

Cross-Technology Federated Matching for Age of Information Minimization in Heterogeneous IoT

Haitham H. Esmat¹, Member, IEEE, Xiaohao Xia, Graduate Student Member, IEEE, Yinxuan Wu², Member, IEEE, Beatriz Lorenzo³, Senior Member, IEEE, and Linke Guo⁴, Senior Member, IEEE

Abstract—Heterogeneous Internet of Things (IoT) networks, which operate using various protocols and spectrum bands like WiFi, Bluetooth, Zigbee, and LoRa, bring many opportunities to collaborate and achieve timely data collection. However, several challenges must be addressed due to heterogeneous data patterns, coverage, spectrum bands, and mobility. This paper introduces a cross-technology IoT network architecture design that facilitates collaboration between service providers (SPs) to share their spectrum bands and offload computing tasks from heterogeneous IoT devices using multi-protocol mobile gateways (M-MGs). The objective is to minimize the age of information (AoI) and energy consumption by jointly optimizing collaboration between M-MGs and SPs for bandwidth allocation, relaying, and cross-technology data scheduling. A pricing mechanism is presented to incentivize different levels of collaboration and matching between M-MGs and SPs. Given the uncertainty due to mobility and task requests, we design a cross-technology federated matching algorithm (CT-Fed-Match) based on a multi-agent actor-critic approach in which M-MGs and SPs learn their strategies in a distributed manner. Furthermore, we incorporate federated learning to enhance the convergence of the learning process. The numerical results demonstrate that our CT-Fed-Match-RC algorithm with cross-technology and relaying collaboration reduces the AoI by 30 times and collects 8 times more packets than existing approaches.

Index Terms—Age of information (AoI), cross-technology offloading, federated learning, heterogeneous IoT, multi-agent deep reinforcement learning, mobile edge computing.

I. INTRODUCTION

WITH the increasing development of the Internet of Things (IoTs), IoT systems are becoming more intelligent and heterogeneous with applications requiring data from devices that use different wireless protocols like WiFi, Bluetooth, Zigbee, and LoRa [1]. Data collection and computing in such a heterogeneous IoT network become more challenging to coordinate and achieve data freshness. Moreover, many

IoT protocols may operate within the same wireless spectrum bands, and thus, without proper coordination of transmission schedules, this overlap can result in significant performance degradation. Theodorou et al. [2] present a multi-protocol Software Defined Network (SDN) platform for multiple services and heterogeneous IoT devices. Hasan and Khan [3] survey the latest works on coexistence spectrum management with a focus on interference management when multiple wireless networks are in close proximity. Cross-technology communications are proposed in [4] to facilitate direct communication among different wireless technologies when they operate in the same spectrum band. Jiang et al. [5] present BlueBee that emulates a ZigBee frame within the payload of a Bluetooth packet. In [6], they present a spectrum allocation strategy CoWBee in which WiFi is used to communicate with ZigBee devices directly, and both devices schedule their transmissions to avoid collisions and reduce interference. However, these works ignore the data freshness.

The idea of data freshness, evaluated through the AoI, has been extensively employed as a performance metric in IoT [7], [8], [9], [10], [11], [12], [13], [14], [15]. Recent works on minimizing AoI rely on mobile edge computing (MEC) for data collection and computation task offloading [7], [11], [14], as summarized in Table I. Most of their network architecture designs focus on evaluating data freshness for a single protocol [16], and the same applies to pricing schemes that incentivize data collection and control the data freshness [17], [18], [19], [20]. On the other hand, the diversity of protocols brings unique opportunities for cross-technology collaboration that remain unexplored. Consequently, there is a need to enable heterogeneous IoT data collection and computational offloading that considers diverse data patterns, coverage requirements, spectrum bands, and caching capabilities, all with the ultimate goal of ensuring data freshness in heterogeneous IoT systems.

In dynamic and distributed scenarios, multi-agent deep reinforcement learning (MADRL) and federated learning (FL) have been utilized to enable agents to learn from local observations and collaborate towards achieving optimal global strategies [11], [16]. To handle large action spaces, policy-based methods like multi-agent deep deterministic policy gradient (MADDPG) are employed [15]. Xie et al. [16] develop a deep reinforcement learning algorithm for trajectory planning of unmanned aerial vehicles (UAVs) and resource allocation in an IoT network. In our preliminary work [11],

Manuscript received 14 November 2023; revised 10 May 2024; accepted 28 July 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Kompella. This work was supported in part by U.S. National Science Foundation under Grant CNS-2008309, Grant CNS-2008049, and Grant CCF-2312616. (Corresponding author: B. Lorenzo.)

Haitham H. Esmat, Xiaohao Xia, Yinxuan Wu, and Beatriz Lorenzo are with the Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA 01002 USA (e-mail: habdelhafez@umass.edu; xiaohaoxia@umass.edu; yinxuanwu@umass.edu; blorenzo@umass.edu).

Linke Guo is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA.

Digital Object Identifier 10.1109/TNET.2024.3436712

we adopt MADDPG to solve the data scheduling problem in a hierarchical IoT network architecture and show a significant reduction in the AoI and energy consumption.

In this paper, we contribute to designing heterogeneous IoT network architectures and present a holistic approach to tackle collaborative data scheduling, spectrum allocation, and cross-technology offloading in multi-protocol IoT networks. In particular, our paper presents the following contributions:

- We present a cross-technology network architecture design for heterogeneous IoT, where a middle layer of M-MGs is deployed by various SPs operating in different wireless protocols. M-MGs are equipped with multiple interfaces and SPs incentivize them to collect data from heterogeneous IoT devices, perform computing tasks, and offload the data to nearby access points (APs) for data aggregation.
- We develop a collaborative framework between SPs and M-MGs to optimize cross-technology data scheduling, relaying, and bandwidth allocation for heterogeneous IoT devices and minimize the AoI and energy consumption. The M-MGs' strategies involve collection, execution, relaying, and selection of the offloading technology and APs. On the other hand, SPs handle spectrum sharing and allocation considering the interference among M-MGs and pricing. Moreover, pricing schemes are proposed to negotiate the collaboration between M-MGs and SPs.
- A cross-technology federated matching algorithm (CT-Fed-Match) is presented to solve the previous optimization problem. It is based on a multi-protocol, multi-agent actor-critic with two levels of FL in which SPs collaborate and match with M-MGs by sharing their bandwidth, and M-MGs collaborate with each other for relaying and offloading data. Our algorithm is capable of adapting to different data patterns and network conditions.
- Extensive simulation results show significant performance improvements of CT-Fed-Match algorithm in terms of AoI and energy cost, amount of data collected, and convergence speed compared to existing approaches.

The structure of the paper is as follows. Section II summarizes the related work. Section III describes the system model. The collaborative data scheduling and relaying scheme is presented in Section IV. Section V describes the cross-technology spectrum allocation and offloading. The multi-protocol federated matching framework for AoI and energy cost minimization is introduced in Section VI. Section VII presents the simulation results. Finally, Section VIII concludes the paper.

II. RELATED WORK

This section summarizes the related work and compares it to our contributions in Table I.

A. AoI Minimization in IoT-Enabled Edge Computing Networks

Xie et al. [7] develop scheduling algorithms for industrial Internet of Things (IIoT) with multiple sensors and servers to minimize the AoI. They adopt a guided exploration-based deep Q-network (GE-DQN) algorithm with a fixed advantage policy. Wang et al. [9] present a dynamic scheduling

TABLE I
LITERATURE REVIEW

Reference	Multi-agent RL	Federated learning	AoI Minimization	Energy efficiency	Edge computing	Interference-aware resource allocation	Relaying collaboration	Cross-technology collaboration	Pricing
[7]			✓		✓				
[8]	✓		✓	✓					
[9]			✓		✓				
[10]	✓		✓	✓					
[11]	✓	✓	✓	✓	✓				
[12]	✓	✓	✓			✓		✓	
[13]	✓		✓	✓					
[14]	✓		✓	✓	✓				
[15]	✓	✓	✓		✓				
[17]			✓						✓
[18]			✓						✓
[19]			✓						✓
[20]			✓						✓
our work	✓	✓	✓	✓	✓	✓	✓	✓	✓

scheme based on the Lyapunov drift framework to minimize the long-term average AoI under a delay constraint. When the probability of successful data delivery is unknown, they propose a scheduling policy using the dueling deep Q network (D3QN). Wang et al. [14] present joint sampling and scheduling control to minimize the weighted sum of AoI and energy consumption. They introduce a distributed QMIX algorithm [21] for a cooperative update of reinforcement learning parameters by edge devices. Zhu et al. [15] employ a combination of MARL and FL techniques to obtain policies for trajectory planning of UAVs and resource allocation. These works focus on a single protocol, whereas this paper tackles AoI minimization in heterogeneous IoT networks where multiple wireless protocols coexist.

B. Multi-Agent RL Approaches for AoI Minimization

In [8] and [10], data collection is studied in UAV networks using monotonic value function factorization and value decomposition networks, respectively. Zhang et al. [12] present a two-tier joint resource scheduling mechanism that considers the AoI and bandwidth requirements to accommodate time-sensitive services in heterogeneous vehicular networks. Oubbati et al. [13] study timely data collection and energy transfer using UAVs and divide them into two teams: energy transmitters and data collectors. They aim to improve energy transfer, minimize the energy utilization of UAVs, and maximize the throughput of IoT devices. Our approach to minimize the AoI in heterogeneous IoT relies on a MADRL algorithm to cope with the uncertainty of data demands, available spectrum bands, and mobility, and two-level FL to facilitate collaboration between SPs and M-MGs.

C. Pricing for Age Control

Zhang et al. [17] introduce a pricing scheme for fresh data, which includes time-dependent and quantity-based pricing for buyers to value the data based on fresh features.

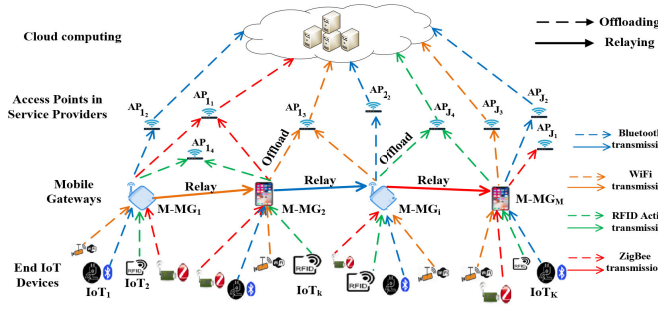


Fig. 1. Cross-technology federated IoT architecture.

Zhang et al. [18] investigate an optimal mechanism design to stimulate fresh updates and enhance the destination's payoff. Wang et al. [19] propose a decentralized mean field pricing scheme and offer age-dependent monetary returns to encourage users to sample information at different rates over time. Modina et al. [20] present a price-based mechanism using AoI to minimize the costs of traffic offloading for SPs. Our work goes a step further and presents cross-technology pricing and relaying incentives to motivate collaboration across technologies and minimize the AoI and energy cost.

III. SYSTEM MODEL

In this section, we present our heterogeneous IoT architecture design and communication, computing, and data scheduling models.

A. Cross-Technology IoT Network Architecture

We consider a heterogeneous IoT network architecture, as shown in Fig. 1, operated by a set of Z IoT SPs denoted by $\mathcal{Z} = \{1, \dots, z, \dots, Z\}$. Each SP z operates a wireless protocol z to serve traffic demands from a subset of K_z IoT devices $\mathcal{K}_z \in \mathcal{K}$, $\mathcal{K} = \{1, \dots, k, \dots, K\}$, where $K = \sum_z K_z$ is the number of IoT devices in the network. Besides, each SP z owns a subset of J_z APs $\mathcal{J}_z \in \mathcal{J}$, where $\mathcal{J} = \{1, \dots, j, \dots, J\}$ is the set of $J = \sum_z J_z$ APs in the network. Each AP has access to its SP's cloud computing center for data aggregation. In this hierarchical architecture, SPs collaborate to access edge computing resources facilitated by a set of M M-MGs denoted by $\mathcal{M} = \{1, \dots, i, \dots, M\}$ that can operate in $Z_i \leq Z$ interfaces and have computing and caching capabilities. We assume there is no coverage overlapping among M-MGs. SPs incentivize M-MGs to collect data from the IoT devices, execute their computing tasks, and eventually offload it to the corresponding AP. M-MGs can also collaborate to relay each other's data if there is no AP in their coverage area. The execution at M-MGs is needed to extract features from the data (e.g., data analysis, AI algorithm). Possible applications of our architecture include industrial IoT, smart city, and smart agriculture, among others.

Moreover, our architecture design facilitates cross-technology collaboration in which SPs z and z' operating on different technologies can share their offloading bandwidth. Since different protocols may have different frame formats, M-MGs perform protocol conversion and

do the necessary embedding to transmit the data collected on interface z' (from a collaborating SP z') and offload it to AP $j_z \in \mathcal{J}_z$ [4], [5]. Furthermore, SPs must schedule the bandwidth allocation to reduce the interference when different protocols use overlapping frequency bands (e.g., cellular, WiFi, ZigBee, and LoRa on 2.4GHz). SPs and M-MGs collaborate for data scheduling, which includes data collection from IoT devices by M-MGs, execution of the collected data at M-MGs, relaying the data among M-MGs, offloading the data from M-MGs to APs, and spectrum allocation to the M-MGs by SPs. We summarize the most important notations in Table II.

B. Communication Model

Let $\mathcal{Z}^I$ denote the set of interfaces operating in overlapping bands and $\mathcal{Z}^{-I}$ the set of interfaces whose transmissions will not cause interference to any other interface, $\mathcal{Z} = \mathcal{Z}^I \cup \mathcal{Z}^{-I}$. For simplicity, we assume M-MGs have only one radio per interface. We model our communication system in a time-slotted manner. Each SP $z \in \mathcal{Z}$ has a total bandwidth W^z and each time t allocates a fraction $w_i^z(t)$ to each M-MG i such that $\sum_i w_i^z(t) = 1$. We assume that each IoT device k of SP z ($\forall k = k_z \in \mathcal{K}_z$) has its own bandwidth W_{ki}^z for transmission to M-MG i . Therefore, M-MG i may use the fraction $w_i^z(t)$ to transmit to AP j ($\forall j = j_z \in \mathcal{J}_z$), i.e., $w_{ij}^z(t) = w_i^z(t)$ or to relay the data to M-MG i' , i.e., $w_{ii'}^z(t) = w_i^z(t)$. We consider that transmissions from different M-MGs on interface z are orthogonal. To avoid collisions, interfaces operating in overlapping bands cannot collect, relay, and/or offload data simultaneously.

At each time t , the link capacity R_{ki}^z between IoT device k and M-MG i on interface z , R_{ij}^z between M-MG i and AP j on z , and $R_{ii'}^z$ between M-MG i and M-MG i' on z are

$$R_{ki}^z(t) = W_{ki}^z \log_2(1 + P_k g_{ki}^z / \xi_i) \quad (1)$$

$$R_{ij}^z(t) = w_{ij}^z(t) W^z \log_2(1 + P_i g_{ij}^z / (\xi_j + \sum_{z' \in \mathcal{Z}^I, z' \neq z} \sum_{i'} \chi_{i'j}^{z'})) \quad (2)$$

$$R_{ii'}^z(t) = w_{ii'}^z(t) W^z \log_2(1 + P_i g_{ii'}^z / (\xi_{i'} + \sum_{z' \in \mathcal{Z}^I, z' \neq z} \sum_{i''} \chi_{i''i'}^{z'})) \quad (3)$$

where P_k and P_i denote the transmission power of device k and M-MG i , respectively. The power propagation gain between device k and M-MG i on interface z is g_{ki}^z , between M-MG i and AP j is g_{ij}^z , and between M-MGs i and i' is $g_{ii'}^z$. ξ_i , ξ_j , and $\xi_{i'}$ denote the Gaussian noise power at M-MG i , AP j , and M-MG i' , respectively. We assume that the interference with adjacent IoT devices and M-MGs can be neglected due to the close proximity between k and i . $\chi_{i'j}^{z'} = P_{i'} g_{i'j}^{z'}$ and $\chi_{i''i'}^{z'} = P_{i''} g_{i''i'}^{z'}$ denote the interference power caused by any M-MG $i' \in \mathcal{I}_j$ and M-MG $i'' \in \mathcal{I}_{i'}$ on z' , respectively, where $\mathcal{I}_j$ and $\mathcal{I}_{i'}$ are the set of nodes in the interference range of AP j and i' , respectively, and $z, z' \in \mathcal{Z}^I$.

C. Data Scheduling and Computing Model

Each IoT device k of SP $z \in \mathcal{Z}$ generates data independently following a Bernoulli distribution [15] with a data

TABLE II
NOTATION

Symbol	Meaning
k, i, j, z	Index of IoT device, M-MG, AP, and SP, respectively.
K, M, Z, J, J_z	Total number of IoT devices, M-MGs, SPs, APs, APs in SP z , respectively.
$\mathcal{K}, \mathcal{Z}, \mathcal{K}_z, \mathcal{J}_z$	Set of IoT devices, SPs, IoT devices served by protocol z , APs of protocol z , respectively.
$\mathcal{M}, \mathcal{Z}^I, \mathcal{Z}^{-I}, \mathcal{Z}^c$	Set of M-MGs, interfaces operating in overlapping bands, interfaces operating in non-overlapping bands, SPs that SP z collaborate with, respectively.
$W^z, w_i^z(t)$	Total bandwidth of SP z , a fraction of bandwidth allocated to each M-MG i at time t , respectively.
$R_{ki}^z, R_{ij}^z, R_{ii'}^z, \xi_i, \xi_j, \xi_{i'}$	Link capacity between IoT device k and M-MG i , M-MG i and AP j , and M-MG i and M-MG i' on interface z , and Gaussian noise power at M-MG i , AP j , and M-MG i' , respectively.
$P_k, P_i, g_{ki}^z, g_{ij}^z, g_{ii'}^z$	Transmission power of device k , transmission power of M-MG i , and channel gain between device k and M-MG i on interface z , between M-MG i and AP j and between M-MGs i and i' , respectively.
$I_{col,i}^z(t), I_{exe,i}^z(t), I_{rel,i}^z(t), I_{off,i}^z(t), I_{off,i'}^z(t)$	Binary indicators for data collection, execution, relay, offloading, and cross-technology offloading of M-MG i on protocol z to z' , respectively.
$c_{ik}^z(t), e_{ik}^{z,q}(t), o_{ij,k}^z(t)$	Binary decision for M-MG i data collection from device k per interface z at time t , for M-MG i execution of a packet q of device k at a time t , for M-MG i data offloading to an AP j of SP z at time t , respectively.
$inc_i^z, inc_{i'}^z$	Incentive from SP z to M-MG i , incentive from M-MG i' to M-MG i , respectively.
$cc^{zz'}, W^{zz}, W^{zz'}$	Fraction of bandwidth that SP z shares with SP z' , the bandwidth SP z allocates to serve its own traffic, the bandwidth SP z shares with SP z' , respectively.
$rc_{ii'}^z, W_{ii'}^z$	Fraction of bandwidth that M-MG i shares with M-MG i' , bandwidth that M-MG i shares with M-MG i' , respectively.
$E_{exe,i}, E_{off,i}, E_{rel,i}$	Energy for execution, offloading, and relaying by M-MG i , respectively.
x_i^z, y_i^z	Binary indicator that indicates whether M-MG i collaborates with SP z , binary indicator that indicates whether SP z is associated with M-MG i , respectively.
$D_{col,ik}^z(t), D_{rel,ii'}^z(t), D_{off,ij}^z(t)$	Overall amount of data collected from device k in interface z at time t , the amount of data relayed from M-MG i to i' , the amount of data offloaded by M-MG i to AP j on interface z .
$B_k^z, B_{col,i}^z, B_{exe,i}^z$	Buffer of IoT device k , collection buffer size of M-MG i per protocol z , execution buffer size of M-MG i per protocol z .
$p^z, p^{z'z}$	Bandwidth price of protocol z , the price that SP z' pays SP z for using its bandwidth.
$U_{M-MG,i}, U_{SP,z}$	Utility of M-MG i , Utility of the SP z .

packet size d_k^z , arrival rate λ_k^z , and elapsed time ψ_k^z . The elapsed time is defined as the difference between the current time and the time when the packet was generated. The generated packets are stored in each IoT device locally until M-MGs collect them. We define the buffer of IoT device k as $B_k^z = \{d_k^{z,q}, \psi_k^{z,q}, k\}$, where q is the index of the packet. We assume M-MGs move following a random mobility model [22] in which the speed is constant, and the direction of movement is chosen randomly in every slot. Each M-MG i can collect data $c_{ik}^z(t) \in \{0, 1\}$ from an IoT device k in its transmission range per interface z at a time

$$I_{col,i}^z(t) = \sum_k c_{ik}^z(t) \leq 1 \quad (4)$$

where $I_{col,i}^z(t)$ denotes the binary indicator for data collection. Each M-MG i has a collection data buffer of size $B_{col,i}^z$ per interface z defined as $B_{col,i}^z = \{d_{col,k}^{z,q}, \psi_{col,k}^{z,q}, k\}$, where $d_{col,k}^{z,q}$ is the size of a data packet in the collection buffer, and $\psi_{col,k}^{z,q}$ is the elapsed time which increases by one every slot since the packet was generated. Therefore, the overall amount of data collected from device k in interface z at time t , $D_{col,ik}^z(t) = \min\{B_{col,i}^z - \sum_{k_p} \sum_{q=1}^{B_{col,i}^z} d_{col,k_p}^{z,q}, R_{ik}^z \Delta t\}$ depends on the available space in the buffer with $k_p \in \mathcal{K}_z$ and the capacity R_{ik}^z of the link between IoT device k and M-MG i , respectively. Δt is the duration of the time slot.

Next, the data packets cached in the collection data buffers will be scheduled for local processing. Each M-MG i makes a decision $e_{ik}^{z,q}(t) \in \{0, 1\}$ to execute a packet q from the collection buffer per interface z at time t ,

$$I_{exe,i}^z(t) = \sum_{q=1}^{B_{col,i}^z} e_{ik}^{z,q}(t) \leq 1 \quad (5)$$

where $I_{exe,i}^z(t)$ is the indicator for execution. The computation time for executing a packet of size $d_{col,k}^z$ in M-MG i with CPU frequency f_i is $\tau_{ik}^z(t) = d_{col,k}^z / f_i$.

After local execution, the data packet is stored in the corresponding execution buffer $B_{exe,i}^z$ until M-MG i offloads it to a nearby AP j or relays it to an adjacent M-MG i' . In the latter case, M-MG i relays data $r_{ii',k}^z \in \{0, 1\}$ to an M-MG i' per interface z at a time

$$I_{rel,i}^z(t) = \sum_{i'} r_{ii',k}^z(t) \leq 1 \quad (6)$$

where $I_{rel,i}^z(t)$ is the relay indicator for M-MG i . The amount of data relayed $D_{rel,ii'}^z(t)$ from M-MG i to i' can be obtained as above based on the available space in the execution buffer of M-MG i' in interface z , and link capacity $R_{ii'}^z$.

Finally, M-MGs offload the data to a nearby AP. Let $o_{ij,k}^z(t) \in \{0, 1\}$ be the binary decision for M-MG i to offload data to an AP j of SP z at a time,

$$I_{off,i}^z(t) = \sum_j o_{ij,k}^z(t) \leq 1 \quad (7)$$

where $I_{off,i}^z(t)$ is the offloading indicator. The amount of data offloaded by M-MG i to AP j on interface z is $D_{off,ij}^z(t) = \min\{\sum_{k_p} \sum_{q=1}^{B_{exe,i}^z} d_{exe,k_p}^{z,q}, R_{ij}^z \Delta t\}$, which depends on the amount of data in the execution buffer collected for all $k_p \in \mathcal{K}_z$ and the capacity R_{ij}^z , respectively.

At time t , each M-MG i can collect, execute, relay, or offload data per interface z ,

$$I_{col,i}^z(t) + I_{exe,i}^z(t) + I_{rel,i}^z(t) + I_{off,i}^z(t) \leq 1 \quad (8)$$

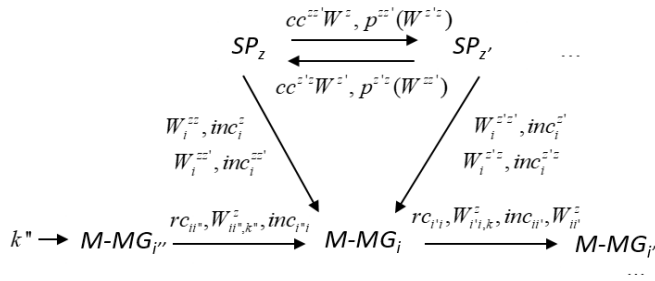


Fig. 2. Collaboration between SPs and M-MGs.

D. Collaboration Incentives

The different options for collaboration between SPs and M-MGs and their revenue share are illustrated in Fig. 2. Each SP z incentivizes M-MG i by a payment inc_i^z to collect data from its IoT devices in $\mathcal{K}_z$, perform computing tasks, and offload the data to any AP j in $\mathcal{J}^z$. In addition, SP z may collaborate with a set $\mathcal{Z}^c$ of SPs, and share a fraction of its bandwidth $cc^{zz'}$ with each SP $z' \in \mathcal{Z}^c$ to offload their data using interface z . We refer to this collaboration as a cross-technology collaboration between SP z and SP $z' \in \mathcal{Z}^c$. The bandwidth SP z shares with z' is $W^{zz'} = cc^{zz'} W^z$, and thus, the bandwidth SP z allocates to serve its own traffic is $W^{zz} = W^z - W^{zz'}$. The SP z splits its bandwidth and allocates a fraction $W_{i,z}^{zz'} = w_{ij}^z W^{zz'}$ to each M-MG i to offload traffic for SP z' , where w_{ij}^z is the fraction of the bandwidth allocated by SP z to M-MG i for traffic offloading. The cross-technology offloading indicator is

$$I_{off,i}^{zz'}(t) = \sum_j o_{ij,k'}^{zz'}(t) \leq 1 \quad (9)$$

where $o_{ij,k'}^{zz'} \in \{0,1\}$ is the binary decision for M-MG i to offload data collected on interface $z' \in \mathcal{Z}^c$ to an AP j ($\forall j = j_z \in \mathcal{J}_z$) using interface z . The data offloaded $D_{off,ij}^{zz'}(t) = \min\{\sum_{k_p} \sum_{q=1}^{B_{exe,i}^{z'}} d_{exe,k_p}^{z',q}, B_{exe,i}^z - \sum_{k_p} \sum_{q=1}^{B_{exe,i}^z} d_{exe,k_p}^{z,q}, cc^{zz'} R_{ij}^z \Delta t\}$ depends on the amount of data in the execution buffer of interface z' waiting to be offloaded, the available space in the execution buffer of interface z to transfer the data from interface z' , and the offloading capacity that SP z shares with z' , respectively. R_{ij}^z is given in (2). The SP z' pays a price $p^{z''z}$ to SP z for using a unit of its bandwidth. In return, SP z incentivizes M-MG i to offload data of SP z' using interface z by $inc_{i,z}^{z'}$. This is aimed to compensate for the cost of protocol conversion.

Similarly, M-MG i may collaborate with a nearby M-MG i' to relay its data. As illustrated in Fig. 2, M-MG i'' relays its collected data to M-MG i' through collaborating with M-MG i for offloading to the corresponding AP. We assume M-MG i shares a fraction $rc_{ii''}^z$ of its bandwidth with M-MG i'' when relaying its data to M-MG i' . Thus, the bandwidth shared by M-MG i is $W_{ii''}^z = rc_{ii''}^z w_{ii''}^z W^z$, where $w_{ii''}^z$ is the fraction of bandwidth allocated by SP z that M-MG i uses for relaying to M-MG i' . In return, M-MG i'' reimburses M-MG i by $inc_{i,i''}^z = rc_{ii''}^z inc_{i,i''}^z$, where $inc_{i,i''}^z$ is the incentive provided by SP z to M-MG i'' . In the sequel, the incentive and the bandwidth price will be negotiated to determine the conditions for the collaboration.

IV. COLLABORATIVE DATA SCHEDULING, RELAYING, AND SPECTRUM ALLOCATION SCHEME

This section presents a collaborative framework in which SPs and M-MGs collaborate for data scheduling, which includes data collection, execution, relaying, offloading, and spectrum allocation. Besides, a pricing scheme is introduced to incentivize M-MGs for their collaboration.

A. Joint SPs and M-MGs Data Scheduling, Relaying Collaboration, and Spectrum Allocation

M-MGs collaborate relaying data for each other to increase the amount of data offloaded to nearby APs and minimize the energy cost. The energy cost of M-MG i to serve IoT device k at time t includes the energy $E_{exe,i}$ to execute the collected data $D_{col,ik}^z$, and the energy for offloading $E_{off,i}$ or relaying $E_{rel,i}$,

$$cost_i^z(t) = \sum_k I_{exe,i}^z(t) D_{col,ik}^z(t) E_{exe,i} + I_{off,i}^z(t) E_{off,i} + I_{rel,i}^z(t) E_{rel,i} \quad (10)$$

where $E_{off,i} = P_i D_{off,ij}^z / R_{ij}^z$ depends on the transmission power P_i of M-MG i , the offloaded data $D_{off,ij}^z$ to AP j and the link capacity R_{ij}^z as in (2). $E_{rel,i} = P_i D_{rel,ii'}^z / R_{ii'}^z$ is a function of the data $D_{rel,ii'}^z$ relayed to M-MG i' and the capacity $R_{ii'}^z$, as in (3). $I_{exe,i}^z$, $I_{off,i}^z$, and $I_{rel,i}^z$ are the indicators for execution, offloading, and relaying, respectively.

On the other hand, SPs allocate the bandwidth to their selected M-MGs to minimize the AoI. The AoI is a performance metric that measures the freshness of the data at the receiver side. We define the AoI in our heterogeneous IoT network as the difference between the current time t and the generation time $t_{G,k}^z$ of the latest data packet from device k operating in protocol z received at the corresponding AP j

$$A_{kj}^z(t) = t - t_{G,k}^z, \quad \forall k = k_z \in \mathcal{K}_z, \quad \forall j = j_z \in \mathcal{J}_z \quad (11)$$

The AoI increases linearly with time until the packet is received at the destination. Therefore, SPs incentivize M-MGs to collect their data and offload it to nearby APs to reduce the AoI. In this section, we assume that the data will be offloaded to APs from the same SPs, and we will explore cross-technology offloading in the next section. The interaction between M-MGs and SPs for relaying collaboration and data scheduling is illustrated in Fig. 3. First, if M-MG i has IoT devices nearby operating in protocol z , it will collaborate with SP z indicated by $x_i^z = 1$, and request bandwidth $w_{req,i}^z W^z$ and an incentive $inc_{req,i}^z$ to relay or offload the data to an AP j from SP z . Depending on the available bandwidth, the network condition, and the other M-MGs requesting bandwidth, SP z decides whether to select M-MG i to serve its data. Therefore, we introduce $\mathbf{y}^z = [y_i^z]_{(1 \times M)}$ with $y_i^z \in \{0,1\}$ indicates that SP z is associated with M-MG i when $y_i^z = 1$ and 0 otherwise. If selected i.e., $y_i^z = 1$, SP z incentivizes the M-MG i by $inc_i^z \geq inc_{req,i}^z$ and allocates bandwidth $w_i^z \leq w_{req,i}^z$ to minimize the AoI. If M-MG i has no AP within its coverage, it will incentivize an adjacent M-MG i' by $inc_{ii'}^z$ to share a portion of its bandwidth and relay its data to a nearby AP j from SP z .

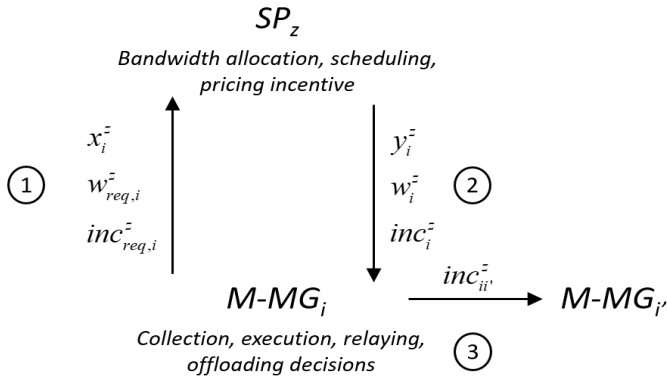


Fig. 3. Pricing framework for M-MGs relaying collaboration and SPs data scheduling.

The M-MG optimization problem includes collection, execution, relaying, and offloading decisions. The utility U_{M-MG_i} of M-MG i at time t is a function of the incentive received from the SP z for offloading data $D_{off,ij}^z$ to any AP j from SP z or relaying the data $D_{rel,ii'}^z$ to M-MG i' , the incentive offered to M-MG i' for relaying its data, the incentive for relaying data $D_{rel,ii''}^z$ from M-MG i'' , the price per unit of the bandwidth $p^z f(w_{req,i}^z W^z)$ requested to SP z , and the energy cost $cost_i^z$,

$$\begin{aligned} U_{M-MG_i}(t) &= \sum_{i'} \sum_{i''} \sum_z \sum_k I_{off,i}^z(t) D_{off,ij}^z(t) inc_i^z \\ &\quad + I_{rel,i}^z(t) D_{rel,ii'}^z(t) (inc_i^z - inc_{ii'}^z) - cost_i^z(t) \\ &\quad + I_{rel,i''}^z(t) D_{rel,ii''}^z(t) inc_{ii''}^z - p^z f(w_{req,i}^z(t) W^z) \end{aligned} \quad (12)$$

where $I_{off,i}^z$ and $I_{rel,i}^z$ are the indicators for offloading and relaying, respectively. The bandwidth pricing function is defined as $f(w_{req,i}^z W^z) = \log(1 + I_{rel,i}^z R_{ii'}^z(w_{req,ii'}^z) + I_{off,i}^z R_{ij}^z(w_{req,ij}^z))$ with R_{ij}^z and $R_{ii'}^z$ given by (2) and (3), respectively. Therefore, the bandwidth must be allocated efficiently to minimize its cost.

The M-MGs collaborative optimization problem includes selecting IoT devices for data collection $\mathbf{C}_i^z = [c_{ik}^z]$, data execution $\mathbf{E}_i^z = [e_{ik}^{z,q}]$, selection of M-MG i' for relaying $\mathbf{R}_i^z = [r_{ii',k}^z]$, and selection of the offloading AP j , $\mathbf{O}_i^z = [o_{ij,k}^z]$,

$$\max_{\mathbf{C}_i^z, \mathbf{E}_i^z, \mathbf{R}_i^z, \mathbf{O}_i^z} \frac{1}{T} \sum_t \sum_i U_{M-MG_i}(t) \quad (13)$$

subject to (1) – (8), (10)

$$\sum_i \sum_k c_{ik}^z(t) \leq M \quad (13.a)$$

$$\sum_z \sum_k e_{ik}^{z,q}(t) \leq Z \quad (13.b)$$

$$\sum_z \sum_{i'} \sum_k r_{ii',k}^z(t) \leq Z \quad (13.c)$$

$$\sum_i \sum_j \sum_k o_{ij,k}^z(t) \leq J_z \quad (13.d)$$

$$c_{ik}^z, e_{ik}^{z,q}, r_{ii',k}^z, o_{ij,k}^z \in \{0, 1\} \quad (13.e)$$

where constraint (13.a) states that all M-MGs can collect data from at most M IoT devices per interface z at time t , constraint (13.b) indicates that the M-MG i can execute data for Z protocols at time t , constraint (13.c) states that the M-MG i can relay data on Z interfaces to the other M-MGs at time t , constraint (13.d) states that overall M-MGs can offload data to a maximum of J_z APs per interface z at time t , and constraint (13.e) summarizes the binary decision variables. We assume that the M-MGs do not need to know the location of IoT devices since they move to collect the data independently on their location. However, we assume that M-MGs are aware of the locations of other M-MGs and APs within their transmission range to make relaying and offloading decisions. We assume the localization information is error-free.

The bandwidth that M-MG i requires from SP z is $W_{req,i}^z(t) = x_i^z(t) w_{req,i}^z(t) W^z$, where $x_i^z(t)$ is the decision of the M-MG i to offload or relay data for SP z at time t ,

$$x_i^z(t) = \begin{cases} 1, & \text{if } I_{rel,i}^z(t) + I_{off,i}^z(t) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

Based on the requests received, the SP optimizes the selection of M-MGs and the bandwidth allocation. The utility U_{SP_z} of the SP z includes the revenue for selling the bandwidth $p^z f(w_i^z W^z)$, the incentive offered to each M-MG i for offloading $D_{off,ij}^z$ or relaying $D_{rel,ii'}^z$, and the AoI A_k^z ,

$$\begin{aligned} U_{SP_z}(t) &= \sum_i p^z f(w_i^z(t) W^z) - I_{off,i}^z(t) D_{off,ij}^z(t) inc_i^z \\ &\quad - I_{rel,i}^z(t) D_{rel,ii'}^z(t) inc_i^z - \varrho A^z(t) \end{aligned} \quad (15)$$

where $A^z(t) = \sum_{k=k_z \in \mathcal{K}_z} \sum_{j=j_z \in \mathcal{J}_z} A_{kj}^z(t)$, and ϱ is a proportionality coefficient. Then, the SP's optimization problem for bandwidth allocation and data scheduling is

$$\max_{\mathbf{y}^z, \mathbf{w}^z} \frac{1}{T} \sum_t U_{SP_z}(t) \quad (16)$$

$$\text{subject to } y_i^z(t) \leq x_i^z(t) \quad (16.a)$$

$$\sum_{i \in \mathcal{M}} y_i^z(t) \leq M \quad (16.b)$$

$$y_i^z(t) w_i^z(t) \geq w_{req,i}^z(t) \quad (16.c)$$

$$\sum_i y_i^z(t) w_i^z(t) W^z \leq W^z \quad (16.d)$$

where $\mathbf{y}^z = [y_i^z]$ with $y_i^z \in \{0, 1\}$ is the decision of SP z to select M-MG i , and $\mathbf{w}^z = [w_i^z]$ is the fraction of bandwidth SP z allocates to M-MG i . The constraint (16.a) indicates the decision of SP z to select the request of M-MG i for serving its data, constraint (16.b) states that each SP z can select a maximum of M M-MGs to serve its data each time t , constraint (16.c) ensures that the fraction of the bandwidth allocated to M-MG i does not exceed the required one, and constraint (16.d) guarantees that the overall bandwidth allocated to the selected M-MGs by SP z does not exceed its available bandwidth.

The previous optimization problems are NP-hard combinatorial problems, and the solution must be obtained iteratively

between M-MGs and SPs. Therefore, in Section VI, we reformulate the problem as an MDP and solve it using multi-agent reinforcement learning.

B. Two-Level Pricing Negotiation for M-MGs Selection, and Relaying Collaboration

In the previous optimization problems, we have assumed that the incentive inc_i^z and the relaying collaboration $rc_{i'i}^z$ between M-MG i' and i are given. Next, we describe the negotiation process between SPs and their selected M-MGs to determine their collaboration. We assume that the incentive $inc_{req,i}^z$ requested by each M-MG i is the minimum it is willing to accept. Consequently, each SP and its set of M-MGs negotiate the incentive as follows: 1) each M-MG i and SP z solve their respective optimization problems (13) and (16) for the requested incentive $inc_{req,i}^z$, 2) SP z offers a higher incentive $inc_i^z = inc_{req,i}^z + \Delta inc_{req,i}^z$, 3) M-MG i solves (13) with $inc_{req,i}^z$ and 4) if $U_{SP_z}(t, inc_{req,i}^z) < U_{SP_z}(t, inc_{req+i}^z)$, and $U_{M-MG_i}(t, inc_{req,i}^z) < U_{M-MG_i}(t, inc_{req+i}^z)$ then M-MG i and SP z agree on the new incentive, $inc_i^{z*} = inc_{req+i}^z$. Steps 2-4 are repeated until one of the inequalities in step 4 does not hold. The final incentive SP z offers to M-MG i is inc_i^{z*} .

Then, M-MGs negotiate the relaying collaboration. As illustrated in Fig. 2, we assume that M-MG i'' has an adjacent M-MG i that can potentially relay its data to M-MG i' . To decide the relaying collaboration $rc_{i'i}^z$ between M-MGs i' and i , we assume that M-MG i relays data to M-MG i' if $U_{M-MG_i}(t, I_{rel,i}^z) \geq U_{M-MG_i}(t, I_{off,i}^z)$,

$$0 \leq rc_{i'i}^z \leq 1 - \left(\left(D_{off,ij}^z(t) inc_i^z - p^z (f(w_{req,ij}^z(t) W^z) - f(w_{req,ii'}^z(t) W^z)) + E_{rel,i} - E_{off,i} \right) / D_{rel,ii'}^z(t) inc_i^z \right) \quad (17)$$

where $E_{rel,i}$, and $E_{off,i}$ are the energy cost for relaying, and offloading for M-MG i , respectively, and $w_{req,ii'}$ and $w_{req,ij}$ are the requested bandwidth for relaying to M-MG i' and offloading to AP j , respectively. Moreover, M-MG i collaborates with M-MG i'' and relays its data if its utility under relaying collaboration $rc_{i'i}^z$ is $U_{M-MG_i}(t, rc_{i'i}^z) \geq U_{M-MG_i}(t, 0)$,

$$\begin{aligned} & \left(I_{off,i}^z(t) D_{off,ij}^z(t) inc_i^z + I_{rel,i}^z(t) D_{rel,ii'}^z(t) (inc_i^z - inc_{i'i}^z) - p^z f(w_{req,i}^z(t) W^z) - cost_i^z(t) \right) / D_{rel,ii'}^z(t) inc_i^z \\ & \leq rc_{i'i}^z \leq 1 \end{aligned} \quad (18)$$

where $w_{req,i} = I_{rel,i}^z w_{req,ii'} + I_{off,i}^z w_{req,ij}$, $inc_{i'i}^z = inc_{i'i}^z rc_{i'i}^z$, and $inc_{i'i}^z = inc_i^z rc_{i'i}^z$.

V. COLLABORATIVE CROSS-TECHNOLOGY DATA SCHEDULING AND SPECTRUM ALLOCATION

In this section, we present a scheme for cross-technology collaboration in which SPs share the offloading bandwidth to

increase the opportunities to offload the data and minimize the AoI. In addition, we optimize the collaboration between SPs and M-MGs for cross-technology data scheduling.

A. Joint SPs and M-MGs Cross-Technology Collaboration, and Data Scheduling

We consider collaboration between SPs and M-MGs for cross-technology offloading, i.e., offload data using a different wireless protocol/interface from data collection, as shown in Fig. 4. In the first step, based on the potential APs available of SP z , each M-MG i decides which data to offload from each of the execution buffers per interface z . The data may be from SP z (i.e., collected on interface z) or from any collaborating SP $z' \in \mathcal{Z}^c$ (i.e., collected on interface z'). If $x_i^z = 1$ M-MG i selects interface z to offload the data collected on z and requests bandwidth $w_{req,i}^z W^z$ to SP z . Otherwise, $x_i^z = 0$. Similarly, if $x_i^{zz'} = 1$, M-MG i selects interface z to offload the data collected on z' and requests bandwidth $w_{req,i}^{zz'} W^{zz'}$ to SP z . Otherwise, $x_i^{zz'} = 0$. Each interface z can be selected to offload data collected by a maximum of Z interfaces, $x_i^z + \sum_{z'} x_i^{zz'} \leq Z$. In addition, M-MG i requests SP z a monetary compensation $inc_{req,i}^z$ or $inc_{req,i}^{zz'}$ for serving its data or the data of SP z' , respectively. The latter requires an extra cost. Secondly, each SP z sets the price p^z for collaborating with each SP z' based on the potential amount of bandwidth required to offload its data. This is elaborated in detail in Section V-B. Finally, each SP z selects the M-MGs depending on the available bandwidth, the network condition, and requests received from other M-MGs. If SP z selects M-MG i to offload its data on interface z , then $y_i^z = 1$, and allocates bandwidth $w_i^z W^z$ and offers an incentive inc_i^z . Otherwise, $y_i^z = 0$. Similarly, SP z may also select M-MG i to collaborate with SP z' and serve its traffic using interface z , $y_i^{zz'} = 1$. In this case, it allocates bandwidth $w_i^{zz'} W^{zz'}$ and offers an incentive $inc_i^{zz'}$. Otherwise, $y_i^{zz'} = 0$.

Next, we formulate the cross-technology collaboration optimization problem for M-MGs and SPs. The M-MGs decide the data collection, execution, and selection of the offloading interface, while SPs perform M-MG selection and bandwidth allocation. The utility of M-MG i includes the incentive provided by SP z to offload the data $D_{off,ij}^z$ and $D_{off,ij}^{zz'}$ collected in interface z and $z' \in \mathcal{Z}^c$, respectively, the price for the requested offloading bandwidth $w_{req,i}^z$ and $w_{req,i}^{zz'}$, and the energy cost $cost_i^z$ and $cost_i^{zz'}$,

$$\begin{aligned} U_{M-MG_i}(t) = & \sum_z \sum_{z'} I_{off,i}^z(t) (D_{off,ij}^z(t) inc_i^z - p^z f(w_{req,i}^z(t) W^z) - cost_i^z(t)) \\ & + I_{off,i}^{zz'}(t) (D_{off,ij}^{zz'}(t) inc_i^{zz'} - p^{zz'} f(w_{req,i}^{zz'}(t) W^{zz'}) - cost_i^{zz'}(t)) \end{aligned} \quad (19)$$

where $cost_i^{zz'}(t) = I_{off,i}^{zz'}(t) E_{off,i}^{zz'}$, and the cross-technology offloading energy is $E_{off,i}^{zz'} = P_i D_{off,ij}^{zz'} / R_{ij}^z$ which depends on the transmission power P_i of M-MG i , the offloaded data $D_{off,ij}^{zz'}$ to AP j and the link capacity R_{ij}^z as in (2) for the requested bandwidth.

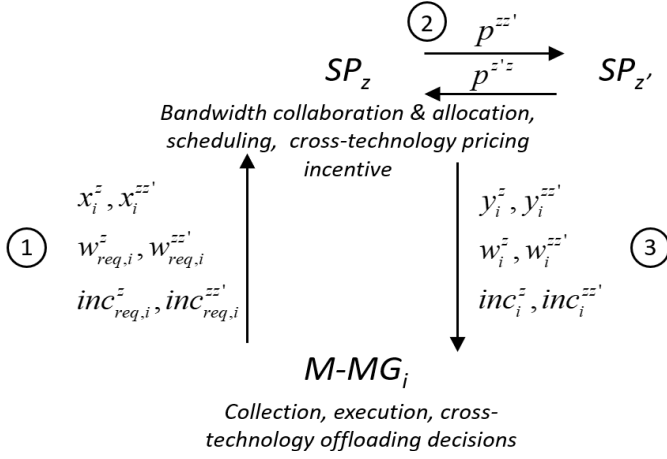


Fig. 4. Pricing framework for cross-technology collaboration.

The cross-technology optimization problem for M-MGs can thus be written as

$$\max_{C_i^z, E_i^z, O_i^z, O_i^{zz'}} \frac{1}{T} \sum_t \sum_i U_{M-MG_i}(t) \quad (20)$$

subject to (1), (2), (4), (5), (7), (9), (10), (13.a), (13.b)

$$I_{col,i}^z(t) + I_{exe,i}^z(t) + I_{off,i}^z(t) \leq 1 \quad (20.a)$$

$$I_{off,i}^z(t) + \sum_{z'} I_{off,i}^{zz'}(t) \leq Z \quad (20.b)$$

$$\sum_i \sum_j \left(\sum_k o_{ij,k}^z(t) + \sum_{z'} \sum_{k'} o_{ij,k'}^{zz'}(t) \right) \leq J_z \quad (20.c)$$

$$c_{ik}^z, e_{ik}^{z,q}, o_{ij,k}^z, o_{ij,k'}^{zz'} \in \{0, 1\} \quad (20.d)$$

where the constraint (20.a) ensures that the M-MG i either collects, executes, or offloads the data on interface z at time t . $I_{off,i}^z$ is a binary offloading indicator which equals to 1 when M-MG i offloads the data collected using any available interface z or z' . Otherwise, it equals 0. The constraint (20.b) guarantees that the data selected for offloading in interface z has been collected from a maximum of Z SPs. The constraint (20.c) states that all M-MGs can offload data to at most J_z APs per interface z at time t . Finally, the constraint (20.d) summarizes the binary decision variables.

The association of M-MG i to SP z to offload its data or the data from $z' \in \mathcal{Z}^c$ on interface z is,

$$x_i^z(t) = \begin{cases} 1, & \text{if } I_{off,i}^z(t) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

and

$$x_i^{zz'}(t) = \begin{cases} 1, & \text{if } I_{off,i}^{zz'}(t) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

respectively. In each case, the offloading bandwidth M-MG i requests to SP z is $W_{req,i}^z(t) = x_i^z(t)w_{req,i}^z(t)W^z$ and $W_{req,i}^{zz'}(t) = x_i^{zz'}(t)w_{req,i}^{zz'}(t)W^{zz'}$, respectively.

Based on the received requests, SPs select the M-MGs and allocate the bandwidth. The utility U_{SP_z} of SP z includes the price $p^z f(w_i^z W^z)$ for the bandwidth allocated for offloading on interface z , the revenue $p^{zz'} f(w_i^{zz'} W^{zz'})$ for collaborating

with SP z' , the price $p^{zz''} f(w_i^{zz''} W^{zz''})$ paid to other SP z'' for sharing their bandwidth with SP z , the monetary incentives for M-MGs to offload data $D_{off,i,j}^z$ and $D_{off,i,j}^{zz'}$ on interface z , and the AoI of SP z 's data A^z and $A^{zz'}$ when offloaded to an AP of SP z and z'' , respectively,

$$\begin{aligned} U_{SP_z}(t) = & \sum_{z'} \sum_{z''} \sum_i p^z f(w_i^z(t) W^z) + p^{z'z} f(w_i^{z'z}(t) W^{z'z}) \\ & - p^{zz''} f(w_i^{zz''}(t) W^{zz''}) - I_{off,i}^z(t) D_{off,i,j}^z(t) inc_i^z \\ & - I_{off,i}^{zz'}(t) D_{off,i,j}^{zz'}(t) inc_i^{zz'} - \varrho(A^z(t) + A^{zz'}(t)) \end{aligned} \quad (23)$$

where $A^{zz''}(t) = \sum_{k=k_z \in \mathcal{K}_z} \sum_{j=j_{z''} \in \mathcal{J}_{z''}} A_{kj}^{zz''}(t)$ and ϱ is a proportionality coefficient. The SP optimizes the cross-technology collaboration, selection of M-MGs, and bandwidth allocation as

$$\max_{y^z, y^{zz'}, w^z, w^{zz'}} \frac{1}{T} \sum_t \sum_z U_{SP_z}(t) \quad (24)$$

subject to (16.a), (16.c)

$$y_i^{zz'}(t) \leq x_i^{zz'}(t) \quad (24.a)$$

$$\sum_i \left(y_i^z(t) + \sum_{z'} y_i^{zz'}(t) \right) \leq M \quad (24.b)$$

$$\begin{aligned} \sum_i \left(y_i^z(t) w_i^z(t) W^z \right. \\ \left. + \sum_{z'} y_i^{zz'}(t) w_i^{zz'}(t) W^{zz'} \right) \leq W^z \end{aligned} \quad (24.c)$$

$$y_i^{zz'}(t) w_i^{zz'}(t) \leq w_{req,i}^{zz'}(t) \quad (24.d)$$

where the constraint (24.a) guarantees that SP z selects an M-MG that has requested a connection to offload data from z' on interface z , (24.b) states that the SP z can recruit a maximum of M M-MGs each time t , (24.c) guarantees that the overall bandwidth allocated to all M-MGs selected to serve the data from SP z or $z' \in \mathcal{Z}^c$ on interface z does not exceed the available bandwidth of SP z , and (24.d) ensures that the fraction of the bandwidth allocated to each M-MG i from SP z to serve the data from SP $z' \in \mathcal{Z}^c$ does not exceed the required bandwidth. Since solving the previous problems is NP-hard, in Section VI, we reformulate them as an MDP. Furthermore, we integrate both collaborative schemes in a global collaborative framework using multi-agent reinforcement learning.

B. Two-Level Pricing Negotiation for M-MGs Cross-Technology Offloading and SPs Collaboration

We assume that each SP z negotiates the incentive with the selected M-MGs, as described in Section IV-B. Let us recall that SP z offers to each M-MG i an incentive $inc_i^z \geq inc_{req,i}^z$ and $inc_i^{zz'} \geq inc_{req,i}^{zz'}$ to serve its traffic and the traffic of a collaborating SP z' , respectively. In both cases, the incentive has to compensate the M-MG i for the offloading cost, and the bandwidth price p^z , i.e., $U_{M-MG_i}(t, I_{off,i}^z) \geq 0$ and $U_{M-MG_i}(t, I_{off,i}^{zz'}) \geq 0$. In addition, the price $p^{z'z}$ that SP

z' pays SP z for using its bandwidth must compensate for the incentive SP z offers to each M-MG i to serve the traffic of z' on interface z . Therefore, the price $p^{zz'}$ must satisfy that $U_{SP_z}(t, cc^{zz'}) \geq U_{SP_z}(t, 0)$,

$$p^{zz'} \geq \frac{D_{off,ij}^{zz'}(t) inc_i^{zz'} + \varrho \left(A_{total}^z(t, cc^{zz'}) - A_{total}^z(t, 0) \right)}{f(w_i^{zz'}(t) W^{zz'})} \quad (25)$$

where $A_{total}^z = A^z + \sum_{z''} A^{z''z}$. Moreover, M-MG i offloads the data from SP z' for the given incentive if the bandwidth allocated or, equivalently, the fraction of the bandwidth shared $cc^{zz'}$ is such that $U_{M-MG_i}(t, I_{off,i}^{zz'}) \geq U_{M-MG_i}(t, I_{off,i}^z)$,

$$D_{off,ij}^{zz'}(t) cc^{zz'} inc_i^{zz'} - p^z f(w_{req,i}^{zz'}(t) cc^{zz'} W^z) \geq D_{off,ij}^z(t) inc_i^z - p^z f(w_{req,i}^z(t) W^z) - E_{off,i}(t, 0) + E_{off,i}(t, cc^{zz'}) \quad (26)$$

where $cc^{zz'}$ can be solved by numerical methods [23]. If multiple bands are available, the SP z will select the frequency band b that results in the highest capacity to increase its utility. Both SPs and M-MGs will benefit from more available spectrum if the offset in the AoI compensates for the increase in the offloading cost, i.e., $A^z(t, R_{ij}^z(t)) - A^z(t, R_{ij,b}^z(t)) \geq cost_i^z(t, R_{ij,b}^z(t)) - cost_i^z(R_{ij}^z(t))$.

VI. MULTI-PROTOCOL FEDERATED MATCHING FRAMEWORK FOR AOI AND ENERGY COST MINIMIZATION

This section presents a global collaborative framework that integrates cross-technology and relaying collaboration. In addition, we describe the multi-protocol federated matching algorithm to solve the minimization of the AoI and energy cost.

A. Global Collaborative Framework for AoI and Energy Cost Minimization

In the previous sections, we discussed the conditions for the SPs to collaborate and share their offloading bandwidths and for the M-MGs to collect, execute, relay, and offload data for different SPs. Our next goal is to define a global collaborative framework in which SPs and M-MGs jointly maximize the social welfare (SW). By summing the utilities of all SPs and M-MGs in the network, we obtain

$$SW(t) = \sum_z U_{total,SP_z} + \sum_i U_{total,M-MG_i} \\ = - \left(\varrho \zeta_1 \sum_z A_{total}^z(t) + \zeta_2 \sum_i \sum_z cost_{total,i}^z(t) \right) \quad (27)$$

which is a weighting optimization of the AoI and energy consumption with weights ζ_1 and ζ_2 , respectively, with $\zeta_1 + \zeta_2 = 1$, and ϱ is a scaling factor. U_{total,SP_z} and $U_{total,M-MG_i}$ are the utilities of SP z and M-MG i , respectively, that include relaying and cross-technology collaboration. They can be obtained from (15) and (23), and (12) and (19), respectively, $A_{total}^z = A^z + \sum_{z''} A^{z''z}$, and $cost_{total,i}^z = cost_i^z + \sum_{z'} cost_i^{zz'}$.

We formulate the minimization of the AoI and energy cost by jointly optimizing bandwidth allocation w^z and $w^{zz'}$ to serve the traffic of SP z and any collaborating SP $z' \in \mathcal{Z}^c$, respectively, data scheduling for data collection C_i^z , execution E_i^z , relaying R_i^z , and offloading O_i^z , and cross-technology offloading $O_i^{zz'}$ as

$$\min_{C_i^z, E_i^z, R_i^z, O_i^z, O_i^{zz'}, w^z, w^{zz'}} \frac{1}{T} \sum_t SW(t) \quad (28)$$

subject to

$$(1) - (7), (9), (10), (13.a) - (13.c) \\ (16.a), (16.b), (20.c), (21), (22), (24.a) - (24.d) \\ I_{col,i}^z(t) + I_{exe,i}^z(t) + I_{rel,i}^z(t) + I_{off,i}^z(t) \leq 1 \quad (28.a)$$

$$c_{ik}^z, e_{ik}^{z,q}, r_{ii',k}^z, o_{ij,k}^z, o_{ij,k'}^{zz'} \in \{0, 1\} \quad (28.b)$$

where the constraint (28.a) ensures that the M-MG i makes one decision per interface z at time t , i.e., collects, executes, relays, or offloads the data, and (28.b) summarizes the binary decision variables. This optimization problem can be easily modified to consider only one type of collaboration. Next, we model problem (28) as a Markov Decision Process (MDP) and solve it by a new multi-agent actor-critic algorithm.

B. Markov Decision Process

We reformulate the optimization problem (28) as a two-level MDP to capture the interactions between SPs and M-MGs.

1) *States*: The state of each M-MG agent $s_i(t)$ contains the local observations of the environment (devices information), the status of its buffer states, nearby APs of SP z and $z' \in \mathcal{Z}^c$, and allocated bandwidth to offload or relay the data. It also observes the interference at the previous time slots on the neighboring M-MGs. The local observations of the environment include the aggregated data size and elapsed time since the generation of the data packets in the coverage area of each M-MG.

The state of each SP agent $s_z(t)$ includes the status of all M-MGs associated with this SP and SP $z' \in \mathcal{Z}^c$, and the interference at the previous time slots on each M-MG's interface.

2) *Actions*: M-MGs collect data as they move, execute the data locally, and relay it to another M-MG or offload it to the corresponding AP, $\mathbf{a}_i^z(t) = [c_{ik}^z(t), e_{ik}^{z,q}(t), r_{ii',k}^z(t), o_{ij,k}^z(t), o_{ij,k'}^{zz'}(t)]$.

The action of each SP z , $\mathbf{a}^z(t) = [w_i^z(t), w_i^{zz'}(t)]$, consists of allocating bandwidth to a selected M-MG i to serve its traffic or the traffic from a collaborating SP $z' \in \mathcal{Z}^c$.

3) *Penalty*: Since there is collaboration among agents (M-MGs and SPs) to minimize the AoI and energy cost, all agents share the global penalty. The current penalty at epoch learning t for each agent g is $p_g(t) = \zeta_1 \bar{A}(t) + \varrho \zeta_2 \overline{cost}(t)$, $\forall g = \{1, \dots, M+1, \dots, M+Z\}$. To explore the global optimization of the 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, devising a comprehensive strategy to cover all state transitions of M-MGs and SPs poses a significant challenge. Due to the diverse nature of these entities and the complex interactions within the network, it is difficult to obtain a formatted strategy that adequately addresses every transition scenario. Therefore, we represent the interactions among entities in the network by $\mathcal{T}(\{s_i(t+1)\}, \{s_z(t+1)\} | \{s_i(t)\}, \{s_z(t)\}, \{a_i(t)\}, \{a_z(t)\})$.

C. Multi-Protocol Fed-Match Algorithm

We design a Multi-Protocol Multi-Agent Actor-Critic (MP-MAAC) framework to solve the previous optimization problem under different types of collaboration between SPs and M-MGs. It consists of primary actor and critic networks, as well as target actor and critic networks [24]. Its purpose is to control the M-MGs' data scheduling decisions and determine the optimal bandwidth allocation for SPs. To cope with the challenge of exploring a large state space as the network size increases, we adopt policy-based methods, such as Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG). These methods rely on dual neural networks to estimate the action-value function $Q(s, a)$ (as mentioned in [15]). By updating the parameters of target networks with the parameters of primary networks, the stability and convergence of the MP-MAAC improve. In Fig. 5, we illustrate the architecture of the MP-MAAC, where each agent (M-MG or SP) interacts with the environment to learn the best action that minimizes the system penalty p_g . The output of the critic network is the Q-value, which is a measure of the expected cumulative reward when different actions are taken in a given state. This Q-value is used to guide the actor network (policy network) to choose the action in the current state in order to minimize system penalty. To address the unstable approximated Q-values, we utilize an experienced replay buffer with a capacity of B and target networks. For each M-MG and SP agent, we construct a neural network with multiple inputs and outputs to learn the corresponding actions in different states, as shown in Fig. 5. Based on the collaboration between SPs and M-MGs, we have different inputs and outputs in the MP-MAAC network that result in different implementations of our multi-protocol Fed-Match algorithm, as described below:

1) *Fed-Match*: When there is no relaying and no cross-technology collaboration, each M-MG observes the positions of all IoT devices in its coverage area and their data size, the collection and execution buffers, the allocated bandwidth by each SP, and the location of APs within its coverage area. Then, the output of the M-MG is the decision on data collection, execution, or offloading to a particular AP. On the other hand, the SP observes the collection and execution buffer states, the locations of M-MGs, and the previous interference per offloading interface. Based on that, the output of the SP is the split of the available bandwidth among the M-MGs.

2) *Fed-Match-RC*: When there is relaying collaboration among the M-MGs, in addition to the previous observations, each M-MG observes the location, execution buffer states of candidate relaying M-MGs in its coverage area, the interference they have received from other M-MGs in its coverage

range, and the locations of the nearby APs. Based on the observation, the M-MG decides on data collection, execution, relaying, and offloading. In this case, the SPs also observe the interference received by relaying M-MGs.

3) *CT-Fed-Match*: When there is cross-technology collaboration between SPs, each M-MG observes the status of each execution buffer and the distance to nearby APs for each interface before making the offloading decision. The rest of the observations are the same as in Fed-Match.

4) *CT-Fed-Match-RC*: When there is relaying and cross-technology collaboration, the observations are as in Fed-Match-RC and CT-Fed-Match.

In all implementations of our algorithm, to promote exploration, we follow an ϵ -greedy policy, which introduces random actions with a probability ϵ . The parameters of the primary actor-network ($\mathcal{A}_g$), primary critic-network ($\mathcal{C}_g$), target actor-network ($\mathcal{A}'_g$), and target critic-network ($\mathcal{C}'_g$) for agent g are denoted as θ_g , ϕ_g , θ'_g , and ϕ'_g , respectively. The primary networks update the parameters of the target actor and critic networks every T_u period.

$$\begin{aligned}\theta'_g &= \tau\theta'_g + (1 - \tau)\theta_g \\ \phi'_g &= \tau\phi'_g + (1 - \tau)\phi_g\end{aligned}\quad (29)$$

where $\tau \in [0, 1]$ is the mixing weight. The learning rates of actor networks and critic networks are η_A and η_C , respectively. The critic networks are updated by minimizing the mean squared error loss function

$$l_{C_g}(\phi_g) := E[||\mathcal{C}_g(s_g, \mathbf{a}_g; \phi_g) - \hat{y}_g||^2] \quad (30)$$

where $\hat{y}_g$ is the estimated long-time Q value, $\hat{y}_g = p_g + \rho \mathcal{C}'_g(s'_g, \mathbf{a}'_g, \phi'_g)$ and 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

$$l_{A_g}(\theta_g) := \mathcal{C}_g(s_g, \mathcal{A}_g(s_g; \theta_g); \phi_g) \quad (31)$$

We design the network structure through multilayer perceptrons (MLPs). Based on both input states and output actions being multimodal, we need to consider that each network has different network layers and structures. For each actor network of M-MGs and SPs $\mathcal{A}_g$, we take the state of each agent $s_g(t)$ as input and the current action $\mathbf{a}_g(t)$ as output. For M-MG agents, we build a multi-input-output network with the perceptron as the basic unit and learn multiple actions by integrating multiple states, as described for each implementation of Fed-Match algorithm. For the SP actor network, we use MLPs to merge multiple state lists of M-MG and use the softmax function as an activation function to allocate bandwidth ratios for M-MG communication.

The critic network of each agent (M-MG or SP) $\mathcal{C}_g$ follows the same objective function to obtain the minimum average penalty. Moreover, we take the outputs of the actor networks and obtain an action-value function, $Q(s_g, \mathbf{a}_g)$. The MLPs architecture is used to combine the inputs of the critic networks, integrating the feature values of all input information through a concatenated layer to output the current Q-value. The main task of the critic network is to evaluate the quality of taking a certain action in each state by $Q(s, a)$. In this

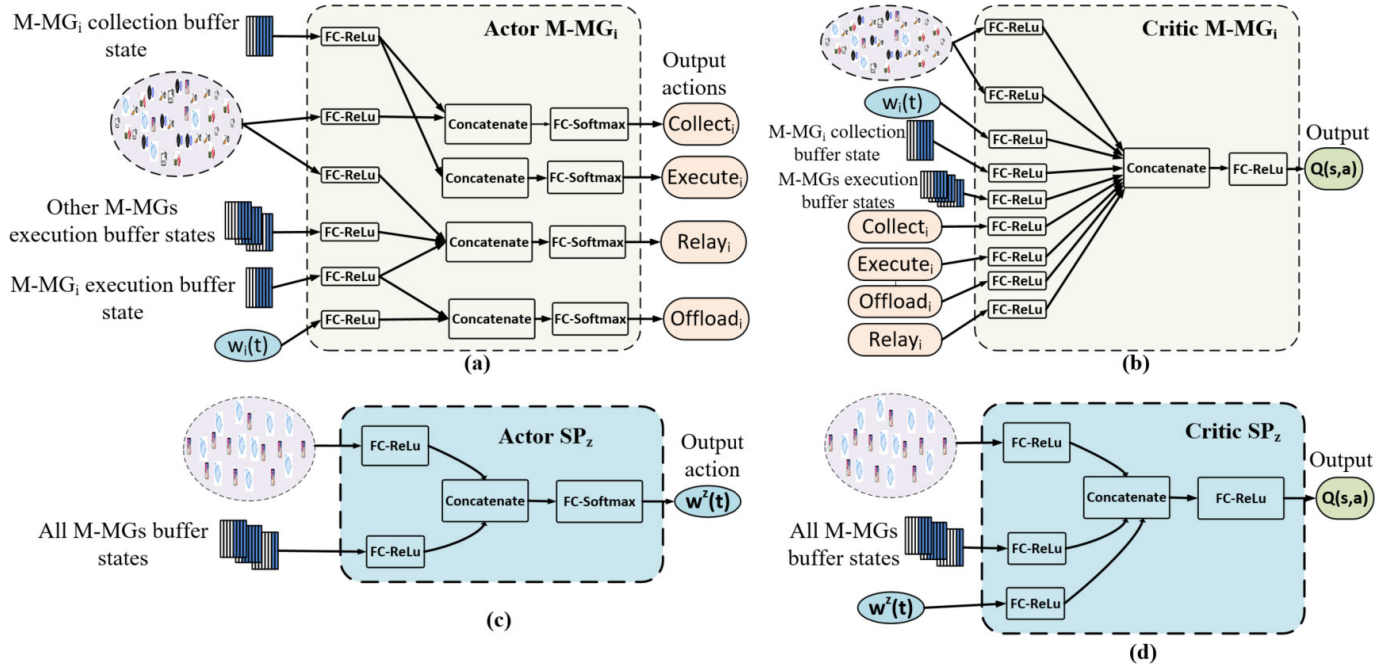


Fig. 5. Neural network models of each actor-critic agent.

regard, the critic needs to know the action taken by the actor network. Actor-critic algorithms typically use the critic's value estimate to calculate gradients to update the parameters of the actor network to minimize the loss function. By using the actor network's actions as input to the critic network, this design can provide the critic network with more information, helping to more accurately evaluate the quality of the state-action pair ($Q(s, a)$). This will allow the actor and critic networks to learn collaboratively to better optimize policies and improve network model performance.

Moreover, every E_f learning epoch, all M-MG agents (and all SP agents) share their actor network parameters and perform federated updating to improve the convergence. Each agent adopts the following update rule in which they maintain their own parameters, weighted by ω , and incorporate parameters from the other agents,

$$\theta_{M-MG}^{t+1} = \theta_{M-MG}^t \cdot \Omega_1 \quad (32)$$

$$\theta_{SP}^{t+1} = \theta_{SP}^t \cdot \Omega_2 \quad (33)$$

where $\theta_{M-MG}^t = [\theta_1^t, \dots, \theta_i^t, \dots, \theta_M^t]$, and $\theta_{SP}^t = [\theta_1^t, \dots, \theta_z^t, \dots, \theta_Z^t]$ are the vectors representing all M-MG actor network parameters and all SP actor network parameters at the learning epoch t . Ω_1 and Ω_2 refer to the federated updating matrix for M-MG and SP, respectively, obtained as

$$\Omega_n = \begin{bmatrix} \omega_n & \frac{1-\omega_n}{N_n-1} & \cdots & \frac{1-\omega_n}{N_n-1} \\ \frac{1-\omega_n}{N_n-1} & \omega_n & \cdots & \frac{1-\omega_n}{N_n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-\omega_n}{N_n-1} & \frac{1-\omega_n}{N_n-1} & \cdots & \omega_n \end{bmatrix} \quad (34)$$

where $n = 1$ and $N_1 = M$ for the M-MG federated learning factor and $n = 2$ and $N_2 = Z$ for SP federated learning. During the learning period, M-MGs retain the parameters with weights ω_1 and exchange the network parameters with weights $(1 - \omega_1)$. The same applies to SPs using weight ω_2 . Based on this proposed learning framework, we develop Fed-Match online collaboration algorithm, outlined in Algorithm 1, where the agents continuously learn and update optimal policies. The system adopts a collaborative model in which global optimality is obtained using federated learning. In this way, M-MGs obtain faster the offloading policy that results in the minimum penalty, and the SP's policy training is also accelerated to allocate the bandwidth more rationally for each M-MG's interface. This will reduce packet stagnation at M-MGs and the AoI. In addition, there is no need to exchange data messages. Thus, the communication cost of sharing model-level parameters can be neglected [15], improving the communication efficiency of the system.

VII. NUMERICAL RESULTS

We have conducted extensive simulations to illustrate the performance of our proposed architecture and multi-protocol Fed-Match algorithm. We compare the different implementations of our algorithm (i.e., Fed-Match [11], CT-Fed-Match, Fed-Match-RC, and CT-Fed-Match-RC), and existing algorithms like deep deterministic policy gradient (DDPG) and multi-agent DDPG (MADDPG) [15] with different levels of FL. Unless otherwise stated, the simulation parameters are given in Table III. We set the IoT environment on a 300×300 map with 15 IoT devices per protocol (i.e., Zigbee, Cellular, WiFi, and LoRa), 14 M-MGs with 4 interfaces each, 4 SPs, 4 APs, and 8000 maximum number of learning epochs. To simplify the selection of the relaying M-MGs, we assume that the mobility path of each M-MG is known. However, our approach can be applied to uncertain mobility scenarios

Algorithm 1 Multi-Protocol Fed-Match Online Collaboration

```

1: Initialize: Hyper parameters of learning algorithms, the
   primary networks' parameters  $(\theta_{M-MG})_i, (\theta_{SP})_z$ ,
    $(\phi_{M-MG})_i, (\phi_{SP})_z$ , and target networks' parameters:
    $(\theta_{M-MG})'_i \leftarrow (\theta_{M-MG})_i, (\theta_{SP})'_z \leftarrow (\theta_{SP})_z$ ,
    $(\phi_{M-MG})'_i \leftarrow (\phi_{M-MG})_i, (\phi_{SP})'_z \leftarrow (\phi_{SP})_z$ .
2: for epoch  $t = 1$  to  $\max\_epoch$  do
3:   Generate  $\nu \in [0, 1]$  randomly;
4:   for each agent  $g$  in  $\{1, \dots, M, \dots, M + Z\}$  do
5:     if  $\nu < \epsilon$  or  $|\mathcal{B}[g]| < B$  then
6:       for each interface  $z$  in  $\{1, \dots, Z\}$  do
7:         if  $dist_{ij}^z > r_{obs}^i, dist_{ij}^{z'} \leq r_{obs}^i$ , and  $B_{exe,i}^{z'}$  has space
8:            $M - MG_i$  performs cross-technology collaboration:
            $M - MG_i$  transfers a packet of interface  $z$  from the  $B_{exe,i}^{z'}$ 
           to the  $B_{exe,i}^{z'}$ .
9:         end if
10:      end for
11:     for each interface  $z$  in  $\{1, \dots, Z\}$  do
12:       if  $dist_{ij}^z \leq r_{obs}^i$ 
13:          $M - MG_i$  performs offloading action randomly;
14:       else if  $dist_{ij}^z > r_{obs}^i$  and  $B_{exe,i'}^{z'}$  has space
15:          $M - MG_i$  performs relaying action randomly;
16:       end if
17:     end for
18:     Choose actions  $\mathbf{a}_g(t)$  randomly;
19:   else
20:     Ensemble local observation and states:  $\mathbf{s}_g(t)$ ;
21:     for each interface  $z$  in  $\{1, \dots, Z\}$  do
22:       if  $dist_{ij}^z > r_{obs}^i, dist_{ij}^{z'} \leq r_{obs}^i$ , and  $B_{exe,i}^{z'}$  has space
23:          $M - MG_i$  performs cross-technology collaboration:
          $M - MG_i$  transfers a packet of interface  $z$  from the  $B_{exe,i}^{z'}$ 
         to the  $B_{exe,i}^{z'}$ .
24:       end if
25:     end for
26:     for each interface  $z$  in  $\{1, \dots, Z\}$  do
27:       if  $dist_{ij}^z \leq r_{obs}^i$ 
28:          $M - MG_i$  sets offloading action:  $\mathbf{a}_i(t) = \mathbf{o}_{ij}^z(t)$ 
29:       else if  $dist_{ij}^z > r_{obs}^i$  and  $B_{exe,i'}^{z'}$  has space
30:          $M - MG_i$  sets relaying action:  $\mathbf{a}_i(t) = \mathbf{r}_{ii'}^{z'}(t)$ 
31:       end if
32:     end for
33:     Set actions:  $\mathbf{a}_g(t) = \mathcal{A}_g(\mathbf{s}_g(t); \theta_g)$ 
34:   end if
35: end for
36: Interact with environment and obtain  $p(t), \mathbf{s}'(t+1)$ ;
37: Add  $\{\mathbf{s}, \mathbf{a}, p, \mathbf{s}'\}$  into  $\mathcal{B}$ ;
38: for each agent  $g$  in  $[1, \dots, M, \dots, M + Z]$  do
39:   if  $|\mathcal{B}[g]| \geq B$  then
40:     Sample  $\{\mathbf{s}_g, \mathbf{a}_g, p_g, \mathbf{s}'_g\}$  from  $\mathcal{B}[g]$ ;
41:     Predict new actions:  $\mathbf{a}'_g = \mathcal{A}'_g(\mathbf{s}'_g; \theta'_g)$ ;
42:     Predict new  $Q$ -value:
43:      $Q'(\mathbf{s}'_g, \mathbf{a}'_g) = \mathcal{C}'_g(\mathbf{s}'_g, \mathbf{a}'_g; \phi'_g)$ ;
44:     Calculate  $\hat{y}_g$ ;
45:     Calculate  $l_{C_g}(\phi_g), l_{A_g}(\theta_g)$  by (30) and (31);
46:     Update network parameters:
47:      $\phi_g^{t+1} \leftarrow \phi_g^t - \eta_C \nabla_{\phi} \tilde{l}_{C_g}(\phi_g^t)$ 
48:      $\theta_g^{t+1} \leftarrow \theta_g^t - \eta_A \nabla_{\theta} \tilde{l}_{A_g}(\theta_g^t)$ 
49:   end if
50: end for
51: if  $t \bmod T_u == 1$  then
52:   Update target actor and critic networks using (29);
53: end if
54: if  $t \bmod E_f == 1$  then
55:   Run M-MG-federated updating using (32);
56:   Run SP-federated updating using (33);
57: end if
58: end for

```

TABLE III
MAIN PARAMETER SETTINGS

Parameter	Value	Parameter	Value
λ_k, φ_k^c	500b/slot, 0.3	P_i	0.4W
r_{move}^i, r_{obs}^i	6, 80	ρ, τ, ϵ	0.85, 0.8, 0.2
$r_{col,ZB}^i$	25	$B_{col,i}^z, \zeta_2$	8, 0.5
$r_{col,C}^i$	40	$B_{exe,i}^z, \zeta_1$	8, 0.5
$r_{col,Wf}^i$	15	T_u, E_f	8
$r_{col,LR}^i$	80	η_A, η_C	$1 \times 10^{-3}, 2 \times 10^{-3}$
W_{b1}^{ZB}, W_{b1}^C	20 MHZ, 20 MHZ	$B, \varrho, \mathcal{X}$	32, 0.0008, 4
W_{b1}^{WF}, W_{b1}^{LR}	80 MHZ, 500 KHZ	ξ_i, ξ_j	10^{-22} W
W_{b2}^{ZB}, W_{b2}^C	20 MHZ, 20 MHZ	$n_{W,b1}$	3
W_{b2}^{WF}, W_{b2}^{LR}	80 MHZ, 500 KHZ	$PL_{ZB,b1}(d_0)$	46.21
$\gamma_{ZB,b1}$	3.3	$PL_{C,b1}(d_0)$	42.86
$\gamma_{C,b1}$	1.86	$\sigma_{C,b1}$	3.25
$PL_{WF,b2}(d_0)$	47	$n_{WF,b2}$	4
$PL_{C,b2}(d_0)$	57.1	$PL_{ZB,b2}(d_0)$	42.13
$\gamma_{ZB,b2}$	2.3	$\sigma_{ZB,b2}$	8.9
$\sigma_{LR,b2}$	6	$PL_{WF,b1}(d_0)$	30
$\sigma_{ZB,b1}$	11.1	Map size	300 X 300

with the consequent increase in convergence time. We consider that WiFi, Zigbee, Cellular, and LoRa work at 2.4 GHz (band b1) with the path loss models described in [25], [26], [27], and [28]. In addition, WiFi and Cellular also work at 5 GHz following the path loss models in [29], and [30], respectively, while Zigbee and LoRa work at 915 MHz as in [26], and [31] (bands b2), respectively. The values of the path loss parameters are given in Table III, where $PL_{z,b}(d_0)$ is the power loss for wireless protocol z at reference distance d_0 and band $b \in \{b_1, b_2\}$, $n_{z,b}$ is the path loss coefficient for protocol z on b , $\gamma_{z,b}$ is the path loss exponent for z on b , and $S_{z,b}$ is a zero mean random variable with normal distribution and standard deviation $\sigma_{z,b}$.

A. Collaboration Between SPs and M-MGs for Data Scheduling, Relaying, and Spectrum Allocation (Fed-Match-RC)

First, we evaluate the impact of the relaying collaboration between M-MGs (with no cross-technology collaboration). We consider different settings with APs located at the center of the map (CAP) and at the edges (EAP) to evaluate the impact of the AP's location on the relay strategy. We consider Fed-Match algorithm as a baseline, which has no relaying and no cross-technology collaboration. As shown in Fig. 6, the average AoI of Fed-Match-RC(EAP) is 3 times lower compared to Fed-Match(EAP). The average AoI of Fed-Match-RC(CAP) is also lower than Fed-Match(CAP). However, in this case, with the APs at the center, there are more chances that the M-MGs are within the coverage area of an AP. Thus, the improvement of relaying is less significant. On the other hand, if the APs are at the edges, the M-MGs will have other communication paths to deliver their data to the corresponding AP, resulting in higher improvement. Besides, the relaying collaboration reduces interference, transmission delay, and congestion. Since more paths are explored to offload the data, we achieve a higher data rate and lower AoI and energy consumption.

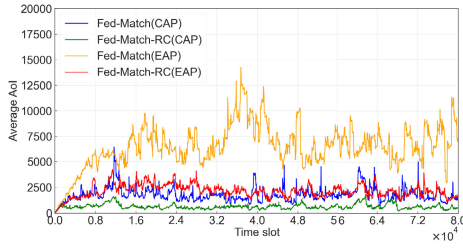


Fig. 6. Average AoI vs time slot.

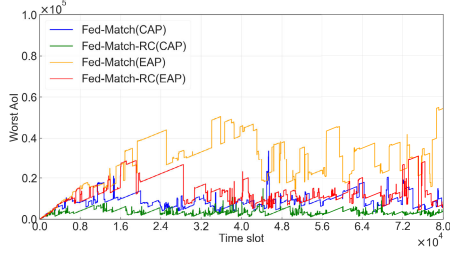


Fig. 7. Worst AoI vs time slot.

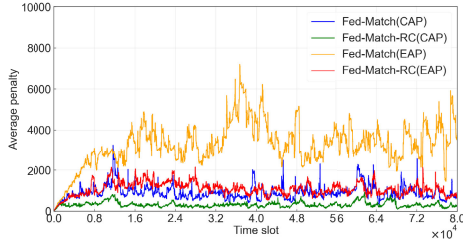


Fig. 8. Average penalty vs time slot.

In Fig. 7, we can see that the worst AoI of Fed-Match-RC(CAP) and Fed-Match-RC(EAP) are smaller and more stable compared to the cases without relaying collaboration. The collaboration between M-MGs brings more flexibility to adapt to the network condition and resource availability since there are more opportunities for the data to be offloaded to different APs. In contrast, without relaying collaboration Fed-Match(CAP) and Fed-Match(EAP), M-MGs have to wait to be in the coverage of an AP to offload the data. Moreover, as shown in Fig. 8, we see that Fed-Match-RC(CAP) and Fed-Match-RC(EAP) have lower and more stable penalty values. The penalty value obtained with Fed-Match-RC(EAP) is half the one with Fed-Match(EAP) because relaying make the training of the algorithm more stable. Furthermore, Fed-Match-RC(CAP) and Fed-Match-RC(EAP) can collect more packets than their respective counterparts without relaying, as shown in Fig. 9. In the same time period, Fed-Match-RC(CAP) and Fed-Match-RC(EAP) can receive more than 60,000 and 40,000 packets than Fed-Match(CAP) and Fed-Match(EAP), respectively. This gap increases in time since the AP can receive more data packets.

Next, we analyze the performance of the neural network models using the critic loss functions of M-MG and SPs as shown in Fig. 10 and Fig. 11, respectively. We can see that the critic loss functions of M-MGs and SPs are more stable and converge faster in Fed-Match-RC(CAP) and Fed-Match-

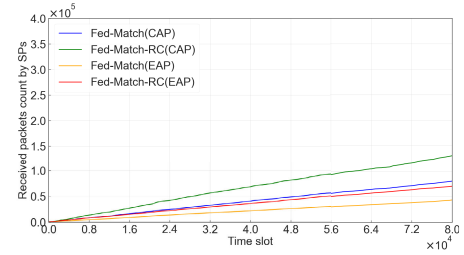


Fig. 9. Packets received vs time slot.

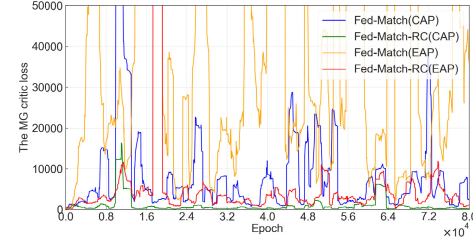


Fig. 10. M-MG critic loss vs epoch.

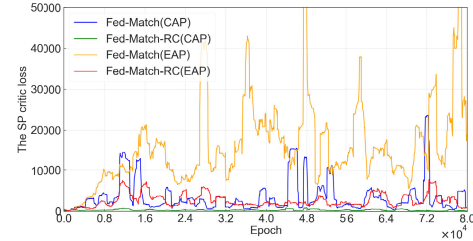


Fig. 11. SP critic loss vs epoch.

RC(EAP) while their baselines have significant fluctuations. This means that network models with relaying are more adaptable to different data inputs and network changes since there are more transmission opportunities. As a result, there is more information available about the network and, thus, a better understanding of the dynamics and predictions. All of which makes the critic loss function more stable.

B. Cross-Technology Collaboration Between SPs and M-MGs (CT-Fed-Match)

To assess the performance of cross-technology collaboration, we place each AP from a different SP at each edge of the map to exploit the collaboration between SPs that have APs at different locations. We consider the Fed-Match algorithm as the baseline, and compare the performance with and without relaying. As shown in Fig. 12, the CT-Fed-Match-RC(EAP) algorithm achieves 30 times lower AoI than Fed-Match (EAP). If we focus only on the cross-technology aspect, we can see that the AoI with CT-Fed-Match(EAP) algorithm is 9 times lower than the Fed-Match baseline.

In Fig. 13 and Fig. 14, we compare the worst AoI and penalty of these four cases. Notably, the CT-Fed-Match-RC(EAP) has the lowest penalty and the lowest worst AoI. As explained before, the multi-hop transmission can effectively reduce the delay of data packets, thereby reducing the worst AoI and penalty. In addition, the cross-technology

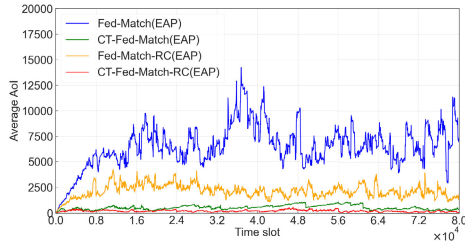


Fig. 12. Average AoI vs time slot.

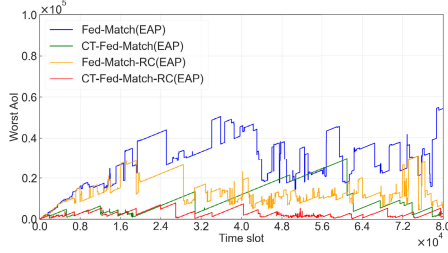


Fig. 13. Worst AoI vs time slot.

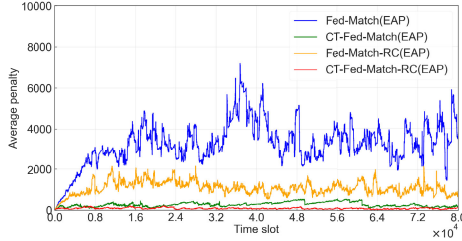


Fig. 14. Average penalty vs time slot.

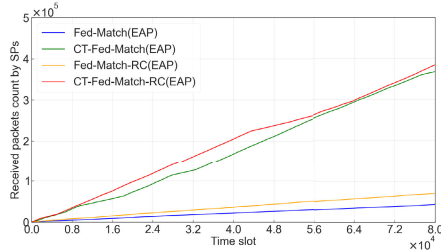


Fig. 15. Packets received vs time slot.

collaboration reduces the storage time of the data packet in the M-MGs, improves the load balance in the network, and thus, it further reduces the transmission delay and the worst AoI.

As shown in Fig. 15, the cross-technology collaboration can also improve the overall transmission capacity of the network. In the case with both relaying and cross-technology collaboration, 5 times more packets are offloaded than in the case with only relaying collaboration and 8 times more than with the baseline. In Fig. 16 and Fig. 17, we evaluate the convergence and stability of our approach using the loss function of the critic network of M-MG and SP, respectively, for all four implementations. The CT-Fed-Match-RC(EAP) has the lowest critic training loss, which means that this network has the best performance and can make the most accurate predictions. On the other hand, the critic loss of the baseline

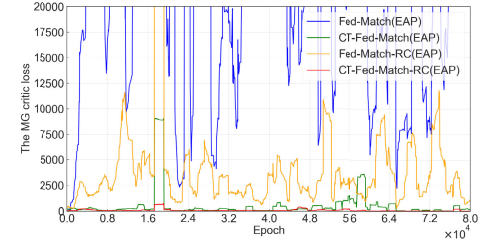


Fig. 16. M-MG critic loss vs epoch.

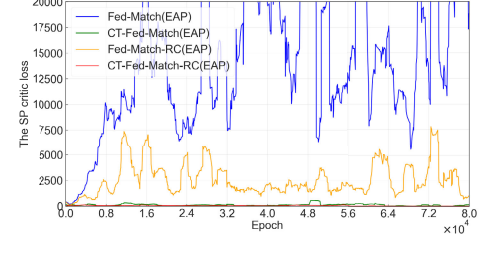


Fig. 17. SP critic loss vs epoch.

TABLE IV

AVERAGE RELAY TIME AND PROPORTION OF PACKETS RELAYED PER PACKET IN EACH PROTOCOL

	Case	WiFi	Zigbee	LoRa	Cellular	Total
Average relay time	Fed-Match-RC(EAP)	3.99	3.42	2.19	2.55	3.04
	CT-Fed-Match-RC(EAP)	2.37	2.46	2.58	2.43	2.45
Proportion of packets relayed	Fed-Match-RC(EAP)	29.34%	26.93%	41.11%	32.94%	32.58%
	CT-Fed-Match-RC(EAP)	23.46%	20.61%	30.92%	25.20%	25.05%

is the highest and fluctuates the most, which means that the relay strategy and cross-technology collaboration strategy can significantly improve the network prediction. The lower delay helps reduce the prediction error of the critic network, which improves the training process and the critic network's prediction accuracy, thereby reducing the loss function value.

In Table IV, we compare the performance of CT-Fed-Match-RC(EAP) with Fed-Match-RC(EAP) per interface in terms of the average relaying time and number of packets relayed. The average relaying time indicates the average time needed to relay a packet until offloaded to an AP. It depends on the available bandwidth and network condition. The lower the capacity, the higher the relaying time. We can see that the average relaying times on WiFi, Zigbee, and Cellular with CT-Fed-Match-RC(EAP) are lower than those with Fed-Match-RC(EAP). This is because the interference is reduced with cross-technology collaboration, and more candidate APs are available, reducing the transmission range. The relaying time for LoRa is higher with CT-Fed-Match-RC(EAP) because it has the largest coverage, and thus, many packets from other interfaces are offloaded using LoRa. In addition, we also evaluate the fraction of packets relayed by each interface with respect to the total number of packets offloaded in Table IV. The total average relay ratio of CT-Fed-Match-RC(EAP) is only 25%, which means that 75% of the data packets have been offloaded directly. On the other hand, with Fed-Match-RC(EAP), only two-thirds of packets can be successfully offloaded directly to the AP. This shows that cross-technology collaboration reduces the number of transmissions and, thus, the energy cost.

TABLE V
THE NUMERICAL COMPARISONS ON MEAN AND STD

Approach	Average AoI		Penalty	
	mean	std	mean	std
Fed-Match(CAP)	1784.19	670.81	892.12	335.41
Fed-Match-RC(CAP)	632.72	243.55	316.39	121.78
Fed-Match(EAP)	6503.63	2050.16	3251.84	1025.08
CT-Fed-Match(EAP)	544.80	210.36	272.43	105.18
Fed-Match-RC(EAP)	2150.89	616.29	1075.47	308.15
CT-Fed-Match-RC(EAP)	210.43	81.76	105.24	40.88
Fed-Match(EAP) single band	7215.75	2128.68	3607.90	1064.34
CT-Fed-Match-RC-with-Interference(EAP)	210.42	81.75	105.24	40.87
CT-Fed-Match-RC-no-Interference(EAP)	233.54	116.83	116.79	58.41
CT-Fed-Match-RC-Random-move(EAP)	104.61	40.35	84.31	31.69
CT-Fed-Match-RC-Fixed-position(EAP)	964.35	363.17	778.85	283.46

In Table V, we analyze the average AoI and penalty for the different implementations of Fed-Match. Comparing Fed-Match-RC(CAP) with its baseline, the former has a lower mean and standard deviation in AoI and penalty. Thus, relaying can improve the system's communication capability and network stability. By comparing CT-Fed-Match-RC(EAP) to its baseline, we can see that in a more challenging communication environment with the M-MGs at the edges of the map, our scheme can significantly improve learning ability. In fact, CT-Fed-Match-RC(EAP) has the lowest average values and standard deviations. As can be seen in Table V, the average AoI and average penalty of CT-Fed-Match-RC-with-Interference and CT-Fed-Match-RC-no-Interference are very similar since our proposed scheme considers the impact of interference in the learning model for CT and RC. To assess the impact of mobility on the performance of our scheme, we have compared our proposed scheme with M-MGs moving randomly (CT-Fed-Match-RC-Random-move) to static M-MGs (CT-Fed-Match-RC-Fixed position). If M-MGs remain in a fixed location, they are unable to collect and offload data from some IoT devices, which results in 9 times higher AoI and penalty in CT-Fed-Match-RC-Fixed compared to CT-Fed-Match-RC-Random-move.

C. Comparison With Existing Algorithms

Finally, we compare our Fed-Match algorithm with two popular reinforcement learning algorithms, DDPG and MADDPG [15], and different levels of FL. We use the baseline Fed-Match(CAP) for comparison. We assume that all APs are located at the center of the map and that the coverage range of APs equals the map size. We set the parameters to include 7 M-MGs and 100 IoT sensors (25 sensors each for WiFi, Zigbee, Cellular, and LoRa) on a 100×100 map. The maximum number of learning epochs is 25,000. Additionally, we also considered three different versions with respect to the levels of FL: only one level of FL between M-MGs (Fed-Match M-MG-FL), one level between SPs (Fed-Match SP-FL), and two levels of FL update between M-MGs and between SPs (Fed-Match M-MG_FL & SP_FL). We have reduced the map size, and the number of M-MGs to avoid having a long training time for the DDPG framework. We attribute this phenomenon to the necessity of employing larger neural network models with intricate structures in centralized collaboration

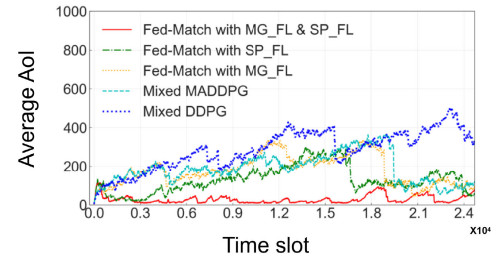


Fig. 18. Average AoI vs time slot.

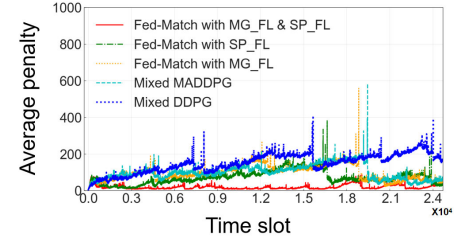


Fig. 19. Average penalty vs time slot.

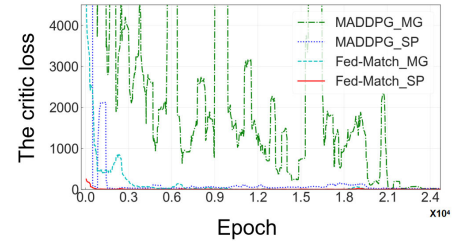


Fig. 20. Critic loss vs epoch.

algorithms. These models are required to effectively capture the relationships between extensive global input states and the individual local policies of each M-MG agent, thereby posing challenges during the training process.

In Fig. 18, we can see that our baseline Fed-Match with two levels of FL (between M-MGs and SPs) has an average AoI 40 times lower than DDPG. By implementing Fed-Match with FL considered only for SPs or M-MGs, we observe the algorithm's convergence is around 16,500 epochs for SPs and approximately 19,000 epochs for M-MGs. On the other hand, our algorithm with 2-levels of FL converges almost instantaneously. In Fig. 19, we can see similar performance improvements in terms of penalty. Fed-Match achieves the lowest penalty, indicating enhanced bandwidth allocation efficiency for SPs and faster data collection and offloading for M-MGs. This outcome highlights the advantages of employing interactive policies between M-MGs and SPs to minimize the penalty. In fact, we can see that compared with a single learning agent (DDPG algorithm), the interactions of multi-agents in our algorithm can effectively improve the communication ability of the system. At the same time, multi-agent collaboration is realized through federated learning, and the exchange of model information can effectively improve data transmission and make the system more stable. In Fig. 20, we compare the training loss of the critic networks for M-MGs and SPs for both MADDPG and Fed-Match. Here, our

scheme incorporates MADDPG and two-level FL to achieve multi-agent collaboration. Notably, in Fed-Match, we observe rapid convergence of the critic loss for both M-MGs and SPs, indicating a faster convergence to the optimal strategy when implementing FL simultaneously between M-MGs and between SPs.

VIII. CONCLUSION

In this paper, we present a cross-technology IoT architecture design to enable timely data collection in heterogeneous IoT networks under different protocols and spectrum bands. The objective is to minimize the AoI and energy consumption by jointly optimizing collaboration between M-MGs and SPs for bandwidth allocation, relaying, and cross-technology data scheduling. Collaborative policies are presented based on a new federated matching framework in which M-MGs and SPs learn their strategies in a distributed manner. The numerical results show that our CT-Fed-Match-RC algorithm with cross-technology and relaying collaboration reduces both AoI and energy consumption significantly compared with existing approaches.

REFERENCES

- [1] Y. Liu, M. Kashef, K. B. Lee, L. Benmohamed, and R. Candell, "Wireless network design for emerging IIoT applications: Reference framework and use cases," *Proc. IEEE*, vol. 107, no. 6, pp. 1166–1192, Jun. 2019.
- [2] T. Theodorou, G. Violettas, P. Valsamas, S. Petridou, and L. Mamatras, "A multi-protocol software-defined networking solution for the Internet of Things," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 42–48, Oct. 2019.
- [3] A. Hasan and B. M. Khan, "Coexistence management in wireless networks—A survey," *IEEE Access*, vol. 10, pp. 38600–38624, 2022.
- [4] Z. Chi, Y. Li, H. Sun, Y. Yao, and T. Zhu, "Concurrent cross-technology communication among heterogeneous IoT devices," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 932–947, Jun. 2019.
- [5] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He, "Boosting the bitrate of cross-technology communication on commodity IoT devices," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1069–1083, Jun. 2019.
- [6] D. Gao, L. Wang, and B. Hu, "Spectrum efficient communication for heterogeneous IoT networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 3945–3955, Nov. 2022.
- [7] X. Xie, H. Wang, and X. Liu, "Scheduling for minimizing the age of information in multi-sensor multi-server industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 573–582, Jan. 2023.
- [8] X. Wang, M. Yi, J. Liu, Y. Zhang, M. Wang, and B. Bai, "Cooperative data collection with multiple UAVs for information freshness in the Internet of Things," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2740–2755, May 2023.
- [9] H. Wang, X. Xie, and J. Yang, "Optimizing average age of information in industrial IoT systems under delay constraint," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 1–10, Oct. 2023.
- [10] Z. Li, P. Tong, J. Liu, X. Wang, L. Xie, and H. Dai, "Learning-based data gathering for information freshness in UAV-assisted IoT networks," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2557–2573, Feb. 2023.
- [11] H. H. Esmat, X. Xia, B. Lorenzo, and L. Guo, "Multi-protocol aware federated matching for architecture design in heterogeneous IoT," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2022, pp. 6277–6282.
- [12] Q. Zhang, H. Meng, Z. Feng, and Z. Han, "Resource scheduling of time-sensitive services for B5G/6G connected automated vehicles," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 14820–14833, Aug. 2022.
- [13] O. S. Oubbati, M. Atiquzzaman, H. Lim, A. Rachedi, and A. Lakas, "Synchronizing UAV teams for timely data collection and energy transfer by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6682–6697, Jun. 2022.
- [14] S. Wang et al., "Distributed reinforcement learning for age of information minimization in real-time IoT systems," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 501–515, Apr. 2022.
- [15] Z. Zhu, S. Wan, P. Fan, and K. B. Letaief, "Federated multiagent actor-critic learning for age sensitive mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1053–1067, Jan. 2022.
- [16] X. Xie, H. Wang, and M. Weng, "A reinforcement learning approach for optimizing the age-of-computing-enabled IoT," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2778–2786, Feb. 2022.
- [17] M. Zhang, A. Arafa, J. Huang, and H. V. Poor, "Pricing fresh data," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1211–1225, May 2021.
- [18] M. Zhang, A. Arafa, E. Wei, and R. Berry, "Optimal and quantized mechanism design for fresh data acquisition," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1226–1239, May 2021.
- [19] X. Wang and L. Duan, "Dynamic pricing and mean field analysis for controlling age of information," *IEEE/ACM Trans. Netw.*, vol. 30, no. 6, pp. 2588–2600, Dec. 2022.
- [20] N. Modina, R. El Azouzi, F. De Pellegrini, D. S. Menasche, and R. Figueiredo, "Joint traffic offloading and aging control in 5G IoT networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4714–4728, Aug. 2023.
- [21] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," 2018, 1803.11485.
- [22] J. Kerdari and T. Veeraklaew, "Visualization of spatial distribution of random waypoint mobility models," *J. Comput.*, vol. 12, no. 4, pp. 309–316, 2017.
- [23] C. T. Kelley, "Numerical methods for nonlinear equations," *Acta Numerica*, vol. 27, pp. 207–287, May 2018.
- [24] H. H. Esmat and B. Lorenzo, "Self-learning multi-mode slicing mechanism for dynamic network architectures," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1048–1063, Apr. 2024.
- [25] M. Hidayab, A. H. Ali, and K. B. A. Azmi, "Wifi signal propagation at 2.4 GHz," in *Proc. Asia Pacific Microw. Conf.*, 2009, pp. 528–531.
- [26] T. O. Olasupo, "Propagation modeling of IoT devices for deployment in multi-level hilly urban environments," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 346–352.
- [27] D. Wang, L. Song, X. Kong, and Z. Zhang, "Near-ground path loss measurements and modeling for wireless sensor networks at 2.4 GHz," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 8, Aug. 2012, Art. no. 969712, doi: 10.1155/2012/969712.
- [28] T. Janssen, N. BniLam, M. Aernouts, R. Berkvens, and M. Weyn, "LoRa 2.4 GHz communication link and range," *Sensors*, vol. 20, no. 16, p. 4366, Aug. 2020.
- [29] Y. Jian, S. Lall, and R. Sivakumar, "Toward a self-positioning access point for WiFi networks," in *Proc. 16th ACM Int. Symp. Mobility Manage. Wireless Access*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 19–28.
- [30] K. Yonezawa, T. Maeyama, H. Iwai, and H. Harada, "Path loss measurement in 5 GHz macro cellular systems and consideration of extending existing path loss prediction methods," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2004, pp. 279–283.
- [31] L. E. Ribeiro, D. W. Tokikawa, J. L. Rebelatto, and G. Brante, "Comparison between LoRa and NB-IoT coverage in urban and rural southern Brazil regions," *Ann. Telecommun.*, vol. 75, nos. 11–12, pp. 755–766, Dec. 2020.