M2M-Routing: Environmental Adaptive Multi-agent Reinforcement Learning based Multi-hop Routing Policy for Self-Powered IoT Systems

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

Jeff Zhang
Department of Electrical Engineering
Harvard University
Cambridge, USA
jeffzhang@seas.harvard.edu

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

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

Wenlu Wang
Department of Computer Science
Texas A&M University–Corpus Christi
Corpus Christi, USA
wenlu.wang@tamucc.edu

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

***Abstract*—Energy harvesting (EH) technologies facilitate the trending proliferation of IoT devices with sustainable power supplies. However, the intrinsic weak and unstable nature of EH results in frequent and unpredictable power interruptions in EH IoT devices, which further causes unpleasant packet loss or reconnection failures in IoT network. Therefore, conventional routing and energy allocation methods are inefficient in the EH environments. The complexity of the EH environment caused a stumbling block to an intelligent routing policy and energy allocation. To address the problems, this work proposes an environment adaptive Deep Reinforcement Learning (DRL)-based multi-hop routing policy, *M2M-Routing*, to jointly optimize energy allocation and routing policy and mitigate these challenges through leveraging the offline computation resources. We prepare multi-models for the complex energy harvesting environment offline. By searching a historically similar power trace to identify the model ID, the prepared DRL model is selected to manage energy allocation and routing policy on the query power traces. Simulation results indicate that *M2M-Routing* improves the amount of data delivery by $\sim 3\times$ to $\sim 4\times$ compared with baselines.**

I. Introduction

Energy harvesting (EH) technologies that harvest energy from the ambient environment have been increasingly employed as an alternative to batteries on IoT devices due to the low maintenance cost and wide availability [1]. There are a variety of energy sources available such as kinetics and thermal conduction. However, the intrinsic weak and unstable nature of EH results in frequent and unpredictable power failures in EH IoT devices [2], [3]. In the IoT network, frequent power failures lead to frequent unpleasant packet loss or continuous reconnection failures, rending data routing unattainable. Therefore, conventional routing methods are inefficient in the EH environments because of the severe degradation of communication efficiency caused by power failures of EH devices.

An intelligent routing strategy is on urgent demand for EH IoT devices, which is however challenging. The main challenge lies in energy allocation. If the EH IoT device collects too much data, it may not have sufficient energy to transmit the collected data when required. Conversely, if the EH IoT device reserves sufficient energy for communication, it may squander harvested energy for nothing. Furthermore, it requires multi-hops among EH IoT devices to reach the sink, and thus those devices are spatially dependent on each other. Existing optimal energy allocations that maximize the data transmission at the current time step might result in insufficient energy for future data transmission, and thus IoT devices are also temporally dependent. Therefore, intelligent routing strategies require joint optimization of energy allocation on data sensing and data transmission with a multi-hop network routing policy.

The spatial and temporal dependency of the partial observable multi-hop routing network environment makes Deep Reinforcement Learning (DRL) [4] an attractive tool. By formulating energy allocation and routing selection as partially observable Markov Decision Processes, we can design a distributed and highly scalable DRL-based approach to maximize the energy efficiency and the packet delivery for sensing and transmitting data in the EH IoT multi-hop routing systems. However, due to the complexity of the EH environment, training a single DRL model will incur significant time overhead and may not achieve a desirable outcome. Because of the nature of energy harvesting, the energy status of an IoT device is firmly associated with its power trace that varies in real-time. For this reason, a *large* DRL model is required to adapt to different power trace variations. However, EH IoT devices are usually compute and memory resource constraint.

To improve the environmental adaptability of DRL agents, instead of training one single model to learn over all power

* Corresponding Author

978-3-9819263-6-1/DATE22/©2022 EDAA

traces, we propose to have multiple representative models trained based on the similarity between power traces. In addition, to alleviate the computation burden of EH IoT devices, our idea is to move most expensive computations offline and only perform limited computations to execute DRL agents online. In the offline phase, we first cluster the historical power traces into K sub-databases. Then we train K representative DRL agents on the corresponding sub-database and deploy into each EH IoT device. To achieve this, we designed a temporal distance-based similarity search algorithm to precisely retrieve the most similar power trace in our sub-databases and its coupled DRL agent ID. The selected DRL agent ID is send to EH IoT devices and manages the next energy allocation and routing policy joint optimization to maximize the amount of data delivery to the sink. The main contributions of this paper are as follows:

(1) We proposed a multi-model RL based routing framework, *M2M-Routing*, to improve the environmental adaptability of EH IoT multi-hop routing by leveraging offline computation capability and power trace prediction techniques.

(2) To precisely retrieve the most similar power trace in sub-databases, we developed an efficient temporal distance-based similarity search algorithm.

(3) We performed comprehensive simulations to evaluate the performance of *M2M-Routing* from the training, querying, testing, and overhead perspectives.

II. Related Work

Traditional Energy-harvesting-aware Routing: Nguyen et al. [5] proposed an energy-harvesting-aware 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. [6] developed ESDSRAA algorithm to explore multi-hop routing for EH IoT systems with energy-harvesting-aware geographic routing and different energy allocation strategies. However, traditional Energy-harvesting-aware Routing without consideration of the uncertainty of the power source is not designed for long-term optimization, resulting in throttled system performance. **Energy-harvesting-aware Routing with DRL:** DRL shows outstanding performance on decision making in an uncertain environment considering the long-term influence of its actions [7]–[9]. [8] employed an algorithm to optimize the power allocation for two-hop EH communications. Q-table is created in [9] to find the optimal routing path for EH multi-hop cognitive radio network. However, these solutions only target small-scale communication environments with centralized RL (i.e., at most 6 nodes [9]), being hard to extend to the realistic IoT system setting that consists of a large number of interconnected nodes. Besides, few works consider how to synergistically perform energy allocation and routing for EH IoT systems. This paper proposes an environment adaptive DRL-based multi-hop routing policy that jointly optimizes energy allocation and routing policy. In the experiment, ESDSRAA [6] and Q-table are taken as baselines for comparison.

III. System Model

This section presents a multi-hop routing system model for EH IoT devices, then describes the energy harvesting system model, and finally formulates the joint optimization problem.

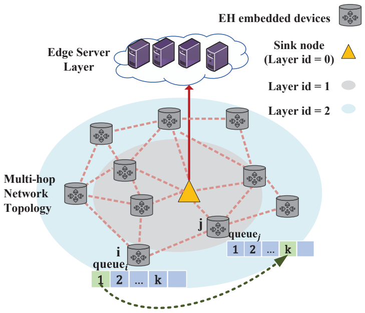

Fig. 1: Multi-hop routing system model for EH IoT devices.

A. Multi-hop routing system model for EH IoT devices

The multi-hop routing system model considers a typical data transmission scenario, where a set of EH IoT devices is deployed in a designed area such as an ocean or forest. Each device can execute four operations including sensing, data transmission/forwarding, receiving, and harvesting energy simultaneously. The set of EH IoT devices is described as $\mathcal{I} = \{1, \dots, i, \dots I\}$. If the distance between device i and device j is within ζ, they are defined as neighbors that can transmit or receive data packets in one hop. The neighbors of device i are defined as a set N_i. EH IoT device aims to transmit their sensed data to sink $\mathcal{D}$. In order to lead devices to transmit data towards the sink, the designed space is divided into multiple layers with the centre sink $\mathcal{D}$. Based on the device location, each device i has a layer id attribute, defined as ρ_i. The multi-hop routing system model detail is illustrated as Fig. 1.

The EH IoT device follows a predetermine sensing speed v to collect data each day. The device keeps sensing if the device has residual energy. Otherwise, the device goes to sleep mode. Suppose that there is a $\textbf{queue}_i$ in device i to buffer the sensed or received data packets. The queue size is denoted as $\mathcal{Q}$. Specifically, a preset threshold Ψ is given to prevent queue overflow. While the number of data packets in the queue is greater than Ψ, sensing is turned off and the left queue size is only used for relaying data. Ψ ($\Psi \leq \mathcal{Q}$) can be obtained by profiling the network traffic. While the number of data packets is beyond $\mathcal{Q}$, the device stop data sensing and receiving.

Each data packet is transmitted with its source device ID D_i. Once the relay device ID j is equal to D_i, there is a routing loop, which causes the energy dissipation. To avoid data expiration, each data packet $A_{i,j}$ transmitted from device i to device j, $j \in N_i$ has an attribute *experienced hops* h to label its experienced hops. Given the maximum experienced hops Γ, h should not exceed Γ. The data packet resulting in routing loop or expired data packets both will be dropped.

B. Energy harvesting model for EH IoT devices

Each EH IoT device harvests energy from ambient sources and consumes energy on sensing, transmission/forwarding, and receiving. The energy consumption $E_{i,consume}$ is equal to the power integration for time. The power of sensing, transmission, and reception are $p_{i,sense}$, $p_{i,trans}$, and $p_{i,recev}$, respectively. Given a time duration T, we separate time into a set of time slots $\mathcal{T} = \{1, \dots, t, \dots, T\}$. The power trace P_{id} on id^{th} day is given as $P_{id} = \{p_{(1,id)}, \ldots, p_{(t,id)}, \ldots p_{(T,id)}\}, \quad t \in \mathcal{T}$, where $p_{(t,id)}$ is the power intensity at time t on id^{th} day. Thus, the harvesting energy is $E_{i,hav} = \int P_{id}\, dt$. Assume the real-time energy status of device i is $E_{i,current}$, given the energy capacity of the device i as $E_{i,max}$, at any given time, $E_{i,current} \leq E_{i,max}$ should be satisfied.

C. Problem formulation

Unlike classical network studies that target real-time communication, in this paper, we study EH IoT devices for data collection applications that are not time-sensitive but are more power-hungry. Thus, our goal is to maximize the amount of data delivery to sink $\mathcal{D}$ while EH devices are powered by an uncertain power trace.

Formally, given a set of deployed EH IoT devices, the onboard battery capacity $E_{i,max}$, and the queue size $\mathcal{Q}$, the energy allocation and routing policy joint optimization problem is formulated as follows:

$$\begin{aligned}
&\underset{i \in N_j, j = \mathcal{D}}{\text{maximize}} && \sum \mathrm{A_{i,j}} && \\
&\text{subject to} && 0 \leq E_{i,current} \leq E_{i,max}, && \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \\
& && \sum_{l=1}^{k} A^{l}_{i,j} \leq \mathcal{Q}, && \forall i \in \mathcal{I}, j \in N_i, \forall t \in \mathcal{T} \\
& && h \leq \Gamma && \forall A_{i,j}, \forall i \in \mathcal{I}, j \in N_i \\
& && j \neq D_i && \forall A_{i,j}, \forall i \in \mathcal{I}, j \in N_i
\end{aligned}$$

IV. M2M-Routing Framework Design

A. *Overview of* M2M-Routing

To address the challenge of the limited computation capability and power-hungry situation of EH IoT devices, *M2M-Routing* is developed that thoroughly take merits of offline sufficient computation and energy resource to fully realize environment adaptive intelligent energy allocation and routing policy. An overview of *M2M-Routing* is indicated in Fig. 2, which is composed of *offline segment* and *online segment*.

The *offline segment* trains the distributed multi-agents DRL with historical power traces. We first cluster the historical power trace into K databases with meanshift clustering [10]. After that, DRL agents $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ are trained for each EH IoT device with the corresponding database. At the outset, the power trace of next work round are predicted with [11]. The query subsystem retrieves the most similar power trace in k databases by the developed temporal distance-based similarity search algorithm. The corresponding DRL agent ID k is informed to the device. The *online segment* receives DRL agent ID and makes that agent take over energy allocation and routing policy at the next round. Note that the real power trace P_{id} can never be known in advance.

B. Multi-agent deep reinforcement learning agent training

Reinforcement learning formulates the decision-making problem as Markov Decision Process (MDP), named *environment*. The agent observes the environment state $s_\tau, s_\tau \in S$ at time step τ. Based on the policy that maps state S to action A, the agent takes an action $a_\tau, a_\tau \in A$ to the environment. Environment transits to $s_{\tau+1}$ and return a feedback $r_{\tau+1}$ to the agent. In order to evaluate agent's performance from long-term perspective, the discounted reward is defined as $R^{\pi}_{\tau} = \sum_{\tau'=\tau}^{\infty} \gamma^{\tau-\tau'} r_{\tau'+1}$, where $\gamma, \gamma \in (0, 1]$ is the discounted factor for the future reward. Reinforcement learning agent aims to find an optimal policy that maximizes the expected long-term reward q-value $Q(s, a)^{\pi} = \mathbb{E}[R^{\pi}_{\tau} | s_\tau = s, a_\tau = a]$. Deep-Q-network trains a neural network as the policy to estimate q-values. This paper adopts deep-Q-network (DQN) [4] as the underlying method to find an optimal policy on energy allocation and routing.

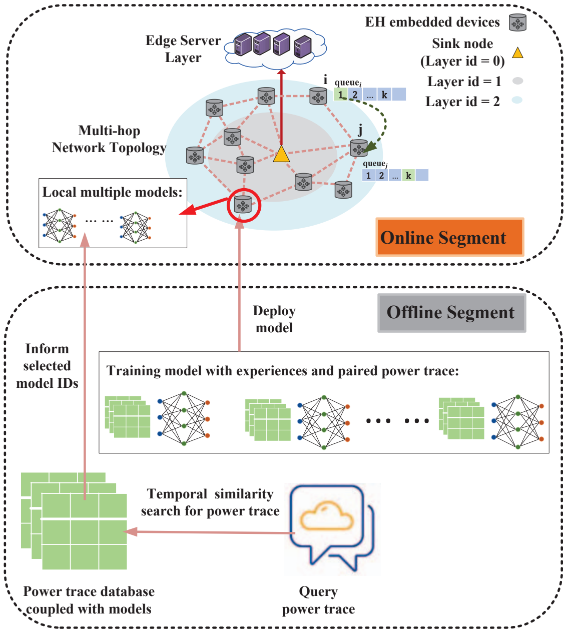

Fig. 2: Overview of *M2M-Routing* for self-powered IoT system.

We deploy a unique DRL agent named *DeepIoTRouting* for each IoT device to control its energy allocation on data sensing and data transmission as well as its packets routing destination. All DRL agents have the same intermediate layer size of neural networks but different weights. Since the environment state of each node is only related to information of its neighbors, the input/output layers are different, which increases the algorithm scalability and reduces the state and action spaces.

Training: At the training period, each device has two neural networks, target network and local network parameterized by θ_i and θ'_i. Local network outputs a_i to environment, then collects the interaction $experience = (s_{i,\tau}, a_{i,\tau}, r_{i,\tau+1}, s_{i,\tau+1})$ into replay buffer G_i. The target network output "label" for the local network. DRL agent is optimized until a mini-batch of *experiences* sampled from replay buffer. To stabilize training, the target network will be frozen its weights. After each n interactions, the target weights will be updated by copying weights from local network. Thus, the loss function of the local network is given as $L(\theta_i) = \sum_{s_{i,\tau}, a_{i,\tau}, r_{i,\tau+1}, s_{i,\tau+1} \in G_{i,\tau}} (r_{i,\tau+1} + \gamma \max_a Q(s_{i,\tau+1}, a_{i,\tau+1}; \theta') - Q(s_{i,\tau}, a_{i,\tau}; \theta))^2$. To avoid local minimum and learn from environment, DRL agents are expected to take more actions on exploration at an early age. After familiar with environment, agents should take actions

Algorithm 1: Training procedure for DRL based agent.

```
Input: $max\_episodes, start\_time, end\_time, n, n\_node,$
$\{\gamma, \epsilon_{start}, \epsilon_{end}, \delta\}$ (for learning and exploration)
Output: optimized $n\_node$ agents $\theta_i$
Initialize $n\_node$ local networks, $\theta_i$, with random weights, $\forall i$;
Initialize $n\_node$ target networks, $\theta'_i \leftarrow \theta_i$, $\forall i$;
Initialize the corresponding replay buffer, $G_i$, $\forall i$;
$n\_episode \leftarrow 0$
while $n\_episode < max\_episodes$ do
    Reset EH IoT system environment;
    $Step_i \leftarrow 0 \quad \forall i$;
    while $time < end\_time$ do
        for $i$ from 0 to $n\_node$ do
            if $i$ request action from agent $\theta_i$ then
                With probability $\epsilon$ select random action $a_{i,t}$
                Otherwise, action $a_{i,t} = \mathrm{argmax}\, Q(s,a)$ by $\theta_i$
                Take action $a_{i,t}$ to environment
                Environment transit to $s_{i,t+1}$; Feedback $r_{i,t+1}$
                $experience_i = (s_{i,t}, a_{i,t}, r_{i,t+1}, s_{i,t+1})$;
                Save experience to buffer $Z_i$;
                if $replaybuffer \geq mini\_batchsize$ then
                    Randomly sample mini-batch of experiences;
                    Optimize $\theta_i$ with sampled experiences;
                    if $Step_i \% m == 0$ then
                        Update target network $\theta'_i$;
                    end
                end
                $Step_i = Step_i + 1$
            end
        end
    end
    $n\_episode = n\_episode + 1$
end
```

based on their experience. Thus, this paper adopts the decaying epsilon-greedy exploration policy [12]. The algorithm detail is described in Algorithm 1.

MDP settings: Each device observes its partial observable environment state s_i that is defined as $\{\Delta_i, \mathcal{E}_i, D'_i\}$. Δ_i is a vector to provide queue size of device i and its neighbors. $\mathcal{E}_i$ is a energy vector to provide the residual energy of device i and its neighbors. Recall that D_i is the data packet source ID to prevent routing loop. D'_i is the source ID of the packet located at $\mathbf{queue}_i$ head. To make decision on routing destination and energy allocation, a_i is designed as $\{\mathcal{N}_i, \Re_i\}$. $\mathcal{N}_i, \mathcal{N}_i \in N_i$ is the routing destination of current ongoing data packet of device i. $\Re_i$ aims to control the real-time energy allocation on data transmission and data sensing. We quantilize the battery capacity ($E_{i,max}$) in multiple levels with $\Re_i$. While the residual energy $E_{i,current}$ is less than $\Re_i * E_{i,max}$, the transmission/forwarding/reception is stopped. The EH IoT device only performs low-power sensing and energy harvesting. Once $E_{i,cuurent} \geq \Re_i * E_{i,max}$, the EH IoT device restarts the stopped actions. By optimizing $\{\mathcal{N}_i, \Re_i\}$, EH IoT devices aim to maximize the amount of data delivery. Thus, our reward function is designed as (1).

$$r_{i,\tau} = \begin{cases} log(1 + \rho_i - \rho_j) + f(j), \text{ else} \\ \\ 0, \text{ if } \mathcal{D}'_i = j, h > \Gamma, E_{j,current} < 0, \sum_{l=1}^{k} A^l_{i,j} > \mathcal{Q} \end{cases} \tag{1}$$

where two cases are described. First case describes the destination device successfully receives the transmitted data packet. Recall that ρ_i is the layer id of device i which aims to lead device i transmitting data packets towards the sink. $f(j)$ is a function of extra reward while the data packet arrives sink directly, where $f(j) = log(I)$, if $j = \mathcal{D}$, and goes to 0 otherwise. The second case of (1) describes the transmission failure. Four kinds of classical transmission failures are considered: (1) While the data packet is routed to its source device, the routing loop is caused; (2) To guarantee the energy efficiency and data freshness, if the experienced hops h exceeds the budget hop Γ, the packet is dropped; (3) Due to the unstable nature power, the receiver device might be offline, which results in retransmission and deplete energy efficiency; (4) While $\mathbf{queue}_j$ is full at the destination device, the data packet is refused and required retransmission. It also results in energy dissipation.

C. Query subsystem

The query subsystem is composed of *power trace prediction* and *temporal distance-based similarity search algorithm.*

Power trace prediction: We adopt *Pro-energy* algorithm developed by [11] to predict power trace for the forthcoming data transmission. The main idea of *Pro-energy* is to leverage the historical energy observations to estimate future energy. Given a number of representative days such as sunny and cloudy, the historical power trace P_{id} is stored into its representative day's profiles. This paper has K representative days. We estimate the predicted power at time slots $t+t'$ by current power $p_{(t,id)}$ and the historical power at $t+t'$ on a set of similar days. Therefore, the predicted power is $p'_{(t+t',id)} = \eta_{t'} p_{t,id} + (1 - \eta_{t'})\mathcal{P}_{t+t'}$, where $\eta_{t'}$ is Pearson autocorrelation coefficient parameter. With the time slots of predicted power is gradually away from current time slots, the correlation progressively decreases. About the value of $\eta_{t'}$ please see [11]. $\mathcal{P}_{t+t'}$ is a weighted value of most similar historical power at time slot $t + t'$. Following paper [11], we first retrieve the top m similar traces with smaller mean absolute error. That is calculated by $d_{(P_{id},P_{id'})} = \sum_{l=t-t'}^{t} \frac{|p(l,id) - p(l,id')|}{H}$. Then, we calculate $\mathcal{P}_{t+t'}$ that is the weighted average power intensity at $t + t'$ on m days as (2).

$$\mathcal{P}_{t+t'} = \frac{1}{m-1} \sum_{id'=0}^{m} \left[1 - \frac{d_{(P_{id},P_{id'})}}{\sum_{id'=1}^{m} d_{(P_{id},P_{id'})}}\right] \cdot p_{(t+t',id')} \tag{2}$$

Temporal distance-based similarity search: To precisely retrieve similar power traces to identify the model ID, an appropriate distance measurement between power traces is needed. However, regarding the power trace P_{id} as a vector and calculating the vector distance to measure similarity that only can capture the power strength but lose the power trace temporal features, since the vector distance accumulates the distance at each time slot. Thus, we developed an algorithm to measure the spatiotemporal distance of power traces. The power trace similarity measure is realized by a *dynamic moving window*[13] scanning that captures the distance of two power trace with a varying size window. Considering two power traces P_{id} and $P_{id'}$, we visualize them as Fig. 3 (a) and Fig. 3 (b), respectively. A distance ($\Lambda = d_{P_{id},P_{id'},q}$) is measured between P_{id} and $P_{id'}$ by scanning the transited power trace using a *dynamic moving window* ϕ with size q, which can be calculated as (3).

$$\Lambda = \sum_{\phi_{\varphi,q} \in \Phi_q} \Big(\sum_{p_{t,id} \in \phi_{\varphi,q}} \lambda_{t,id} p_{t,id} - \sum_{p_{t,id'} \in \phi_{\varphi,q}} \lambda_{t,id} p_{t,id'} \Big) \tag{3}$$

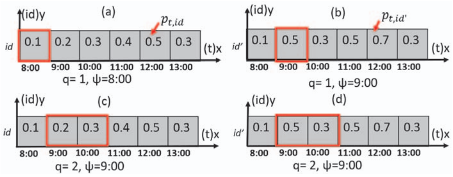

Fig. 3: Spatiotemporal distance measurement.

where $\phi_{\varphi,q}$ represents the window with size q is located at φ; φ is identified by t; Φ_q is the full set for the dynamic window of size q; $\lambda_{t,id}$ is the contribution factors that means the contribution of unit $p_{t,id}$ in window $\phi_{\varphi,q}$. Given the max size of dynamic window q_{max}, the total distance of P_{id} and $P_{id'}$ is the summation of all distance obtained by all sizes moving window, $D_{P_{id},P_{id'}} = \sum_{q=1}^{q_{max}} d_{P_{id},P_{id'},q}$. Through the total distance measurement, query subsystem retrieves the corresponding DRL ID whose training dataset has the most similar power trace with the predicted power trace.

V. Evaluation

A. Experimental Settings

Multi-hop routing system: We simulate a 20-node multi-hop routing system similar with Fig. 1. We deploy 20-node in a $100m \times 100m$ open space as Fig. 4. The hardware parameters of EH IoT devices are listed in Table I. The multi-hop routing system starts running from 8:00 to 18:00 every day. EH IoT device starts with $E_{i,max}$ to ensure sufficient start-up energy.

TABLE I: EH IoT Network Parameters

Notation	Definition	Parameter Value
$A_{i,j}$	size of data packet	3072 bits
$\mathcal{Q}$	maximum queue size	$20 * 3072$ bits
$P_{i,trans}$	transmission power	0.1 w
$P_{i,recev}$	receiving power	0.05 w
$P_{i,sense}$	sensing power	0.01 w
$E_{i,max}$	energy storage capacity	1 J
v	process speed	200 bits/s
ξ	transmission range	20 m
layer number	layer numbers	3
$\Re_i$	the energy threshold	$\mathcal{T}_i \in \{0, 0.3, 0.6\}$
Γ	Initial hops budget	6

M2M-Routing algorithm: We use the real solar power trace downloaded from [14]. We cluster 40 power traces($id \in [1, 40]$) into 2 sub-databases (K=2). We train *M2M-Routing* for 120 episodes, where each episode means the one-day operation of multi-hop network. To prevent the network system influenced by the previous day, the network environment will be initialized at the beginning of each day. Two neural network models of *M2M-Routing* have same architectures, the input/output layer is determined by the definition of state/action in Section IV. Their immediate layer is set as 128-256-128. The discount factor γ is 0.95. We set learning rate $\delta = 5e^{-4}$.

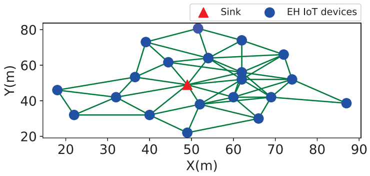

Fig. 4: Topology of 20-node multi-hop routing system.

Baselines: We compare our algorithm with three baselines.1) *DeepIoTRouting*: single DRL model, where one DRL model is deployed on each EH IoT device. 2) Q-table: one Q-table is deployed on each IoT device. 3) ESDSRAA: a traditional energy-harvesting aware routing algorithm, which allocates energy and sensing rate at the beginning of each hour, based on which, it decides the routing destination.

B. Performance Analysis on DRL Agents

Training: We first measure the *M2M-Routing* performance during the training period. As indicated in Fig. 5, we use the total rewards as a raw score to measure *M2M-Routing*'s learning performance. Three reinforcement learning methods converge quickly at about 15^{th} day. *M2M-Routing* and *DeepIoTRouting* gains $\sim 3\times$ rewards over Q_table. Although *DeepIoTRouting* and *M2M-Routing* obtain similar rewards, the obtained reward of *M2M-Routing* maintains at a stable level but *DeepIoTRouting* is fluctuating regularly. That is because we run a multi-hop system with 40 power traces. While the power trace has greater strength, *DeepIoTRouting* obtains more rewards. In Fig. 5, *DeepIoTRouting* reaches the peak point at every 40 days, which further proves that *DeepIoTRouting* fluctuates with the power trace variance. Clustering similar power traces to the same sub-databases not only guarantees a better-optimized performance but also stabilizes the DRL agent.

To evaluate the efficiency of *M2M-Routing* in multi-hop routing network, Fig. 6 shows the daily amount of data delivery. We observe that Fig. 6 is generally coincident with Fig. 5, which proves the designed reinforcement learning agent reward is appropriate for our multi-hop routing system. At training period, *M2M-Routing* achieves around the maximum data delivery at 71.8Mb for model $k = 1$ (model1) and the maximum data delivery at 59.7Mb for model $k = 2$ (model2). Although the maximum data delivery amount by *DeepIoTRouting* (72.1Mb) slightly exceeds the amount of data delivery by *M2M-Routing*, it is intuitive that after agents arriving convergent, the average amount of data delivery of *M2M-Routing* is greater than that achieved by *DeepIoTRouting*.

Query: After training DRL agents with 40 traces, we test *M2M-Routing* with 20 power traces. *Pro-energy* predicts one power trace as shown in Fig. 7 (purple) for the red real trace. The error between the predicted power trace and the real trace is 4.42%. Through predicted power trace, *temporal distance-based similarity search algorithm* finds the blue *similar trace1*. The vector distance-based similarity search finds the green *similar trace2*. The corresponding DRL agent for *similar trace1* is model1 and the corresponding agent for *similar trace2* is model2. Therefore, we tested the red real trace with both model1 and model2. Model1 and model2 achieved 61.2Mb and 58.9Mb amount of data delivery, respectively. Based on Fig. 6, model2 always outperforms model1. However, it achieves less amount of data delivery than model1.

There are two possible reasons. First, the *Pro-energy* is not enough precise so that the similarity search can not find the right similar trace. Second, the power trace prediction has no influence, which proves that the *temporal distance-based similarity search algorithm* can find a more similar trace and

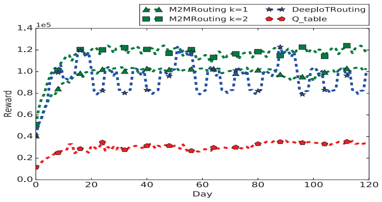

Fig. 5: Daily reward.

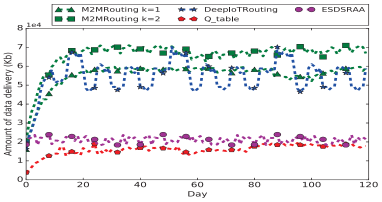

Fig. 6: Daily amount of data delivery.

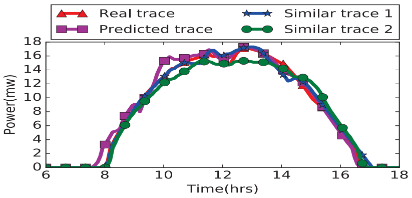

Fig. 7: Solar energy trace sample.

TABLE II: The efficiency evaluation (daily average).

Algorithm	Delivery rate (%)	Delivery amount (Mb)
M2M-Routing	96.48	62.73
DeepIoTRouting	95.98	54.25
Q_Table	41.31	16.68
ESDSRAA	56.56	20.90

choose an appropriate model since model1 achieves a higher amount of data delivery. The visualization of power traces in Fig. 7 indicates the trend of *similar trace1* better fits to the red real trace than the *similar trace2*, though the vector distance of *similar trace2* and the real trace is smaller. We will further discuss the influence of power trace prediction in our future work. However, no matter what the influence of power trace precision is, the *M2M-Routing* achieves an outstanding performance, which is proved by Table II.

Testing: Table II indicates the average amount of daily data delivery and *delivery rate* $= \frac{\textit{the amount of data delivered to sink}}{\textit{the total sensed data size}}$, where system operates with 20 power traces. Even that prediction error arrives 5.26%, *M2M-Routing* achieves 96.8% *delivery rate*. The amount of data delivery of *M2M-Routing* improves 15.63% compared with *DeepIoTRouting*. While compared with Q_table and ESDSRAA, *M2M-Routing* achieves $\sim 4\times$ and $\sim 3\times$ the amount of data delivery. Whether a higher prediction accuracy results in the performance gain of *M2M-Routing*, we will explore it in our future work. So far, *M2M-Routing* achieves the best performance compared with the three baselines.

Overhead: We also discuss the online computation overhead in Table III with a CPU frequency of 16MHz. Since *DeepIoTRouting* achieves 0.84 *delivery rate* while its neural network architecture is the same to *M2M-Routing*, we increase its neural network size to 256-512-128 such that it can achieve a higher delivery rate of 95.98%. However, even after the neural network size is increased, the amount of data delivery and delivery rate of *M2M-Routing* is still higher than that of *DeepIoTRouting*, as shown in Table II. Therefore, *M2M-Routing* does not only achieves a better performance but also has a lower computation cost and latency (0.019s). Although it has a slightly higher online computation cost than that of Q_table and ESDSRAA, the performance of *M2M-Routing* dramatically exceeds that of Q_table and ESDSRAA. Moreover, the computation cost is within the EH IoT device budget.

TABLE III: Overheads

Method	Cycles	Latency(s)
M2M-Routing	304492	0.019
DeepIoTRouting	861292	0.054
Q-Table	262144	0.016
ESDSRAA	116703	0.0073

VI. Conclusion

In this work, we developed a novel environment adaptive multi-hop routing policy for a self-powered IoT system. By leveraging the offline computation resource, we proposed an intelligent deep reinforcement learning-based routing policy for the energy harvesting multi-hop routing. Through clustering power traces into multiple sub-databases and training multiple models, we alleviate the computation burden of EH IoT devices and improve the routing efficiency. To retrieve the similar query power trace precisely, a temporal distance-based similarity algorithm is developed in this paper.

References

[1] "Iot energy-harvesting market size in 2021 with top countries data latest 103 pages report." [Online]. Available: https://www.marketwatch.com

[2] C. Pan, M. Xie, S. Han, Z.-H. Mao, and J. Hu, "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.

[3] C. Pan, M. Xie, Y. Liu, Y. Wang, C. J. Xue, Y. Wang, Y. Chen, and J. Hu, "A lightweight progress maximization scheduler for non-volatile processor under unstable energy harvesting," *ACM SIGPLAN Notices*, vol. 52, no. 5, pp. 101–110, 2017.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, "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.

[6] T. Lu, G. Liu, and S. Chang, "Energy-efficient data sensing and routing in unreliable energy-harvesting wireless sensor network," *Wireless Networks*, vol. 24, no. 2, pp. 611–625, 2018.

[7] S. Tang and L. Tan, "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.

[8] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "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.

[9] X. He, H. Jiang, Y. Song, C. He, and H. Xiao, "Routing selection with reinforcement learning for energy harvesting multi-hop crn," *IEEE Access*, vol. 7, pp. 54 435–54 448, 2019.

[10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[11] A. Cammarano, C. Petrioli, and D. Spenza, "Pro-energy: A novel energy prediction model for solar and wind energy-harvesting wsn," in *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*. IEEE, 2012, pp. 75–83.

[12] M. Morales, *Grokking Deep Reinforcement Learning*. Manning Publications, 2020.

[13] W. Zhang, J. Xie, and Y. Wan, "Spatiotemporal scenario data-driven decision-making framework for strategic air traffic flow management," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. IEEE, 2019, pp. 1108–1113.

[14] "Measurement and instrumentation data center (midc)." [Online]. Available: https://midcdmz.nrel.gov/apps/sitehome.pl?site=ORNL