kashef, K. B. Lee, L. Benmohamed, and R. Candell, "Wireless Network Design for Emerging IIOT Applications: Reference Framework and Use Cases," Proceedings of the IEEE, vol. 107, no. 6, pp. 1166–1192, 2019.

[4] T. H. Laine, C. Lee, and H. Suk, "Mobile Gateway for Ubiquitous Health Care System Using Zigbee and Bluetooth," in 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2014, pp. 139–145.

[5] A. P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi, "Architecture and protocols for the Internet of Things: A case study," in Proc. 8th IEEE Int. Conf. Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, pp. 678–683.

[6] X. Xie, H. Wang and M. Weng, "A Reinforcement Learning Approach for Optimizing the Age-of-Computing-Enabled IoT," IEEE Internet of Things Journal, vol. 9, no. 4, pp. 2778-2786, 2022.

[7] Q. Kuang, J. Gong, X. Chen, and X. Ma, "Analysis on Computation Intensive Status Update in Mobile Edge Computing," IEEE Trans. Veh. Technol., vol. 69, no. 4, pp. 4353–4366, Apr. 2020.

[8] X. Song, X. Qin, Y. Tao, B. Liu, and P. Zhang, "Age Based Task Scheduling and Computation Offloading in Mobile-Edge Computing Systems," in Proc. IEEE WCNCW, Marrakech, Morocco, 2019, pp. 1–6.

[9] Z. Zhu, S. Wan, P. Fan and K. B. Letaief, "Federated Multiagent Actor–Critic Learning for Age Sensitive Mobile-Edge Computing," in IEEE Internet of Things J., vol. 9, no. 2, pp. 1053-1067, Jan., 2022.

[10] X. Xie, H. Wang and M. Weng, "A Reinforcement Learning Approach for Optimizing the Age-of-Computing-Enabled IoT," IEEE Internet of Things Journal, vol. 9, no. 4, pp. 2778-2786, 2022.

[11] Kerdsri J, Veeraklaew T. "Visualization of Spatial Distribution of Random Waypoint Mobility Models." J. Comput. 2017, 12(4): 309-316.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on February 01,2025 at 03:47:30 UTC from IEEE Xplore. Restrictions apply.