Energy Harvesting Aware Multi-hop Routing Policy in Distributed IoT System Based on Multi-agent Reinforcement Learning

Wen Zhang  
Texas A&M University–Corpus Christi  
Corpus Christi, USA  
wzhang3@islander.tamucc.edu

Tao Liu  
Department of Math and Computer Science  
Lawrence Technological University  
Southfield, USA  
tliu3@ltu.edu

Mimi Xie  
Department of Computer Science  
University of Texas at San Antonio  
San Antonio, USA  
mimi.xie@utsa.edu

Longzhuang Li  
Texas A&M University–Corpus Christi  
Corpus Christi, USA  
longzhuang.li@tamucc.edu

Dulal Kar  
Texas A&M University–Corpus Christi  
Corpus Christi, USA  
dulal.kar@tamucc.edu

Chen Pan*  
Texas A&M University–Corpus Christi  
Corpus Christi, USA  
chen.pan@tamucc.edu

Abstract—Energy harvesting technologies offer a promising solution to sustainably power an ever-growing number of Internet of Things (IoT) devices. However, due to the weak and transient natures of energy harvesting, IoT devices have to work intermittently rendering conventional routing policies and energy allocation strategies impractical. To this end, this paper, for the very first time, developed a distributed multi-agent reinforcement algorithm known as global actor-critic policy (GAP) to address the problem of routing policy and energy allocation together for the energy harvesting powered IoT system. At the training stage, each IoT device is treated as an agent and one universal model is trained for all agents to save computing resources. At the inference stage, packet delivery rate can be maximized. The experimental results show that the proposed GAP algorithm achieves $\sim 1.28 \times$ and $\sim 1.24 \times$ data transmission rate than that of the Q-table and ESDSRAA algorithm, respectively.

I. INTRODUCTION

The emerging Internet of Things (IoT) is becoming increasingly popular. From wearable devices to environmental sensors, the growing number of IoT devices continue weaving a seamless web that benefits each corner of our daily life. There are more than 50 billion embedded IoT devices worldwide [1]. How to power these billions of distributed embedded IoT devices sustainably becomes as a pressing issue. Out of all viable solutions, energy harvesting (EH) becomes one of the most preferable substitutes to the battery to addresses the cost, sustainability, and environmental concerns [2], [3]. However, due to the weak and transient nature of energy harvesting, IoT devices have to work intermittently, rendering conventional routing selection impractical. If a receiver experiences a power outage, the transmitter has to re-transmit all the data, resulting in unpleasant energy dissipation. Besides, routing loop avoidance is also challenging which needs frequent flooding [4]. Yet, those methods require adequate and reliable energy supply, which is unattainable under EH environments.

Existing studies [5], [6] have made efforts to improve the reliability and energy efficiency of routing selection by either finding the shortest routing path or minimizing the transmission time for the small-scale IoT systems powered by EH. However, the packet loss and energy depletion brought by the power intermittency are still yet to be solved. Since the data delivery is related to multiple tasks such as sensing and communication on embedded IoT devices, with a limited power budget, energy allocation for sensing will inevitably affect the energy allocation for communication making tasks on embedded IoT devices interdependent to each other. Such interdependence, however, offers us an opportunity to ameliorate the power intermittency problem. Specifically, by integrating energy allocation into routing selection, each embedded IoT device becomes capable of controlling the time and amount of data for communication with the proper energy allocation among runtime tasks. Therefore, with the joint optimization of the routing selection and energy allocation, we can help maximize both the sink received data and the energy efficiency.

Nevertheless, the uncertainty of harvesting power creates a partially observable environment. In multi-hop networks, each node’s current decisions influence the future. The current optimal energy allocation that maximizes the data transmission might result in insufficient energy for future data transmission. Moreover, the optimal decisions of a single IoT device also have an influence on its neighbors, which makes decisions of all IoT devices entangled. Therefore, whenever a single device is optimizing energy allocation, it also needs to consider whether this decision is long-term optimal or global optimal, which is crucial but has barely been explored. To this end, we are in a pressing need of joint-optimization of energy allocation and routing selection for each individual to address the complexity under this partially observable multi-hop environment. Of all viable solutions, Deep Reinforcement Learning (DRL) [7] has been emerged as one of the best candidates. By naturally formulating a joint-optimization problem of energy allocation and routing selection as a partially observable Markov Decision Process (POMDP), DRL is able to make a sequence of decisions under uncertainty with an
outstanding performance from a long-term perspective.

Since decentralized networks are more commonly implemented in real-life IoT systems than well researched centralized networks, in this paper, for the very first time, a multi-agent DRL algorithm has been proposed to conduct joint-optimization for energy allocation and multi-hop routing to address the aforementioned concerns. In a nutshell, the major contributions of this paper are as follows:

1) A comprehensive multi-hop data transmission models for energy-harvesting powered distributed IoT networks (as shown in Fig. 1) has been developed.

2) A global training distributed deploy multi-agent reinforcement learning (MARL) algorithm known as global actor-critic policy (GAP) has been proposed to address the aforementioned POMDP for the long-term optimality.

3) A spatial reward mechanism is proposed for the entangled problem optimization of the multi-hop EH IoT system.

The remainder of this paper is organized as follows. Section II discusses related work. Then, Section III builds an EH-powered distributed IoT system model. After that, Section IV formulates the joint-optimization problem and proposes the corresponding DRL algorithm known as GAP. Finally, Section V conducts experiments to demonstrate the performance of GAP and Section VI concludes this work.

II. RELATED WORK

Traditional Methods: Nguyen et al. [8] proposed an EH routing algorithm for multi-hop heterogeneous IoT networks where the energy prediction model and the energy back-off parameters are integrated into the proposed routing algorithm. [9] jointly optimizes the power allocation and routing selection for the EH Multi-hop Cognitive Radio Networks. However, these solutions are not designed for long-term optimization, resulting in throttled system performance. [10] developed an algorithm, named ESDRARA, to explore multi-hop routing for EH IoT systems with energy-harvesting-aware geographic routing and different energy allocation strategies. However, it did not consider the uncertainty of the power source and the inter-dependency of each device.

DRL-based Methods: DRL shows outstanding performance on decision making in uncertain environments considering the long-term influence of its actions [11]–[13]. [11] proposed a multi-layer Markov fluid queue model to optimize the transmission by maximizing the reward for individual IoT devices rather than the multi-hop communication system. [12] employed an algorithm to optimize the power allocation for two-hop EH communications. Q-table is created in [13] to find the optimal routing path for EH multi-hop cognitive radio network. However, these solutions only target on small-scale communication environments (i.e., at most 6 nodes [13]), which is far away from the realistic IoT system that consists of a large number of interconnected nodes. Besides, few has considered synergizing energy allocation and routing for EH IoT systems. In this paper, GAP is compared with ESDRARA [10], and Q-table, which represent energy-harvesting-aware routing, and energy-harvesting-aware routing with DRL, respectively.

III. SYSTEM MODELING

In this section, a comprehensive network (III-A) and energy (III-B), models will be given along with the problem formulation (III-C) for multi-hop EH IoT system.

A. Network Model

In this subsection, the detailed network model is formulated [14]. Each IoT device has four runtime operations, including sensing, transmission, receiving, and energy harvesting. Due to the large network scale, Sink node (C) is beyond the one-hop transmission range (C) of most IoT devices. In this case, aside from sensing and transmitting its own data, each IoT device also conducts packet relay (receive and then transmit) for its one-hop neighbors, as shown in Fig. 1.

Assume that there are M nodes in the network, where \(i_{th}\) node is labeled as \(i \in M\{1, 2, \cdots, M\}\). The neighbors of \(i\) are stored in set \(N_i\). While the node \(i\) sends data to \(j\) we can retrieve the transmission rate \(v_{i,j}\) in [13]. Further, given \(A_{i,j}\) as size of data transmitted from node \(i\) to \(j\). The transmission time \(\tau_{i,j,trans}\) is calculated by

\[
\tau_{i,j,trans} = \frac{A_{i,j}}{v_{i,j}}, \quad i \in M, j \in N_i.
\]

There is a FIFO queue in each node to buffer the transmission data. The maximum queue size is \(\Psi\) bits. Suppose that there are \(k\) data packets in queue \(\Psi\). The waiting time of the \(k_{th}\) task is as follows:

\[
\tau_{i,k,wait} = \sum_{l=1}^{k} \tau_{i,j,wait} + \sum_{l=1}^{k-1} \tau_{i,j,trans}, \quad i \in M, j \in N_i, k \in \Psi
\]

Therefore, the total time costs of \(k_{th}\) packet is \(\tau_{i,k} = \tau_{k,wait} + \tau_{k,trans} \in M, j \in N_i, k \in \Psi\).

To avoid the routing loop and guarantee the data delivery before getting expired, the data packet structure is organized as Fig. 2, where \(h_i\) is the packet’s current number of hops when it arrives at node \(i\) and \(\tau_{i,k}\) is the time cost at node \(i\). Given the maximum number of hops as \(H\) and the expiration time as \(T\), \(h_i\) and \(\tau_{i,k}\) can not exceed \(H\) and \(T\), respectively.

B. Energy Model

Energy consumption mainly comes from four activities, including sensing, transmission, receiving, and sleep. Thus, the total energy cost \(E_{i,con}\) for node \(i\) can be given by

\[
E_{i,con} = \int P_{i,trans} dt + \int P_{i,recv} dt + \int P_{i,sleep} dt + \int P_{i,sense} dt, \quad i \in M.
\]

Further, the residual energy of \(i\) is formulated as \(E_{i,res}\) =
\[ E_{i,\text{int}} + E_{i,\text{hav}} - E_{i,\text{con}}, \quad i \in \mathcal{M}, \] where \( E_{i,\text{int}} \) is the initial energy and \( E_{i,\text{hav}} \) is the harvested energy that is equal to the power integral over time. With the maximum energy storage capacity as \( E_{i,\text{max}} \), we have \( E_{i,\text{res}} \leq E_{i,\text{max}} \).

C. Problem Formulation

This paper aims to maximize the received data on Sink within the expiration time. Therefore, the optimization problem can be described as follows:

\[
\begin{aligned}
\text{maximize} & \quad \sum_{s,a} A_{i,j} \\
\text{subject to} & \quad 0 \leq E_{i,\text{res}} \leq E_{i,\text{max}}, h_i \leq H \quad \forall i \in \mathcal{M} \\
& \quad \sum_{i=1}^k A_{i,j} \leq \Psi \quad \forall i \in \mathcal{M}, j \in N_i \\
& \quad \tau_{i,k} \leq T \quad \forall i \in \mathcal{M}, k \in \text{queue},
\end{aligned}
\]

The constraints describe the EH IoT devices have a limited data storage and limited on-board power capacity.

IV. GAP ALGORITHM DESIGN

In this section, we first briefly review policy-based DRL in section IV-A. Based on the system models in section III we then design a scalable and distributed multi-agent DRL algorithm, known as GAP, to optimize the energy efficiency of the EH IoT system for communication, which is shown in subsection IV-B. After that, the detailed configuration of GAP will be discussed in subsection IV-C.

A. Reinforcement Learning

Reinforcement learning formulates the targeting environment as a Markov Decision Process (MDP), where the agent captures the environment state \( (s_t) \) at time step \( t \), then follows a policy \( (\pi) \) to take action \( (a_t) \). Based on the result of environment transition, an immediate reward \( (r_{t+1}) \) and the observable environment state \( (s_{t+1}) \) will be given to the agent at \( t+1 \). The long-term maximized reward \( (R_t^\pi) \) can be found at \( t \). It is formulated as \( R_t^\pi = \sum_{t=0}^{\infty} \gamma^t r_{t+1} \), where \( 0 < \gamma \leq 1 \) is a discounted factor with which the expected future reward can be formulated as a Q-function \( Q(s, a) = \mathbb{E}[R_t^\pi | s_t = s, a_t = a] \). In DRL, the policy is defined as \( \pi_\theta(a|s) \), where with the input environment state \( s \), the parameter model \( \theta \) will output the action \( a \) that has the maximum expected future reward. Specifically, by integrating deep neural networks (DNN), \( \theta \) is represented by the DNN inference model. Therefore, our goal, after training, is to obtain the optimal inference model \( \theta^* \) with the corresponding optimal action \( a = \text{argmax}_{a} Q(s, a)_{\theta^*} \).

To improve the training performance, actor-critic method is adopted. First, we use a new neural network model \( \Omega \) (critic) to estimate the Q-value \( \mathbb{E}[R_t^\pi | s_t = s] \) conditioned on the state \( s \) as V-function \( V_{\Omega, s_t} \). The loss function to update \( \Omega \) is:

\[
\mathcal{L}(\Omega) = \frac{1}{2B} \sum (R_t - V_{\Omega, s_t})^2
\]

Then, the advantage action function, \( Ad(s_t, a_t) = Q(s, a)_\pi - V^\pi(s_t) \), is adopted that indicate the advantage degree of action \( a \). The advantage value at time step \( t \) is denoted as \( Ad_t = R_t -
\]

\[ V_{\Omega, s_t} \cdot \] For the policy neural network \( \theta \), we aims to optimize the probability so that the action with higher \( Q \)-value will has a higher probability to be selected. Hence, the loss function for \( \theta \) can be formulated as in Eq. (3).

\[
\mathcal{L}(\theta) = -\frac{1}{B} \sum \log \pi_\theta(a|s_t) Ad_t
\]

B. GAP for EH IoT System

To build a scalable and distributed algorithm, instead of using the single-agent DRL, we employ MARL to control each IoT device at local. The intuitive method is that each \( i \) has a local agent which learns its own policy independently with the local experience information \((s_{i,t}, a_{i,t}, r_{i,t+1}, s_{i,t+1})\) at \( t \). Each local agent concurrently interacts with the EH IoT system environment and asynchronously optimizes its local policy model \( \theta_i \) as well as local V-function model \( \Omega_i \) (left side of Fig. 3). However, in multi-hop network the decision-making of local agents affect each others. To maximize the received data on Sink, local agents need to frequently exchange information and take cooperative actions which is unattainable due to the limited power budgets. Although the local agent can not obtain the actions of other agents after deploying, it can obtain the rewards of other agents in the training period so that the agents can consider the inference of the actions of other agents. Therefore, instead of providing a local reward \( r_{i,t+1} \) to the agent \( i \), we design to feed a spatial global reward \( s_{i,t+1} \) to the agent \( i \) which reflects the action influence to the whole system, as indicated in Fig. 3.

To save computation resources, instead of training multiple local agents, we train an actual universal agent by creating virtual agents at local. Local agents concurrently and asynchronously optimize their local virtual models, \( \theta_i \) and \( \Omega_i \). After that, they update the universal agent. Fig. 3 shows the GAP overview, and the algorithm is described in Algorithm 1.

C. GAP Settings

In the multi-hop EH IoT system, each IoT devices has to decide on its energy allocation and next routing neighbor independently and simultaneously based on its local environment information. Such an EH IoT system scenario can be formulated as POMDP and we design GAP settings as follows:

\textbf{State} \( s_i; \quad s_i \in \mathcal{M} \) is the local environment state of agent \( i \), which is denoted by \( s_i = \{n_{E_i}, n_{Q_i}, n_{I_i}\} \). \( n_{E_i} \) is the vector to describe the current energy of node \( i \) \( (E_{i,\text{res}}) \) and its neighbors. \( n_{Q_i} \) represents the queue size of node \( i \) and its...
Algorithm 1: GAP Training Procedure.

\begin{verbatim}
Input: max_episodes, γ, ι, Ω, n_nodes, s_time, e_time, max_step
Output: θ, Ω
1 Initialize Actor network, θ, with random weights;
2 Initialize Critic network, Ω, with random weights;
3 Reset EH IoT system environment;
4 n_episode ← 0
5 Create n_nodes threads (Threads ID i is from 0 to n_nodes - 1)
do in parallel
7 while n_episode < max_episodes do
8     Download θ, Ω to virtual local agent θ_i, Ω_i
9     Reset local environment and obtain initial local state s_i,0
10    i_step ← 0;
11   for s_time < real_time < e_time do
12       Calculate the action a_i,t on the environment
13       Take action a_i,t on the environment
14       Obtain r_i,t+1 and s_i,t+1; Update r_i,t+1 to all agents;
15       Calculate the spatial related reward sr_i,t+1
16       Collect experience (s_i,t, a_i,t, sr_i,t+1, s_i,t+1)
17       i_step ← i_step + 1
18   end
19   if i_step ≥ max_step then
20       Update θ_i with θ\theta_i \nabla L(θ_i) \[3\]
21       Update Ω_i with Ω_i \nabla L(Ω_i) \[2\]
22       Update θ_i and Ω_i to θ and Ω, respectively
23 end
24 n_episode ← n_episode + 1
end
\end{verbatim}

neighbors. nI_i is the source id of heading in queue, which is useful to prevent routing loop.

**Action a_i:** The action consists of two members, defined as a_i = \{E_i, R_i\}. E_i ∈ [0, 1] is to control energy allocation between sensing and transmission. If the current energy of device i(E_i, res) is less than E_i * E_i,max, the device i stop its transmission and only perform sensing and energy harvesting. While E_i, res > E_i * E_i,max, it restart transmission operation. The second term R_i ∈ N_i aims to select relaying destination node for the current outgoing data packet on node i.

**Reward r_i:** The reward of i’s agent at time step t, r_i,t, is designed as below:

\[
 r_i,t = \begin{cases} 
 \frac{\Lambda + 1/k + E_R}{k} & \text{else} \\
 0 & \text{if } I_i = i, h_i > H, \tau_i,k > T, E_i, res < 0 \sum_{l=1}^{k} A_{i,j} > \Delta 
\end{cases}
\]  

where two cases of reward are described. In the first case, the data packet is transmitted successfully. \Lambda is the one-step basic reward for agents to transmit the data packet out. k is the number of the data packet being stored in queue, which encourages agents to send the task to the device that does not have a heavier transmission burden. The \( E_R \) is a bonus reward if the agent transmits data to Sink node, given as (5).

\[
\begin{align*}
 E_R &= \begin{cases} 
 \log(\text{total number of IoT devices } \times \Lambda), & \text{if } j = 0 \\
 0, & \text{else} 
\end{cases} \\
&= \begin{cases} 
 \log(N_i \times \Lambda), & \text{if } j = 0 \\
 0, & \text{else} 
\end{cases}
\end{align*}
\]  

The second cases describe the device i fails to transmit data. Five common failures are considered. (1) Routing loop: The data packet gets stuck in a loop and is send back to its source node i. (2) Hop expiration: The data packet with a limited budget \( H \) on the number of routing hops. Without careful routing path design, the data packet is not delivered to Sink but gets stuck in relaying nodes that caused energy dissipation. (3) Time expiration: If the data packet stays in one node for a long time which might result in expired non-effectiveness data. (4) Receiver fails: Because of the transient nature of energy harvesting, the device at the destination might be offline. (5) Queue overflow: The storage of destination node j reaches its capacity Ψ. The packet will be refused by the destination but resent by the source node.

Eq. (1) encouraged Sink neighbors to transmit data to Sink obtaining a large reward. But for those nodes that are not Sink neighbors, they are only encouraged to transmit data to the device have low transmission burden(\( E_R = 0 \)). It is unclear if the transmission is useful for data delivery to Sink. Thus, a global spatial related reward \( sr_i,t \) is designed as \( sr_i,t = r_i,t + \sum_{\tau = t-1}^{\tau} \sum_{l:i,j,l \notin M} (1/d_{i,j} \times r_{j,t}) \), where \( d_{i,j} \) is the distance between i and j. \( sr_i,t \) represents a spatial reward accumulation of all agents from time \( a_{i,t-1} \) to time \( a_{i,t} \).

V. Evaluation

We developed a solar energy powered [15] IoT network simulator to evaluate our proposed GAP algorithm. Fig. 4 shows the topology of a 15-node example of multi-hop EH IoT system. The network consists of a single Sink (data transmission destination) and multiple EH IoT nodes.

A. Experimental Settings

**EH IoT network.** We simulate our multi-hop EH IoT network as Fig. 1 in 50 days. Each day is running from 8:00 to 17:00. For fairness comparison, at the beginning of each day, the environment will be reset. To ensure the sufficient start-up energy, \( E_i, res \) is initialized to energy storage capacity (1.J). TABLE I lists the parameters used in our simulation.

**GAP algorithm.** We train GAP for 50 episodes, which matches 50 days in our EH IoT network simulation. We set discount factor \( \gamma = 0.9 \) and the learning rate \( \epsilon_0 = 1e-4 \) and \( \epsilon_4 = 3e-4 \), for Actor \( \theta \) and Critic \( \Omega \), respectively. The architecture of the Actor (policy) neural network model and Critic (value) neural network model are 49-256-512-256-128 and 49-512-1024-256-1. Note that the single output of Critic is the estimated Q-value. Agent can select the energy threshold \( \epsilon_i \) from \[0, 0.3, 0.6, 0.9]\.

V. Evaluation

We developed a solar energy powered [15] IoT network simulator to evaluate our proposed GAP algorithm. Fig. 4 shows the topology of a 15-node example of multi-hop EH IoT system. The network consists of a single Sink (data transmission destination) and multiple EH IoT nodes.

**A. Experimental Settings**

**EH IoT network.** We simulate our multi-hop EH IoT network as Fig. 1 in 50 days. Each day is running from 8:00 to 17:00. For fairness comparison, at the beginning of each day, the environment will be reset. To ensure the sufficient start-up energy, \( E_i, res \) is initialized to energy storage capacity (1.J). TABLE I lists the parameters used in our simulation.

**GAP algorithm.** We train GAP for 50 episodes, which matches 50 days in our EH IoT network simulation. We set discount factor \( \gamma = 0.9 \) and the learning rate \( \epsilon_0 = 1e-4 \) and \( \epsilon_4 = 3e-4 \) for Actor \( \theta \) and Critic \( \Omega \), respectively. The architecture of the Actor (policy) neural network model and Critic (value) neural network model are 49-256-512-256-128 and 49-512-1024-256-1. Note that the single output of Critic is the estimated Q-value. Agent can select the energy threshold \( \epsilon_i \) from \( [0, 0.3, 0.6, 0.9] \). In case the time interval is too long while optimizing the neural network after each day ending, the neural network model is optimized for every 50 steps.

**Baseline.** We compare our GAP algorithm with three baselines. 1) GAPR Compared with GAP, GAPR makes decision
on routing selection only; 2) **Q-table** baseline will deploy a Q-table in each EH IoT devices. Since one centralized Q-table is impractical, we did not adopt it; 3) **ESDSRAA** baseline allocates energy budget and decides a constant sensing rate per hour. After energy allocating, it decides the relaying node based on the delivery rate and geo-information of neighbors.

**Measurement.** We evaluate the algorithms performance from two perspectives: **efficiency** and effectiveness. We adopt the total reward, the Sink received data and the delivery rate across all agents w.r.t. training episodes as our measurements on efficiency. Specifically, the delivery rate is defined as 
\[ \text{Sink received data size} \div \text{the total sensed data size} \]. The reward is measured as a natural scoring mechanism. We listed detail on the amount of sensed data, received data and transmitted data of several nodes and their corresponding energy cost to analyze effectiveness.

### B. Analysis and Discussion

1) **Efficiency:** As shown in Fig. 5, our GAP agent firmly and increasingly earns rewards in the first 5 training days. It surpasses the Q-table and GAPR significantly. During the training period, the neural network is optimized every 50 steps. Although it converges fast within 5 days, the neural network model goes through a large amount of data each day. Since ESDSRAA is the traditional method to calculate the energy allocation and routing selection strategy, it dose not have a reward parameter. The Q-table is convergent at 10th-day and obtained half reward of GAP agent. This result indicates the best learning efficiency of the proposed GAP. To evaluate the agent’s efficiency on data delivery, we further measure the total Sink node received data on each training day in Fig. 6. GAP agent achieved 14.18Mb data, where GAPR, Q-table, and ESDSRAA is at 7.29Mb, 11.10Mb, 11.39Mb, respectively. The proposed GAP improved 24.50% Sink node received data compared with the best baseline ESRSDAA.

If we only evaluate the network performance based on the received data on Sink, the EH IoT devices might “play a tricky”. They can generate a huge number of data but routing a few portions of generated data, which is not a fair comparison. Thus, to scientifically evaluate the performance, we measure the delivery rate. Fig. 7 indicated the delivery rate achieved by GAP (0.95) and ESDSRAA (0.93). Although they have a similar delivery rate, in Fig. 6 GAP has a higher Sink received data. After jointly considering the delivery rate and Sink received data amount, the result proves the proposed GAP agent can generate an outstanding routing policy. Q-table and GAPR complete almost half Sink received data and half rate of GAP. We will figure out the reason in Section V-B2. Moreover, the curve trend of reward in Fig. 5 is very similar with Fig. 6 and Fig. 7, which further proves the reward design is suitable in our case.

2) **Effectiveness:** From Fig. 8 to Fig. 13 we listed the data size and their energy cost details on sensing, transmission, and receiving on the 50th training day for three nodes. Without loss of generality, we select node-7, node-4, and node-9, labeled in Fig. 4 that presents three major kinds of nodes. Node-7 is located at the corner of the network, which rarely relays for others; Node-4 is located at the network center, which can directly communicate with Sink; Node-9 is a neighbor of node-4, which affects the decision-making of node-4.

Intuitively, due to its position, node-7 should not receive data. As shown in Fig. 6 GAP and ESDSRAA agents did not receive any data on node-7. However, for GAPR and Q-table, node-7 has an unanticipated receiving data, which means there is a routing loop or agent complete a wrong routing selection resulting in energy dissipation. Therefore it implies Q-table and GAPR should complete fewer data delivery tasks. It is exactly proved by Fig. 6 and Fig. 7 where both Sink received data and delivery rate is half that of GAP.

In Fig. 9 data size of sensing, transmitting, and receiving on node-4 are similar to node-7. However, node-4 is a central node that is close to Sink (one-hop). While the node located near Sink should have a heavier burden on relaying, it should relay data for the edge nodes. But node-4 did not receive data in GAP algorithm. For nodes near Sink such as node-4, even that they sense plenty of data, the sensed data can not be totally delivered to Sink. Because it has a limited battery capacity, if it still costs much energy on sense, there is insufficient energy for transmission. However, Fig. 6 indicated GAP delivered the most data to Sink. The reason is that relaying task is completed by other nodes which are also near Sink. To further validate it, we also listed the details on node-9 in Fig. 10. It indicated the transmission data is dramatically higher than the sensing data on node-9 for GAP agent. While nodes have heavier transmission/relaying task that is related to neighbor numbers and nodes positions, sensing operation is anticipated to be reduced so that nodes can save more energy for transmission.

If the relaying missions are not heavy, the node should sense data as much as possible for sensing quality(node-9 of GAP). However, the overall distribution on sensing, receiving and transmitting of GAPR, Q-table, ESDSRAA in Fig. 8 Fig. 10 is generally similar. But, the ratio of sensing and transmission for GAP is varied with different nodes. The result of GAP in Fig. 8 Fig. 10 implies its ability to dynamically adapt the sensing and transmission ratio for the system’s global optimal goal with the consideration of the nodes local information.

Received data plus sensed data should be equal to transmitted data. Otherwise, the packet drop occurs. In Fig. 8 Fig. 10 this is completed by GAP and ESDSRAA. GAPR and Q-table did not achieve it because of packet drop caused...
by routing loop or packet cost, which implies wrong routing selection. GAPR only makes decisions on routing. Better routing selection is expected. However, its routing performance is degraded. It further illustrates jointly considering energy allocation and routing selection is critical.

In Fig. 8 the transmission data size of GAPR is less than the transmission data size of GAP. However, in Fig. 11 the energy consumption for transmission of GAPR is more than that of GAP, which means re-transmission happened. That is another manner that caused the energy dissipation. A similar situation also happens in Fig. 13 while we compare transmission energy cost of Q-table and GAPR. Moreover, the total energy cost in the other three algorithms is much more than the energy cost in GAP on each node, which means they have more opportunities for harvesting energy. More Sink received data should be achieved. But Fig. 6 reflects they did not.

VI. CONCLUSION

This paper constructs a comprehensive multi-hop data transmission model for EH distributed IoT networks. With the objective of maximizing the amount of Sink received data, we proposed GAP establishing a distributed MARL for EH IoT networks to jointly optimize the routing policy and energy allocation with the consideration of global long-term optimality and devices entanglement. The spatial global reward is integrated to MARL to address the entangled problem in the multi-hop EH IoT system. The experimental results indicated the outstanding performance of the proposed GAP compared with the three baselines.

REFERENCES

[1] “The future of iot miniguide: The burgeoning iot market continues.” [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html

[2] C. Pan et al., “Enzyme: An energy-efficient transient computing paradigm for ultralow self-powered iot edge devices,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2440–2450, 2018.

[3] ——, “Modeling and optimization for self-powered non-volatile iot edge devices with ultra-low harvesting power,” ACM Transactions on Cyber-Physical Systems, vol. 3, no. 3, pp. 1–26, 2019.

[4] K.G. Shin et al., “A simple distributed loop-free routing strategy for computer communication networks,” IEEE transactions on parallel and distributed systems, vol. 4, no. 12, pp. 1308–1319, 1993.

[5] V. Rao et al., “Renew: A practical module for reliable routing in networks of energy-harvesting wireless sensors,” in 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019, pp. 1–6.

[6] T. Wang et al., “A multi-featured actor-critic relay selection scheme for large-scale energy harvesting wsns,” IEEE Wireless Communications Letters, 2020.

[7] V. Mnih et al., “Human-level control through deep reinforcement learning,” nature, vol. 518, no. 7540, pp. 529–533, 2015.

[8] T.D. Nguyen et al., “A distributed energy-harvesting-aware routing algorithm for heterogeneous iot networks,” IEEE Transactions on Green Communications and Networking, vol. 2, no. 4, pp. 1115–1127, 2018.

[9] A. Banerjee et al., “Joint power allocation and route selection for outage minimization in multihop cognitive radio networks with energy harvesting,” IEEE Transactions on Cognitive Communications and Networking, vol. 4, no. 1, pp. 82–92, 2017.

[10] T. Lu et al., “Energy-efficient data sensing and routing in unreliable energy-harvesting wireless sensor network,” Wireless Networks, vol. 24, no. 2, pp. 611–625, 2018.

[11] S. Tang et al., “Reward rate maximization and optimal transmission policy of eh device with temporal death in eh-wsns,” IEEE Transactions on Wireless Communications, vol. 16, no. 2, pp. 1157–1167, 2016.

[12] A. Ortiz et al., “Reinforcement learning for energy harvesting decode-and-forward two-hop communications,” IEEE Transactions on Green communications and networking, vol. 1, no. 3, pp. 309–319, 2017.

[13] X. He et al., “Routing selection with reinforcement learning for energy harvesting multi-hop crn,” IEEE Access, vol. 7, pp. 54 435–54 448, 2019.

[14] W. Zhang et al., “Sac: A novel multi-hop routing policy in hybrid distributed iot system based on multi-agent reinforcement learning,” in 2021 22nd International Symposium on Quality Electronic Design (ISQED). IEEE, 2021, pp. 129–134.

[15] “Measurement and instrumentation data center (midec).” [Online]. Available: https://midcdmz.nrel.gov/apps/sitehome.pl?site=ORNL