



Intelligent Networking for Energy Harvesting Powered IoT Systems

WEN ZHANG, Wright State University, USA

CHEN PAN, University of Texas at San Antonio, USA

TAO LIU, Lawrence Technological University, USA

JEFF (JUN) ZHANG, Arizona State University, USA

MEHDI SOOKHAK, Texas A&M University–Corpus Christi, USA

MIMI XIE, University of Texas at San Antonio, USA

As the next-generation battery substitute for IoT system, energy harvesting (EH) technology revolutionizes the IoT industry with environmental friendliness, ubiquitous accessibility, and sustainability, which enables various self-sustaining IoT applications. However, due to the weak and intermittent nature of EH power, the performance of EH-powered IoT systems as well as its collaborative routing mechanism can severely deteriorate, rendering unpleasant data package loss during each power failure. Such a phenomenon makes conventional routing policies and energy allocation strategies impractical. Given the complexity of the problem, reinforcement learning (RL) appears to be one of the most promising and applicable methods to address this challenge. Nevertheless, although the energy allocation and routing policy are jointly optimized by the RL method, due to the energy restriction of EH devices, the inappropriate configuration of multi-hop network topology severely degrades the data collection performance. Therefore, this article first conducts a thorough mathematical discussion and develops the topology design and validation algorithm under energy harvesting scenarios. Then, this article develops *DeepIoTRouting*, a distributed and scalable deep reinforcement learning (DRL)-based approach, to address the routing and energy allocation jointly for the energy harvesting powered distributed IoT system. The experimental results show that with topology optimization, *DeepIoTRouting* achieves at least 38.71% improvement on the amount of data delivery to sink in a 20-device IoT network, which significantly outperforms state-of-the-art methods.

CCS Concepts: • **Networks** → *Network algorithms*;

Additional Key Words and Phrases: Internet of Things (IoT), energy harvesting, deep reinforcement learning

ACM Reference Format:

Wen Zhang, Chen Pan, Tao Liu, Jeff (Jun) Zhang, Mehdi Sookhak, and Mimi Xie. 2024. Intelligent Networking for Energy Harvesting Powered IoT Systems. *ACM Trans. Sensor Netw.* 20, 2, Article 45 (February 2024), 31 pages. <https://doi.org/10.1145/3638765>

This research is supported by National Science Foundation under Grants No.2153524.

Authors' addresses: W. Zhang, Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH 45435; e-mail: wen.zhang@wright.edu; C. Pan (Corresponding author) and M. Xie, University of Texas at San Antonio, 1 UTSA Circle, San Antonio, TX 78249; emails: chen.pan@utsa.edu, mimi.xie@utsa.edu; T. Liu, Lawrence Technological University, 21000 W 10 Mile Rd, Southfield, MI 48075; e-mail: tliu3@ltu.edu; J. (J.) Zhang, Arizona State University, 1151 S. Forest Ave, Tempe, AZ 85287; e-mail: jeffzhang@asu.edu; M. Sookhak, Texas A&M University Corpus Christi, 6300 Ocean Dr, Corpus Christi, TX 78412; e-mail: mehdi.sookhak@tamucc.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1550-4859/2024/02-ART45

<https://doi.org/10.1145/3638765>

1 INTRODUCTION

The last 20 years have witnessed the rapid expansion of IoT devices in a wide variety of applications, such as smart farming, home automation, wearable devices, and so on. Since most IoT devices are currently powered by batteries, they suffer from the limited lifetime and high cost of maintenance on recharging and replacement. Instead of powering by batteries, **energy harvesting (EH) technology** [14] powers IoT devices through harvesting energy from the ambient surrounding environments such as electromagnetic radiation, thermal conduction, kinetics, and the like [9, 47, 48]. Due to the property of environment-friendly, ubiquitous access, and sustainability, energy harvesting is a promising solution to eliminate the dependency on batteries and realize a self-powered IoT system.

Underneath such a prosperous future, the intrinsically weak and unstable power supply of energy harvesting can, however, lead to frequent and unpredictable interruptions to the EH IoT devices [36, 37]. Such intermittent work patterns of EH IoT devices pose imminent challenges to the data routing in the multi-hop EH IoT network. Specifically, unlike battery-powered IoT devices, an EH-powered IoT receiver won't respond to the corresponding transmitter's request if a power outage happens. Thus, in the multi-hop network, the synchronization of energy availability between the transmitter and the receiver is needed for routing. Moreover, with the limited on-board power budget [38, 56], unbalanced energy allocation for runtime operations such as sensing, transmitting, receiving, and forwarding can lead to either energy starvation or squandering. This is because, if the EH IoT device senses too much data, it may not have sufficient energy for routing. Conversely, if the EH IoT device reserve too much energy for routing, it may squander harvested energy for nothing after routing for all the data. To this end, joint optimization of energy allocation and routing selection is needed.

Nevertheless, the aforementioned joint-optimization can be extremely challenging due to the inappropriate configurations of the network topology which determines the performance upper bound of the aforementioned joint-optimization. Specifically, if a large number of EH devices are placed far away from the sink, yet only a few closer-to-sink EH devices (white device in Figure 1) are available for the relay, those relay devices will exhaust their limited stored energy quickly and frequently due to the large relay requests. Such energy exhaustion of relay devices can lead to the infamous *network overflow* problem which inevitably downgrades the overall performance for data delivery regardless of any sort of joint optimization applied. Therefore, before conducting joint optimization of the energy allocation and routing selection, a reasonable topology design for EH IoT devices is expected to foster joint optimization. However, even with an optimized topology, achieving synergy between energy allocation and routing selection for all EH IoT devices together is still full of complexity due to the spatiotemporal dependencies under power-intermittent scenarios. Temporally, for a single device, the current optimal energy allocation that maximizes the data transmission might result in insufficient energy for future data transmission, hence, lowering the overall performance. Spatially, every single device's energy allocation affects its neighbors' decision-making. For example, a device budgets more energy for transmission and also consumes more energy from its next-hop neighbors for the relay.

As EH IoT devices can not obtain global network information, the aforementioned joint optimization needs to be conducted under a partially observable environment. To address the challenges and the complexity in the partial observable environment, **Deep Reinforcement Learning (DRL)** [27, 28, 31] has been considered as one of the most promising optimization tools. DRL trains an agent that directly learns a policy from rewards by interacting with the environment. The policy is a mapping from the agent's current observed state to its best next action. The environment changes the state and generates a reward corresponding to the agent's current action.

DRL is able to make a sequence of decisions under uncertainty with an outstanding performance from a long-term perspective. By formulating energy allocation and routing selection as partially observable Markov Decision Processes, in this article, we design a distributed and highly scalable DRL-based approach known as *DeepIoTRouting*, to maximize the energy efficiency and the packet delivery for sensing and transmitting data in the EH IoT systems.

This article first conducts a thorough mathematical discussion and builds a corresponding theoretical foundation regarding the influence of topology on the network performance under energy harvesting scenarios. Then, based on the foundation, an optimized topology design flow is given to help place the EH devices in a way that can facilitate the following joint optimization. Next, to optimize the energy allocation strategy and routing selection jointly whilst addressing the challenges and the complexity in the partial observable environment, a distributed and highly scalable DRL-based approach, known as *DeepIoTRouting*, is proposed. The key idea of *DeepIoTRouting* is to train and deploy a unique DRL agent on each IoT device in the network. Each unique agent controls the local energy allocation on sensing and transmission and determines the data routing destination. With the goal of maximizing the amount of data delivery to the sink, *DeepIoTRouting* dynamically fine-tunes its energy allocation policy in response to the real-time behavior of sensing and transmission activities. By controlling the relaying destination, *DeepIoTRouting* finds the near-optimal routing path for EH IoT devices. More important, each agent shares the same neural network architecture (except for the input/output layers) but has its own set of optimized *weights*. During inference, each device only requires its neighbors' information and thus has a significantly smaller state space than centralized DRL for large-scale IoT systems. In a nutshell, the major contributions of this article are as follows:

- (1) A comprehensive multi-hop routing system is developed for the EH IoT system that incorporates EH IoT device topology, multi-hop routing, and energy-harvesting models (as shown in Figure 8);
- (2) Spatial dependency among self-powered IoT network devices is analyzed mathematically. Based on the analysis, the topology design requirement of the self-powered IoT system is given in Theorem 1 and Corollary 1.1;
- (3) A distributed DRL-based approach, *DeepIoTRouting*, is designed to address the coupled energy allocation and routing selection problem for large-scale EH IoT networks;
- (4) A set of comprehensive experiments is conducted to evaluate the proposed *DeepIoTRouting* from five perspectives, including topology influence exploration, parameter discussion, overall comparison, network performance evaluation, and design space exploration.

The remainder of this article is organized as follows: Section 2 discusses related work. Section 3 gives the example to illustrate the motivation of this article. Section 4 introduces the system model, analyzes the topology requirements of the self-powered IoT system, and formulates the joint-optimization problem. In Section 5, based on the topology requirement analysis in Section 4 we explore the topology design strategy of the self-powered IoT system. Also, our reinforcement-based multi-hop routing and energy allocation learning algorithms are proposed in this section. Evaluations and experiments are demonstrated in Section 6. Finally, Section 7 concludes this work.

2 RELATED WORK

Existing works related to this article can be mainly summarized into two categories including topology design and routing policy. The routing policy can be further grouped into two categories: traditional routing policy and energy-harvesting-aware routing policy.

2.1 Topology Design

Great efforts have been devoted to the topologies of the wireless network. The classical topology architectures include point-to-point, ring, tree, star, and grid network topologies. There are several summative surveys as well as studies on specific exploration. [6, 24] provided a comprehensive survey on the topology design of wireless sensor networks. [30] concluded the network topologies adopted in water wireless sensor networks. A trunk-and-branch tree network design on the surface of the earth was proposed [51]. Based on the designed tree topology, [51] studied the path planning of submarine cables. In [10], a topology with optimal synchronization is explored, where the ring network is discussed in detail. [53] compared the spectrum utilization for the flexible grid topology and the fixed grid topology. There are also a set of studies that researched topology to maximize the sensor coverage [11, 29], extend the network lifetime [24], or optimize the sensor connectivity [25]. However, existing studies on topology design mainly focus on the communication performance and the scalability of topology [23]. None of them considered the energy availability of devices and explored the topology design requirements for energy harvesting devices.

2.2 Routing Design

2.2.1 Traditional Routing. Given that the router can work continuously in the traditional network where the devices are connected to the wall outlet, existing works [3, 58] mainly investigated the shortest path (reducing latency) and minimum energy consumption in data transmission [17]. Also, geographic routing [13, 46, 57] is proposed, where a set of sensors were available to obtain the GPS location and determine the next hop for a transmitted packet based on the location information. In [26], a geographic routing with unreliable link (GRUL) protocol was proposed to route the collected data along energy-efficient paths. However, those traditional routing policies adapted well in the stable power environment are impractical for the EH environment due to unpleasant data packet loss caused by power interruptions.

2.2.2 Energy-harvesting-aware Routing. Non-Learning: To address the above issues, developing an energy-harvesting-aware routing policy that maximizes the transmission rate under limited energy consumption is essential for the emerging intelligent EH system [12, 17, 43, 44, 50]. Nguyen et al. [33] 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. [4] jointly optimized 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. [26] developed an algorithm, named ESDSRAA, 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 which is the core for EH devices.

Learning: Recently, DRL is increasingly adopted in EH IoT system design to improve performance. DRL shows outstanding performance in decision making in an uncertain environment considering the long-term influence of its actions [18, 34, 49]. [49] 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. [34] employed a typical DRL algorithm to optimize the power allocation for two-hop EH communications. Q-table was created in [18] to find the optimal routing path for EH multi-hop cognitive radio network. Several existing approaches have been introduced in the literature that exploits intelligent methods in order to support the EH in IoT environments, such as [8, 45] by introducing low complexity solutions in contrast to the DRL approaches that are characterized by high computational complexity. However, these solutions only target small-scale communication environments (i.e., at most 6 devices [18]), far away from the

realistic IoT system setting that consists of a large number of interconnected devices. Besides, few works consider how to synergistically perform energy allocation and routing for EH IoT systems.

2.3 Novelty Analysis

Different from the previous studies that design topology without consideration of energy availability or only focus on routing policy for EH IoT network with the centralized method, this paper mathematically discusses the influence of the topology in EH IoT system for the first time. Based on our findings, we provide detailed requirements on topology design for the multi-hop EH IoT system. Moreover, we also propose *DeepIoT Routing* that jointly optimizes the energy allocation and routing policy in a distributed manner. The current existing works target small-scale communication environments (i.e., at most 6 devices [18]), far away from the realistic IoT system setting that consists of a large number of interconnected devices. Besides, few works consider how to synergistically perform energy allocation and routing for EH IoT systems. Thus, in the simulation, we first compare the data collection performance of the topology qualified by the proposed validation algorithm, typical tree, and grid topology on the energy harvesting aware distributed system. After that, given the qualified topology, the proposed *DeepIoT Routing* is compared with *GRUL* [26], *ESDSRAA* [26], and *Q-table*, which represent traditional routing, energy-harvesting-aware routing, and energy-harvesting-aware routing with DRL, respectively.

3 MOTIVATION EXAMPLE

This section presents two examples to illustrate the motivations for the reasonable topology design and the extreme challenges of the routing and energy allocation joint optimization for the multi-hop self-powered IoT system.

3.1 Why Reasonable Topology Design Matters

Figure 1 shows an 8-device self-powered IoT edge network organized in a tree topology. Each IoT device performs environmental sensing sustainably in the deployment area. The sensed data will be delivered to the sink (Star). Due to the limited transmission distance, only the devices (White) have direct access to the sink. Therefore, white devices need to forward data coming from the rest of devices (Purple). However, while fulfilling the continuous forwarding request from the six purple devices, due to insufficient harvesting power and limited data storage capacity, two White devices can neither satisfy the continuous forwarding request nor support their own data gathering. Thus, the sensed data get stuck on devices resulting in *data overflow* of the network. To clarify the example, we assume $E_{sense} = 0.1J$, $E_{trans} = 1J$, and $E_{recev} = 0.5J$ as the energy cost of the individual device for sensing, receiving, and transmitting 2KB data as one data unit. Each device is required to sense $N_{data} = 10$ unit data and transmit them to the sink. Therefore, the total energy consumption of devices 3-8 is $E_{sense} * N_{data} + E_{trans} * N_{data} = 11J$. Because devices 1 and 2 need to receive all data packets from six devices and relay those received data packets to the sink, the average individual energy consumption of devices 1 and 2 is $E_{sense} * N_{data} + 4E_{trans} * N_{data} + 3E_{trans} * N_{data} = 56J$, which implies the insufficient power supply for device 1 and 2. This situation will get worse when the network scales up since each one-hop device (White) of the sink needs to forward data for more EH IoT devices.

To address this issue, if we adapt the old topology to a new topology with more devices closer to the sink as shown in Figure 2, the aforementioned problem can be significantly alleviated. This is because right now four white devices will not only be able to support their own data sensing/transmission but also can help forward data for the rest of the four purple devices to the sink. Thus, it is vital to reasonably deploy self-powered IoT devices considering limited energy harvesting power. To provide clear guidance on topology design, we first conduct the

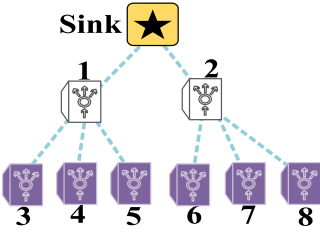


Fig. 1. Example 1 on topology of EH IoT system.

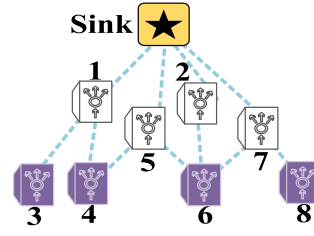


Fig. 2. Example 2 on topology of EH IoT system.

mathematical analysis regarding the influence of topology on the performance of self-powered IoT systems in Section 4.1, and then illustrate the topology design strategy in Section 5.1.

3.2 The Challenge of Routing under Energy Harvesting Scenarios

Even with a reasonable topology, a poor routing policy can significantly degrade the network performance. However, due to the weak and intermittent nature of energy harvesting power, developing an appropriate routing policy becomes extremely challenging. Specifically, routing consists of multiple point-to-point communications which require sufficient energy from both the transmitter and the receiver. Hence, the amount of energy allocated for communication will heavily affect the network performance. However, due to the limited onboard energy of each single device, the amount of energy allocated for communication temporally depends on the amount of energy allocated for previous local tasks. Moreover, if the receiver's energy is inappropriately allocated or the receiver encounters unstable harvesting power, point-to-point communication may not have sufficient energy to support the data receiving from the spatially dependent transmitters. In the following, we will use two examples to illustrate the routing challenges from temporal and spatial perspectives.

Temporal Dependency. Figure 3 shows an example of the real-time energy status of an individual device without any optimization. Here, $E_{m,t}$, $E_{m,r}$, $E_{m,s}$ represent the minimum energy threshold to start the transmission, receiving, and sensing operations where E_{sleep} means the device will be put into sleep mode for charging if the energy depletes to below E_{sleep} . As we can observe, right after the harvested energy meets the threshold requirements, the device immediately starts the corresponding sensing, receiving, and transmission based on its task queue. In this way, even if charging power is large enough to support constant receiving and sensing, it is still unable to support the transmission continuously. As a result, the energy reservoir will be quickly drained as the storage energy at threshold $E_{m,t}$ can barely support such short $t_1 - t_0$ for completing the transmission.

Alternatively, if we slow down the sensing and receiving to reserve energy for transmission, the device will experience a long period of recharge time and fails to perform the task reactively (the first blue region of Figure 4). Most importantly, while the device is reserving the energy for transmission by slowing down the sensing and receiving, it might cause that there is not enough data for transmission during the high power supply period (the second blue region in Figure 4). This situation will further result in the device missing the opportunity to utilize energy during the high power supply period. As the energy storage of the device is limited, if the device reserves more energy in the storage, the available space for energy storage is less. If the device only performs sensing operation during a high power supply period (no data for transmission), it implies that the device misses the opportunity to harvest and utilize energy from natural sources, because of the availability of energy storage.

Spatial Dependency. Given an optimized topology in Figure 2, the goal of the IoT edge network is for each IoT device to deliver enough data to the sink for comprehensive analysis. Since an

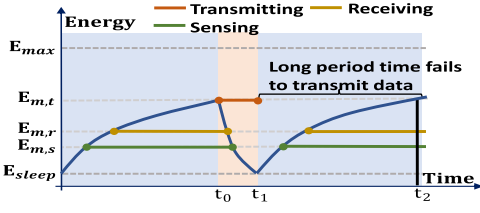


Fig. 3. Example 1 on individual EH device execution.

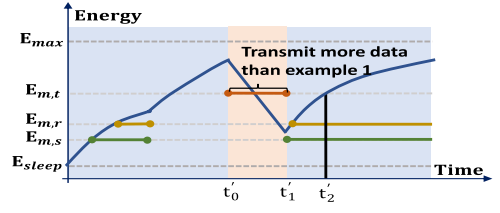


Fig. 4. Example 2 on individual EH device execution.

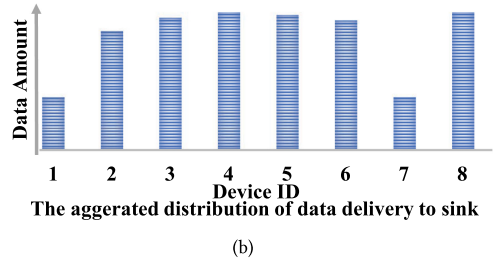
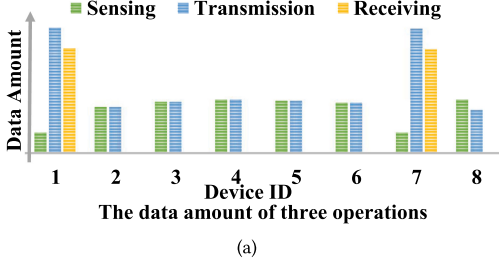


Fig. 5. Example 1 on routing selection of EH IoT system.

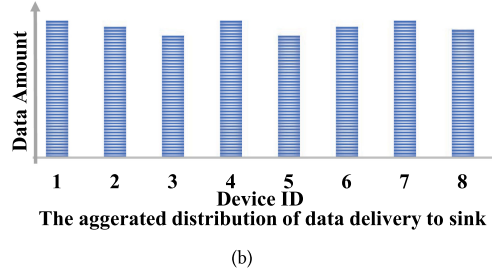
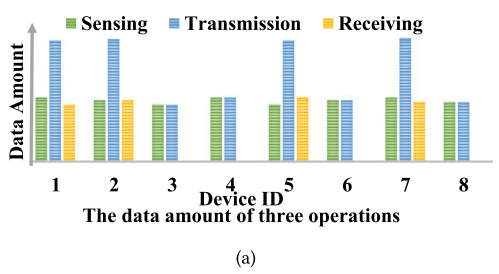


Fig. 6. Example 2 on routing selection of EH IoT system.

individual's routing choice is unknown to others, without any optimization, devices 3 and 4 may select device 1 for relaying, and devices 6 and 8 may select device 7 for relaying. As a result, if devices 1 and 7 fulfill all the relay requests, based on the aforementioned discussion (temporal dependency), less energy will be allocated for sensing, which results in the data distribution in Figure 5(b) extremely uneven on the sink. As we can observe, even if devices 1 and 7 devote the majority of their harvested energy for transmission and receiving as shown in Figure 5(a), the data originating from devices 1 and 7 are far less than the data originating from others. To address this issue, if devices 1 and 7 are allocated less energy for relay by properly reducing the relay queue, corresponding relay requests can be evenly distributed among all the white devices (1, 5, 2, 7) as shown in Figure 6(a). As a result, more energy can be allocated for sensing and delivery of their own data for devices 1 and 7 while at the same the originality of the data can become more evenly distributed in Figure 6(b)).

Therefore, the routing optimization under energy harvesting scenarios is tightly coupled with each individual's local energy allocation. Hence, it is crucial to conduct a joint optimization of energy allocation and routing selection. Nevertheless, such a joint optimization is extremely challenging. First, the future harvesting power magnitude is hard to predict in advance making energy allocation extremely difficult. Second, each IoT device only has knowledge of its neighbors in close

Table 1. List of Notations

Symbol	Description	Symbol	Description
i	id of device (Subscript i indicates transmitter)	j	id of device (Subscript j indicates receiver)
n	the total number of EH devices	$\mathcal{D}$	the set of EH devices
$\mathcal{S}$	the sink (the edge server)	ξ	the transmission range
N_i	the set includes i 's neighbors	Δ	the capacity of queue
Λ	a threshold to inactivate sensing to prevent overflow	$A_{i,j}$	the size of data packet transmitting from i to j
$\omega_{i,j}$	the transmission rate	k	the k th data packet in queue
D	the set to include the forwarding devices' label of the data packet	Γ	the number of forwarding hops of the data packet (the cardinality of D)
Φ	the budget of forwarding hops	$E_{i,cost}$	the energy cost of i
$E_{i,current}$	the residual energy of i	$E_{i,ini}$	the initial energy of i
$E_{i,hav}$	the harvested energy of i	$E_{i,max}$	the i 's capacity of energy
$\mathcal{H}_i$	the layer id of i	H	the maximum geo-graphical layer id
h	the h th geo-graphical layer	M_h	the number of devices in the h th layer
E_t	energy cost for transmitting one bit	E_r	the energy cost for receiving one bit
E_s	energy cost for sensing one bit	v_s	the average sensing rate
h'	the h' th hop layer	$M_{h'}$	the number of devices that is $H' - h' + 1$ hops away from the sink
$d_{i,hop}$	the average hop distance when i delivers data packets to the sink	$N_{a,i}$	the set that contains the next-hop neighbors of i ; the layer id of devices in $N_{a,i}$ is no smaller than $\mathcal{H}_i$
$M_{h'}^{exp}$	the minimum expected number of devices in the h' th layer	s_t	the state of environment at t time step
a_t	the DRL agent action at t time step	r_t	the returned reward at t time step
γ	the discounted factor	Q	Q value of DRL agent
θ, θ'	the weights of the local neural network and the target neural network of <i>DeepIoT</i> Routing	$\mathcal{E}_i$	the energy status of i 's neighbors
Q_i	the available queue size of i 's neighbors	$\mathcal{I}_i$	the data packet heading in the queue _{i} is generated by $\mathcal{I}_i$
$\mathcal{T}_i$	the energy threshold of i	$\mathcal{N}_i$	the destination of current data packet

proximity and thus can not select optimized routing paths from a global perspective. To address the above challenges with the complexity of joint optimization for energy allocation and routing selection in a partially observable environment, we will employ powerful DRL as optimization tools. The details of the proposed design are in Section 5.2. Table 1 lists the notations used in this article.

4 SYSTEM MODELS AND ANALYSIS

In this section, we first introduce the system models of the multi-hop routing EH IoT system. Then, based on the models, we conduct mathematical analysis regarding topology design. In the end, we formulate the problem and specify the optimization objectives for the EH IoT system.

We consider a classical data transmission scenario [35, 59, 60], where a set of EH IoT devices are deployed in an open space such as in a wild forest or farmland. Each device in Figure 7 is equipped with environmental sensors, a transmission module, and an energy harvesting module, which enable the devices to conduct sensing, transmitting/relaying, and receiving continuously with harvested ambient energy. Since the edge server (the sink) collects data from all devices including those at a long distance, each device also serves as a router for relaying incoming packets towards the sink $\mathcal{S}$ (the edge server). Figure 8 shows the overview of the whole system.

Deployment. Assume the system has n EH IoT devices and each device switches between runtime status “awake” and “sleep” according to its energy level. Let i denote the i th EH IoT device, where $i \in \mathcal{D} = \{1, 2, \dots, n\}$. Device i and device j are neighbors if the distance between them is within the transmission range ξ . The device only can communicate with its neighbors. Specifically, all one-hop neighbors of device i are grouped into a set N_i . In addition, the deployment area is divided

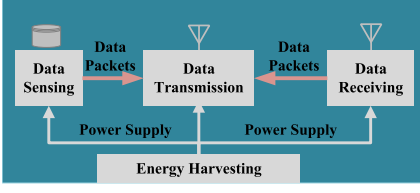


Fig. 7. Operations of an EH IoT device.

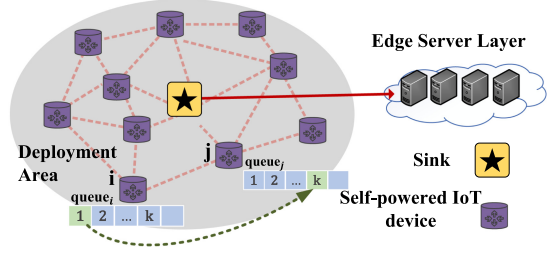


Fig. 8. System model of multi-hop routing for EH IoT system.

into H layers based on the distance to the sink (S) as shown in Figure 9. The width of each layer is equal to the transmission range ξ .

Sensing. Each EH IoT device i has a transmission queue **queue _{i}** for buffering the local sensing data or received data for the relay. The size of each **queue _{i}** is Δ . Without loss of generality, Each EH IoT device will follow a preset schedule to start its sensing. If the stored energy is enough, it keeps sensing (“awake”) at a preset rate v until the energy has been drained. By then, the system will go to low-power “sleep” mode. To prevent the queue overflow, a threshold Λ ($\Lambda \leq \Delta$) is set to inactivate sensing. When the queue is above the threshold, the remaining space in the queue will only be used for routing. In practice, Λ can be chosen by profiling the data traffic of each device.

Communication. We define the size of each data packet transmitting from i to j as $A_{i,j}$, where $i = j$ means the data packet from local sensing. The transmission time $t_{i,j,send}$ can be obtained from

$$t_{i,j,send} = \frac{A_{i,j}}{\omega_{i,j}}, \quad i \in \mathcal{D}, j \in N_i \cup S, \quad (1)$$

where $\omega_{i,j}$ is the transmission rate, which can be calculated from Shannon theorem [5, 7, 42, 54].

Since the EH IoT device works intermittently, the device enters sleep mode frequently. Thus, the time cost of the k th data packet on device i is equal to the transmission time of the past $k - 1$ data packets plus the device sleeping time ($t_{i,j,sleep}$), given by (2).

$$t_i^k = \sum_{l=1}^k t_{i,j,send}^l + \sum_{l=1}^k t_{i,j,sleep}^l, \quad i \in \mathcal{D}, j \in N_i \cup S. \quad (2)$$

In the multi-hop network, the data packets might be trapped in the network for several routing rounds. If the data packets get trapped and are routed repeatedly among EH IoT devices, it wastes the energy of EH IoT devices. Therefore, those data packets got trapped for too many routing rounds are considered as expired. To drop the data packets that get trapped in the network all the time, the budget on routing path hops Γ is assigned to the data packet. When the number of routing path hops Φ becomes greater than Γ , the data packet expires and will be dropped. Furthermore, the data packet is routed with ids of its relaying devices labeled as set D to prevent the routing loop. If the data packet is transmitted back to the same device ($j \in D$), the data packet will be dropped. Moreover, to prevent collision during transmitting, this paper employs the well-known CSMA/CA protocol [19, 21, 40]. Specifically, during the connection setup phase, the transmitter sends a “Hello” message, and the receiver capable of decoding the message reply with an “ACK” that includes its ID and current energy information, facilitating the connection establishment. Once the connection is established, the transmitter can send data to the receiver.

Energy Harvesting. The total energy cost of device i includes energy consumption of sensing, transmitting, receiving, and sleeping, and thus defined as,

$$E_{i,cost} = \int P_{i,sense} dt + \int P_{i,send} dt + \int P_{i,recv} dt + \int P_{i,sleep} dt, \quad i \in \mathcal{D}. \quad (3)$$

$\int P_{i,send} dt$ and $\int P_{i,recv} dt$ represent the energy costs on transmitting and receiving data packets, respectively. In particular, $\int P_{i,send} dt$ and $\int P_{i,recv} dt$ include the energy costs on building the connection between the transmitter and the receiver (such as energy costs on handshaken mechanism).

The real-time residual energy of device i is,

$$E_{i,current} = E_{i,ini} + E_{i,hav} - E_{i,cost}, \quad i \in \mathcal{D}, \quad (4)$$

where $E_{i,ini}$ is the initial energy and $E_{i,hav}$ is energy harvested from ambient energy sources. At any given time, $E_{i,current}$ of device i should not exceed the $E_{i,max}$, which is the energy capacity of the device.

4.1 Topology Analysis

To ensure the system has a reasonable topology, motivated by Section 3.1, we provide the mathematical discussion in terms of the energy constraint, device number, and network topology. The mathematical analysis provides the constraints on the deployment of EH IoT devices.

As shown in Figure 9, the deployment area is divided into H layers based on the distance to the sink (S). The width of each layer is equal to the transmission range ξ . Thus, each EH IoT device also has a *layer id* attribute based on their geo-locations, denoted as $\mathcal{H}_i, \mathcal{H}_i \in [1, H]$. The *layer id* attribute of the EH IoT device implies the minimum number of routing hops, i.e., it at least needs $H-h+1$ times of relaying when the data packet is delivered from the h th layer to the sink. Moreover, the devices located at $(h-1)$ th layer will finally transmit their data packet to the h th layer devices. This means the data from the h^{th} layer include the sensed data from the devices within the h th layer and all the relay data from devices with lower *layer id* ($\mathcal{H}_i < h, \forall i \in \mathcal{D}$). Therefore, given the operation time T and energy consumption for sensing, transmission, and receiving as E_s, E_t, E_r per bit, respectively, we have (5). Here, M_h represents the number of devices located in h th layer, E_T is defined as the maximum energy that can be utilized by an individual device, and v_s is the average sensing rate of the self-power IoT system. This inequality represents that the total energy of M_h devices should be greater than the energy cost of sensing, transmission, and receiving of devices located at h th layer.

$$M_h E_T \geq v_s T E_s M_h + v_s T E_t \sum_{l=1}^h M_l + v_s T E_r \sum_{l=1}^{h-1} M_l \quad (5)$$

Based on (5), we will further have (6) which provide a constraint regarding to the number of devices M_h in the the h th layer.

$$M_h \geq \frac{(E_t + E_r)v_s T}{E_T + (E_r - E_s)v_s T} \sum_{l=1}^h M_l. \quad (6)$$

To better guide the topology design, based upon Equation (6), we further propose Theorem 1 which gives the relationship between M_h and M_1 . Hence, the number of devices in any layer can have a clear constraint if M_1 is given.

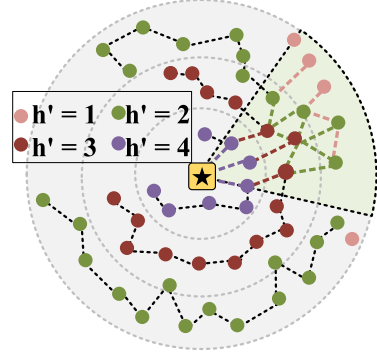
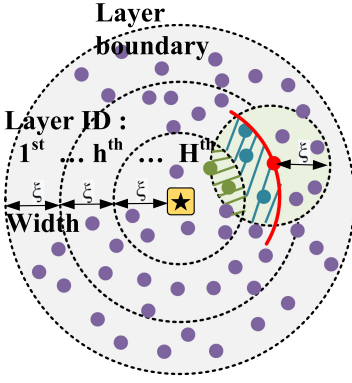


Fig. 9. Deployment EH IoT devices: Geo-geographical layer. Fig. 10. Deployment EH IoT devices: Hop layer.

THEOREM 1. *Given M_1 as the number of devices located at 1st layer, the number of devices in h th layer must satisfy (7).*

$$M_h \geq \frac{v_s T(E_t + E_r)(E_t + (E_r - E_s)v_s T)^{h-2}}{(E_t - E_s v_s T - E_t v_s T)^{h-1}} M_1 \quad (7)$$

The proof of Theorem 1 is illustrated in Appendix A.

Notice that, in (5), (6), and (7), only the across layer transmission and receiving are considered (inter-layer communication). However, there are communications among devices that are within the same layer (intra-layer communication). For instance, in Figure 9, the red device (i) has multiple choices to select its next-hop neighbors (blue and green devices) located in the shadow area. If a blue device (j) is selected for relay, both i and j are within the same layer ($\mathcal{H}_i = \mathcal{H}_j$). As the intra-layer communication consumption has not been considered in (5), the inequality is incomplete. To formulate the constraints regarding to the number of devices in each layer more precisely, instead of distinguishing layers based on the distance to the sink, we employ the number of “hops” to the sink as the metric to determine the *layerid*. To formulate the constraints regarding to the number of devices in each layer more precisely, instead of distinguishing layers based on the distance to the sink, we employ the number of “hops” to the sink as the metric to determine the *layerid*, as shown in Figure 10. Specifically, we define $M_{h'}$ as the number of devices that is $H' - h' + 1$ “hops” away from the sink. In this new setting, the devices that are $H' - h' + 1$ hops from the sink will send all the sensed data from the devices within the h' th layer and all the relay data from devices with lower *layer id* ($\mathcal{H}_i < h', \forall i \in \mathcal{D}$).

Since a device might have multiple paths to the sink, the number of hops to the sink is uncertain, which makes it difficult to determine $M_{h'}$. To address this issue, we assume that device i will select its next-hop neighbors for data relay with the same probability. Under such context, if i selects its neighbor j for relay, the hop counts from i to the sink is determined by the hop counts from j to the sink, hence, $d_{i,hop} = d_{j,hop} + 1$. Since every neighbor has the same chance to be selected, the hop counts from i to Sink can be calculated as (8), which represents the mathematical expectation of hop counts for device i

$$d_{i,hop} = \text{Round} \left(\sum_{j \in N_{a,i}} d_{j,hop} \frac{1}{|N_{a,i}|} \right) + 1, N_{a,i} \in N_i. \quad (8)$$

Here, $N_{a,i}$ is a set containing all possible next-hop neighbors of device i , $|N_{a,i}|$ represents the number of neighbors in $N_{a,i}$, $\frac{1}{|N_{a,i}|}$ represents the probability to select j for the next-hop relay,

$d_{j,hop}$ is the mathematical expectation of j 's hop counts to sink, and $Round(x)$ always rounds x to the nearest integer. Since the data should flow towards the sink, $N_{a,i}$ only contains neighbors that has smaller Euclidean distance to the sink compared with i .

As we can observe that if we categorize the devices with the same hop count into the same layer, there will be no intra-layer communication. Under such context, the conditions of Equations (5)–(7) become valid. Hence $M_{h'}$ represents the number of devices whose $d_{i,hop}$ is equal to $H' - h' + 1$. Based on Equation (6), Corollary 1.1 further provides the constraint of $M_{h'}$.

COROLLARY 1.1. *Given the maximum energy that can be utilized by EH IoT devices and the hardware parameters of EH IoT devices (E_s, E_t, E_r , and v_s), the number of devices $M_{h'}$ in layer h' should satisfy*

$$M_{h'} \geq \frac{(E_t + E_r)v_s T}{E_T + (E_r - E_s)v_s T} \sum_{l=1}^{h'} M_l. \quad (9)$$

Corollary 1.1 will further be used in Section 5. Equation (9) highlights that, as we move from the outermost layer to the sink layer, the number of sensors increases exponentially. This relationship can be readily demonstrated using Taylor Series, as shown in Appendix A. It is interesting to note that even though the outer layer has a larger area, it requires fewer sensors. This implies a higher sensor density as we approach the sink layer. Considering this observation, it becomes evident that in real-world scenarios, additional communication sensors should be deployed. This would account for the increasing density of sensors as we move closer to the sink layer, ensuring robust and efficient communication within the network.

4.2 Problem Formulation

The main goal of this article is to maximize the total amount of received data by the sink (S) in EH IoT systems under the energy harvesting setting. Formally, given the EH IoT devices with limited onboard battery capacity, buffer queue size and the system operation time T , a dynamic multi-hop routing topology (varies according to each device's status), and an external power source, our optimization problem is described as:

$$\begin{aligned} & \text{maximize} && \sum_{i \in \mathcal{N}, j \in \mathcal{S}} \sum_{t=0}^{t=T} A_{i,j} \\ & \text{subject to} && 0 \leq E_{i,current} \leq E_{i,max}, && \forall i \in \mathcal{D} \\ & && \sum_{l=1}^k A_{i,j}^l \leq \Delta, && \forall i \in \mathcal{D}, j \in N_i \\ & && j \notin D_i, && \forall i \in \mathcal{D}, j \in N_i \\ & && \Phi \leq \Gamma, && \forall i \in \mathcal{D}, j \in N_i \end{aligned}$$

Unlike traditional cellular network that targets real-time communication, in this article, we explore energy harvesting powered embedded IoT devices for sensing applications without time constraint.

5 ALGORITHM DESIGN

In this section, we will first provide the device deployment strategy along with the detailed topology validation procedures. Based on the optimized topology, we will provide the detailed design flow for *DeepIoT Routing*.

5.1 Topology Design and Validation

Based on the analysis from Section 4.1, in this subsection, we will first provide the device deployment strategy in Algorithm 1. Then, we will provide detailed topology validation procedures in Algorithm 2. Since at the very beginning we will not know the hop-count without given topology, the devices will (1) based on Theorem 1 for fast deployment with distance-based metric. After deployment, we will (2) use Corollary 1.1 to conduct accurate validation with the hop-count metric.

5.1.1 Topology Design. In the first phase, given the boundary of the deployment area and estimation of M_1 , we first estimate the approximate minimum number of devices in each geographical layer through (7) of Theorem 1. Following this requirement on the number of devices in each geographical layer (Algorithm 1), the topology will be initialized based on the specific service requests.

ALGORITHM 1: Topology Design

```

Input:  $\Delta, \xi, M_1$  ;                               /*  $\Delta$ : The maximum width of deployment area; */
                                                    /*  $\xi$ : The transmission range; */
                                                    /*  $M_1$ : The approximate number of devices at the 1th layer */

Output:  $\langle M_{id1} \rangle$ ; /* The initial requirement of topology from geo-graphical perspective */

1  $H \leftarrow \Delta/\xi$ ;
2 for  $id1 \leftarrow 2$  to  $H$  do
3   | Calculate  $M_{id1}$  based on (7);
4 end
5 return  $\langle M_{id1} \rangle$ 
  
```

5.1.2 Topology Validation. After the initial topology is established, in the second phase, based on the provided topology, we will validate the topology based on Corollary 1.1 for effective deployment. First, we will calculate $d_{i,hop}$ based on (8) to assign hop IDs for all devices starting from the inner circle of the sink to the outer rim of the network. Then, we can categorize the devices with the same hop count into the same layer to calculate the number of devices on each hop-based layer and record in set $\langle M_{h'} \rangle$ (Line 1-6, Algorithm 2). Finally, the provided topology is validated by Corollary 1.1 (Line 8-14, Algorithm 2). In particular, for each layer h' , we also calculate the lower bound of $M_{h'}$ as $M_{h'}^{exp}$ based on (9), which represents the minimum number of devices required in each hop layer. If every layer satisfies $M_{h'} \leq M_{h'}^{exp}$, the topology is qualified for deployment. Otherwise, the topology can be fine-tuned based on $\langle M_{h'} \rangle$ and $\langle M_{h'}^{exp} \rangle$.

Heterogeneity Discussion. While it is true that real-world sensor networks often exhibit device heterogeneity, including variations in transmission range, such heterogeneity does not significantly affect the performance of our proposed approach and can be simplified to the homogeneous network (our current experimental setup) without compromising the validity and performance of our work. We found that these variations in transmission range did not result in significant performance differences compared to the homogeneous case. The reason behind this is that we can correctly identify the corresponding neighbors of the sensor, even that the transmission range among sensors are different. As long as the neighbors of each sensor are correctly identified, the exact topology of the network is determined and becomes less important in our context. Our work primarily focuses on optimizing energy allocation and routing policies, which primarily depend on local interactions among neighboring sensors. Therefore, after figuring out the location of all sensors, even that sensors have different transmission ranges, we can identify the neighbors of all sensors. We then can simplify and model them using our current experimental setup without

compromising the validity and performance of our work. Our current approach captures the essential dynamics and behaviors of the system, enabling us to evaluate and validate our proposed algorithms effectively.

ALGORITHM 2: Topology Validation.

```

Input:  $(x_i, y_i), \forall i \in \mathcal{D} \cup \mathcal{S};$  /* The locations of all devices and sink */
Output:  $\text{Is\_qualify}, \langle M_{id3} \rangle, \langle M_{id4}^{exp} \rangle;$ 
1 for  $id2 \leftarrow 1$  to  $n$  do
2   | Calculate  $d_{id2, hop}$  by (8); /* Calculate the expected hop distance of each device */
3 end
4 for  $id3 \leftarrow 1$  to  $H'$  do
5   | Count  $M_{id3};$  /* Count  $M_{h'}$  for the current topology design */
6 end
7  $\text{Is\_qualify} \leftarrow \text{True};$ 
8 for  $id4 \leftarrow 2$  to  $H'$  do
9   | Calculate  $M_{id4}^{exp}$  based on (9); /* Count the expected minimum  $M_{h'}^{exp}$  for each hop layer */
10  | if  $M_{id4} < M_{id4}^{exp}$  then
11    |  $\text{Is\_qualify} = \text{False};$  /* Validate the current topology */
12  | end
13 end
14 return  $\text{Is\_qualify}, \langle M_{id3} \rangle, \langle M_{id4}^{exp} \rangle$ 

```

5.2 DeepIoTRouting Design

In this section, we present *DeepIoTRouting* framework to address the multi-hop routing problem. We formulate the program in the RL setting and discuss the designs for the RL state, action, and reward in detail.

5.2.1 RL Overview and Q-Learning. Reinforcement learning is the training of an agent to make a sequence of decisions. The agent learns an *optimal policy* by interacting with a complex *environment*. At time step t , the environment has a *state*, $s_t, \forall s_t \in S$ and it changes in response to the agent's *action* $a_t, \forall a_t \in A$. After action a_t , environment state transits from s_t to s_{t+1} and the agent also receives an immediate reward r_{t+1} from the *environment*. The goal of the agent is to maximize the total *reward* by optimizing its policy $\pi(s, a)$ (i.e., the mapping from the state S to action A).

Q-learning [52] is a classic example of RL. In Q-learning, the cumulative discounted reward from time step t is defined under the policy π as $R_t^\pi = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_{\tau+1}$, where the γ is the discounted factor for future reward, $\gamma \in (0, 1]$. Then, the Q-function can be formulated as:

$$Q(s, a)^\pi = \mathbb{E}[R_t^\pi | s_t = s, a_t = a]. \quad (10)$$

Essentially, Q-learning maintains a lookup table (Q-table) and follows ϵ -greedy [52] to update its q-value iteratively until convergence. The update rule can be described as follows:

$$Q(s, a) \leftarrow Q(s, a) + \delta[r_t + \gamma \max_a Q(s, a_{t+1}) - Q(s, a)] \quad (11)$$

$$\pi = \underset{a}{\operatorname{argmax}} Q(s, a). \quad (12)$$

5.2.2 DeepIoTRouting Framework. Q-learning can achieve optimal results for small-scale RL problems. However, for problems with large state size (e.g., continuous variable) and action space, the Q-table will grow significantly huge. Further, for large-scale RL problems, the amount of time to explore each state of the Q-table will be unrealistic. To solve these challenges, deep-Q network [31,

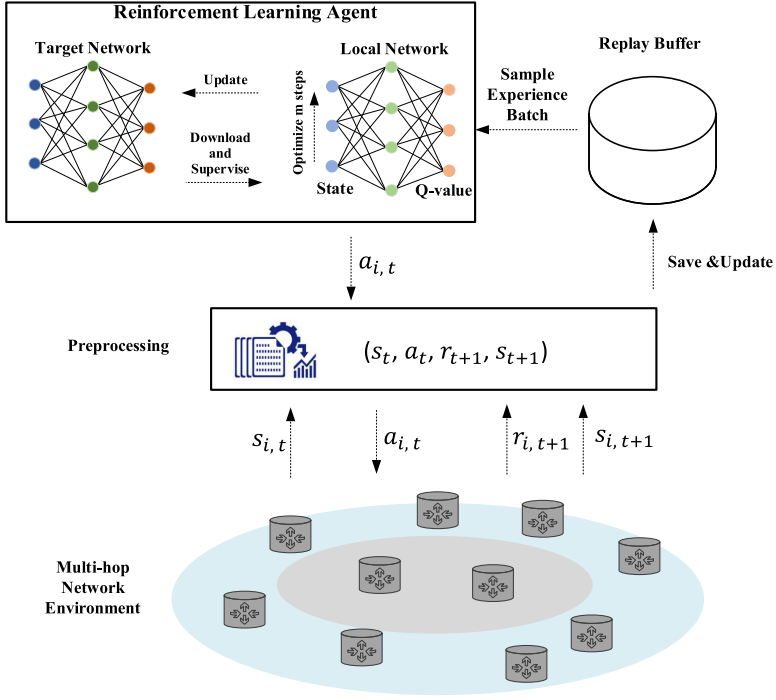


Fig. 11. DeeploTRouting framework.

55] has been proposed to *train* a function approximator, i.e., neural network, to estimate the Q-values. The proposed *DeeploTRouting* adopts deep-Q-network as the underlying agent to solve the multi-hop routing and energy allocation problems for the EH IoT system.

Figure 11 illustrates our *DeeploTRouting* framework. At bottom, *DeeploTRouting* models the multi-hop IoT system (discussed in Section 4) as the *environment*. For each EH IoT device in the system, *DeeploTRouting* deploys a unique DRL agent that controls the data packet destination device and energy allocation on sensing and transmitting. Doing so makes *DeeploTRouting* easily scale as the number of devices in the system increases. Moreover, all DRL agents in *DeeploTRouting* share the same neural network architecture (except for the input/output layers), but have their own set of learned *weights*. During inference, each device only requires its neighbors' information and thus has a significantly smaller state space than a centralized RL agent for large-scale IoT systems.

Training. To facilitate the training process, *DeeploTRouting* utilizes an experience replay buffer to record interactions between the agent and environment as *experience* = $(s_t, a_t, r_{t+1}, s_{t+1})$. During training, instead of updating the agent network parameters (θ) after each agent-environment interaction, *DeeploTRouting* waits until a mini-batch of training samples are available from the replay buffer. To further stabilize the training, *DeeploTRouting* also adopts a target network (θ') that has the same architecture as the original (local) network but with frozen parameters for every m iteration (hyperparameter). In other words, the weights of the target network are periodically updated from the weights of local network. Thus, the loss function of a local agent i for a randomly sampled mini-batch Z_t is described as:

$$L(\theta) = \sum_{\substack{s_{i,t}, a_{i,t}, \\ r_{i,t+1}, s_{i,t+1} \in Z_t}} (r_{i,t+1} + \gamma \max_a Q(s_{i,t+1}, a_{i,t+1}; \theta') - Q(s_{i,t}, a_{i,t}; \theta))^2 \quad (13)$$

ALGORITHM 3: Training procedure for *DeepIoT*Routing.

Input: $max_episodes, start_time, end_time, m, n_device, \{\gamma, \epsilon_{start}, \epsilon_{end}, \delta\}$ (for learning and exploration)

Output: optimized n_device agents θ_i

```

1 Initialize  $n\_device$  local networks,  $\theta_i$ , with random weights,  $\forall i$ ;
2 Initialize  $n\_device$  target networks,  $\theta'_i \leftarrow \theta_i, \forall i$ ;
3 Initialize the corresponding replay buffer,  $Z_i, \forall i$ ;
4  $n\_episode \leftarrow 0$ 
5 while  $n\_episode < max\_episodes$  do
6   Reset EH IoT system environment;
7   Initialize  $start\_time$ ;
8    $Step_i \leftarrow 0 \quad \forall i$ ;
9   while  $time < end\_time$  do
10    for  $i$  from 0 to  $n\_device$  do
11      if  $i$  request action from agent  $\theta_i$  then
12         $n\_random \leftarrow random.random(), n\_random \in [0, 1);$       /* Perform decaying
13          epsilon-greedy policy; */
14        if  $n\_random \leq \epsilon_i$  then
15           $a_{i,t} \leftarrow random\ action;$ 
16        else
17           $a_{i,t} \leftarrow \operatorname{argmax} Q(s, a)$ 
18        end
19        Perform  $a_{i,t}$ ; Feedback  $r_{i,t+1}$  based on (14);      /* DRL agent interacts with the
20          environment; */
21         $experience_i = (s_{i,t}, a_{i,t}, r_{i,t+1}, s_{i,t+1});$ 
22        Save experience to buffer  $Z_i$ ;      /* Collect experiences of agents and save to
23          the replaybuffer; */
24        if  $replaybuffer \geq mini\_batchsize$  then
25          Randomly sample mini-batch of experiences;      /* Update neural network
26            weights; */
27          Optimize  $\theta_i$  with sampled experiences;
28          if  $Step_i \% m == 0$  then
29            Update target network  $\theta'_i$ ;
30          end
31        end
32        Update  $\epsilon_i$  with decaying epsilon algorithm;
33         $Step_i = Step_i + 1$ ;
34      end
35    end
36  end
37   $n\_episode = n\_episode + 1$ 
38 end

```

The training process is described in Algorithm 3. Sample data are normalized in the preprocessing step. To avoid local minima, we adopt a decaying epsilon-greedy policy [15, 32] during training. At early stages, the DRL agents take more actions to explore, and as the training progresses, our agents make decisions based on their learning.

Deployment & Computational Analysis. We pre-train the models on the server. Once the models are well-trained, the corresponding models will be deployed into the EH devices for runtime decision-making. At the runtime stage, the models are optimized during the off-peak time. The computational complexity of *DeepIoTRouting* during the runtime stage relies on the architecture of the adopted neural network. For example, if the fully connected neural network is adopted, the computation complexity is $O(x^2)$. The neural network architecture setting is given in Section 6.

5.2.3 Reinforcement Learning Settings for DeepIoTRouting. We describe our designs of the *state*, *action*, and *reward* for the sensing and multi-hop routing in the DRL setting.

State. Each EH IoT device i in the system has its own state $s_i, i \in \mathcal{D}$ to represent its energy level, queue size, and data packets. The state is denoted by $s_i = \{\mathcal{E}_i, \mathcal{Q}_i, \mathcal{I}_i\}$. $\mathcal{E}_i$ is a vector that represents the current energy status of the device i and its neighbor devices N_i . $\mathcal{Q}_i$ represents the current queue size of the device i and its neighbor devices N_i . $\mathcal{I}_i$ stores the source *id* of the data packet heading the queue. Data packet source *id* can be useful to prevent routing loop.

Action. Action $a_i = \{\mathcal{T}_i, \mathcal{N}_i\}$ of an EH IoT device i in the system has two components. First, $\mathcal{T}_i$ is a *dynamic* parameter that controls the device i 's *real-time* energy allocation policy between sensing (energy harvesting) and transmission/receiving. We quantize the device's residual energy into multiple levels. If the quantized energy is less than $\mathcal{T}_i * E_{i,max}$ after an action, the EH IoT device stops its transmission/receiving and only performs low-power sensing and harvesting. When the device recharges to sufficient energy level ($\geq \mathcal{T}_i * E_{i,max}$), it resumes the transmission/receiving tasks and actions related to routing is described with second component $\mathcal{N}_i, \mathcal{N}_i \in N_i$. Since each IoT device in *DeepIoTRouting* has its own DRL agent, $\mathcal{N}_i$ is the destination of the current outgoing data packet in device i . Note that all *DeepIoTRouting* DRL agents (EH IoT devices) execute in a distributed parallel fashion, but each agent's action affects their neighbors' energy. Thus a holistic training scheme from *DeepIoTRouting* is required.

Reward. Our goal is to maximize the amount of received data at the sink. Our reward design also explicitly considers routing loop prevention, packet dropping reduction that directly affects the network performance. The intermediate reward $r_{i,t}$ of *DeepIoTRouting* at time step t is designed as:

$$r_{i,t} = \begin{cases} 0, & \text{if } \mathcal{I}_i = i, \Gamma < 0, E_{j,current} < 0, \sum_{l=1}^k A_{i,j}^l > \Delta \\ \log(2 + \mathcal{H}_i - \mathcal{H}_j) + f(j), & \text{otherwise} \end{cases} \quad (14)$$

Equation (14) captures two scenarios. First, the transmitted packet is successfully received by the destination device j (The second equation of (14)). If a data packet gets routed closer to the sink (Recall that $\mathcal{H}_i$ is the layer id attribute of i), the agent gets a positive reward. While the data packets are routed toward the opposite direction of sink, $r_{i,t} \leq 0$ is satisfied. $f(j)$ is a bonus for the agent if the packet gets routes to the sink directly at this step, given in Equation (15).

$$f(j) = \begin{cases} \log(n), & \text{if } j = S \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

Second, the agent gets 0 reward for the transmission failure (The first equation of (14)). *DeepIoTRouting* considers several common data transmission failure scenarios in real EH IoT systems: (1) *Routing Loop*. Without careful design, data packets may get stuck in a loop (i.e., $\mathcal{I}_i \in D_i$) and waste the whole system's resources and energy. Real IoT system has a mechanism to detect such scenario and drop the problematic data packet [22]; (2) *Data Expiration*. Each data packet in the system also has a limited "hop" budget Γ to ensure its routing efficiency. Data packets that have less than 0 "hop" budget (i.e., $\Gamma < 0$) will get dropped; (3) *Receiver Failure*. Because of the transient nature of energy harvesting, the device at the destination might be offline ($E_{j,current} < 0$).

Table 2. EH IoT Network Parameters

Notation	Definition	Value
$A_{i,j}$	size of data packet	3072 bits
Δ	maximum queue size	$15 * 3072$ bits
$P_{i,send}$	transmission power	0.1 w
$P_{i,recev}$	receiving power	0.05 w
$P_{i,sleep}$	sleeping power	0.0005 w
$P_{i,sense}$	sensing power	0.01 w
$E_{i,max}$	energy storage capacity	1 J
v	sensing rate	200 bits/s
ξ	transmission range	20 m
H	layer number	3
$\mathcal{T}_i$	the energy threshold	$\mathcal{T}_i \in \{0, 0.25, 0.5, 0.75\}$

In this case, the sender cannot get the ACK resulting in wasted energy. (4) *Device Queue Overflow*. In the case that the **queue** _{j} of the destination device j reaches its capacity Δ ($\sum_{l=1}^k A_{i,j}^l > \Delta$), the packet will be refused by the destination device but will be transmitted to the different neighbor device. We also consider this as a failure since it costs energy waste. Our evaluation in Section 6 incorporates all these failure scenarios that are modeled by *DeepIoTRouting*.

6 EVALUATION

In this section, we develop an energy harvesting powered IoT network simulator to explore the topology influence and evaluate the performance of the proposed *DeepIoTRouting* framework. To demonstrate the transferability of the framework, we conduct experiments under both solar and wind EH scenarios.

6.1 Experimental Settings

System Initialization. We simulate our EH IoT networks of the similar architecture as shown in Figure 8. The system begins operating from 8:00 AM to 18:00 PM, and the residual energy on each device ($E_{i,current}$) is initialized to 0.5J, half energy storage capacity (1J) to ensure the sufficient start-up energy. We simulated 80 episodes, and each episode represents the network behaviors on one day in the real world. As Zigbee is a widely used wireless communication protocol known for its low-power consumption, low data rate, and suitability for applications involving sensor networks and IoT devices, we utilize Zigbee [16, 41] as the underlying protocol. The transmission range has been set to 20 meters. Table 2 lists the parameters used in our EH IoT network simulation.

DeepIoTRouting. We train *DeepIoTRouting* for 80 episodes. We set discount factor $\gamma = 0.95$ and the learning rate $\delta = 5e^{-4}$. The number of device neighbors decides the neural network architecture in the *DeepIoTRouting*. The dimensions of the input layer and output layer are determined based on the definition of state and action in Section 5. The structure of distributed neural network algorithm is 256-1024-512-128. At the beginning of each day, the environment will be reset, and all measured parameters will be re-initialized. This is for a fair comparison that prevents the system from being influenced by residual effects such as remaining energy from the previous day.

Topology Baselines. We compare our designed topology by following Section 5.1 with the typical (1) **Grid Topology** and (2) **Tree Topology**. As shown in Figure 12, we deploy 3, 9, and 8 devices in M_1 , M_2 , and M_3 , respectively. The two baselines are (1) **Grid Topology** [20]. In Figure 13, we deploy devices following the grid topology, where the devices are deployed at each vertex of the 20×20 grids; and (2) **Tree Topology** [39]. Figure 14 indicates a typical tree topology, where

each master device has two members. Note that, instead of giving a certain network topology design, Section 5.1 aims to validate a reasonable topology for a self-sustaining IoT multi-hop routing system. Thus, this article evaluates the proposed topology validation algorithm by comparing the amount of data delivery to the sink with the same routing policy among different topologies. Therefore, the amount of data delivery in Figures 12–14 to the sink are compared, where the topology in Figure 12 is validated by Algorithm 2 but topologies in Figures 13 and 14 fails to be validated.

DeepIoTRouting Baseline. We compare our *DeepIoTRouting* algorithm with three baselines: (1) **EH-aware Routing with RL: Q-table** [18]. The Q-table baseline [18] deploys one Q-table in each EH IoT device with a decaying ϵ -greedy policy; (2) **EH-aware Routing: ESDSRAA** [26]. This baseline allocates energy and gives a constant sensing rate at the beginning of each hour with TPAA. After that, it decides the destination device (GRUL) based on the delivery rate (link quality) and geo-location information [26]; (3) **Traditional Routing: GRUL** [26]. This baseline makes decisions on the relaying destination device based on the delivery rate and geo-location information without considering the energy allocation [26].

6.2 Experimental Analysis on Solar Powered IoT Network

We first apply the real solar power traces from the *National Renewable Energy Laboratory (NREL)* [1] to evaluate proposed EH IoT system from the follow five perspectives: (1) topology influence exploration, (2) parameter discussion, (3) overall comparison, (4) network performance, and (5) design space exploration. To obtain an appropriate system topology and further prove Theorem 1 and Corollary 1.1 via experiments, we measure the amount of data delivery to sink for three different topologies, including the topology generated based on the proposed topology design strategy, the grid network topology, and the typical tree topology. After selecting a suitable topology for the self-powered IoT system, the system sensing rate and the energy threshold levels of *DeepIoTRouting* for energy allocation need to be configured. Thus, we investigate the influence of the sensing rate and the action $\mathcal{T}_i$. We then compare the proposed *DeepIoTRouting* with three baselines. We also analyze the network performance of the self-powered IoT system through providing details on the sensing, transmission, receiving, and the action of *DeepIoTRouting*. Finally, we scale up the system to explore the *DeepIoTRouting* performance in a large-scale system.

6.2.1 Topology Influence Exploration. To evaluate the three topologies, we separately train *DeepIoTRouting* with three topologies. After *DeepIoTRouting* is well trained, we measure the corresponding amount of data delivery to the sink in one day. As shown in Figure 15, *DeepIoTRouting* achieves 28.31Mb data delivery in one day with the topology of Figure 12, while it completes 9.75Mb and 5.83Mb with the grid topology and the typical tree topology, respectively. The topology of Figure 12 achieves $\sim 2.90\times$ and $\sim 4.86\times$ more delivery data than that of the grid topology and typical tree topology. The reason is that the number of devices directly linked with the sink (one-hop devices) in Figure 12 is far more than that in Figures 13 and 14. In Figures 13 and 14, there are only 4 and 3 one-hop devices, which have to relay the data packets for 20 devices to the sink. Because of the insufficient energy and limited capability of one-hop devices, the generated data can not be delivered, as analyzed in Section 3.1. We also provide the amount of data generated by all devices of each topology in Figure 15. However, even if the typical tree is with more sensed data than that of the other two topologies, the sensed data of the typical tree topology cannot be transmitted to the sink. Therefore, there is a *data overflow*, which further proves the analysis in Section 3.1.

6.2.2 Parameter Discussion. Given the topology and the hardware parameters of self-powered IoT devices, the sensing rate has to be configured. Figure 16 indicates the amount of data delivery to the sink versus the training day for the different sensing rate settings. When the sensing rate

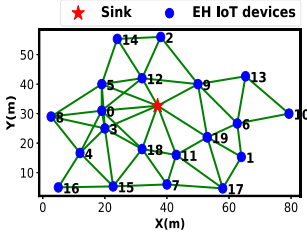


Fig. 12. Topology based on Theorem 1.

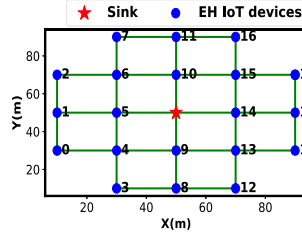


Fig. 13. Grid topology.

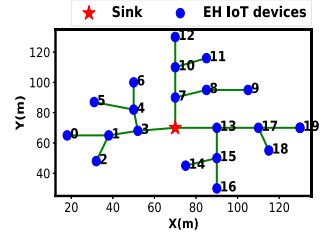


Fig. 14. Typical tree topology.

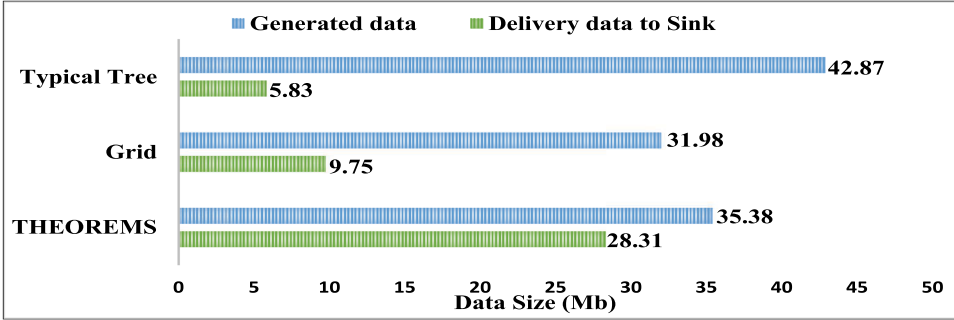


Fig. 15. The sink received data of three topologies.

increases from 100bps to 200bps, the data delivery to sink is increased dramatically. However, as the sensing rate continuous increasing, the growth of the data delivery progressively slows down. The amount of data delivery to sink is almost the same when the sensing rate is at 300bps and 400bps. As shown in Figure 16, 400bps should be the best choice of the sensing rate. However, the system performance can not be evaluated only via the total amount of data delivery to sink. The fairness among all devices should also be considered; namely, the data delivery to sink should come from all devices and the delivery amount of each device should be at the same level in general. Therefore, we also investigate the data distribution by comparing the amount of data delivery to sink from each device.

As shown in Figure 18, when the sensing rate is 300bps or 400bps, despite that the sink receives more data, the distribution is severely aggregated on a set of devices. The sink barely receives any data from devices 3, 9, 11, and 19 (Device ID is noted in Figure 12). Although in Figure 16, the amount of data delivery at sensing rate 200bps is slightly smaller than that at sensing rate 300bps and 400bps, the distribution of sensing rate 200 bps is uniform and the amount of data delivery at this rate is higher than that at sensing rate 100bps. The data distribution is aggregated when the system is with a high sensing rate. This is because the devices far away from the sink (marginal devices) do not need to relay data for other devices and always sense at a high rate. Thus, those marginal devices will transmit all their data to their neighbors (intermediate layer devices). Due to the limited storage resource (limited queue size) and on-board energy, the intermediate devices have to stop their own sensing task to assist data relay for their neighbors, which results in a low data delivery amount of the intermediate devices. The fact that devices 3, 9, 11, and 19 all are located at the intermediate layer between the marginal devices and the sink validates the above analysis.

To explore the appropriate setting of $\mathcal{T}_i$, Figure 17 indicates the amount of data delivery to sink for different $\mathcal{T}_i$ settings. Recall that $\mathcal{T}_i, \mathcal{T}_i \in [0, 1]$ is the action of *DeepIoT Routing* to quantize energy to multiple levels. In Figure 17, we increase the granularity of $\mathcal{T}_i$ from 1 to 4. Specifically, when

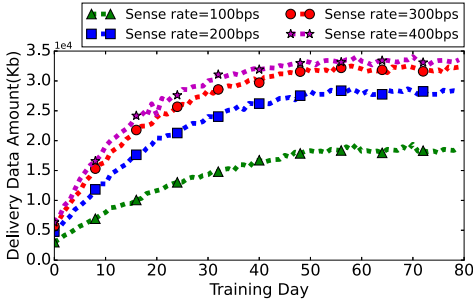


Fig. 16. Sensing rate for DeepIoT Routing.

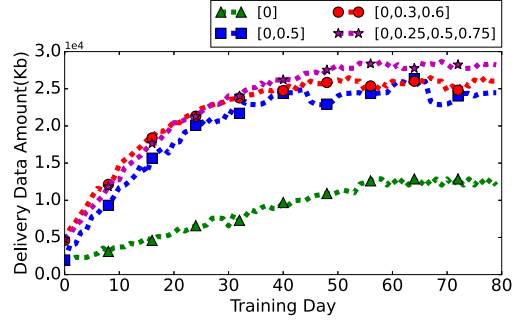


Fig. 17. Energy threshold for DeepIoT Routing.

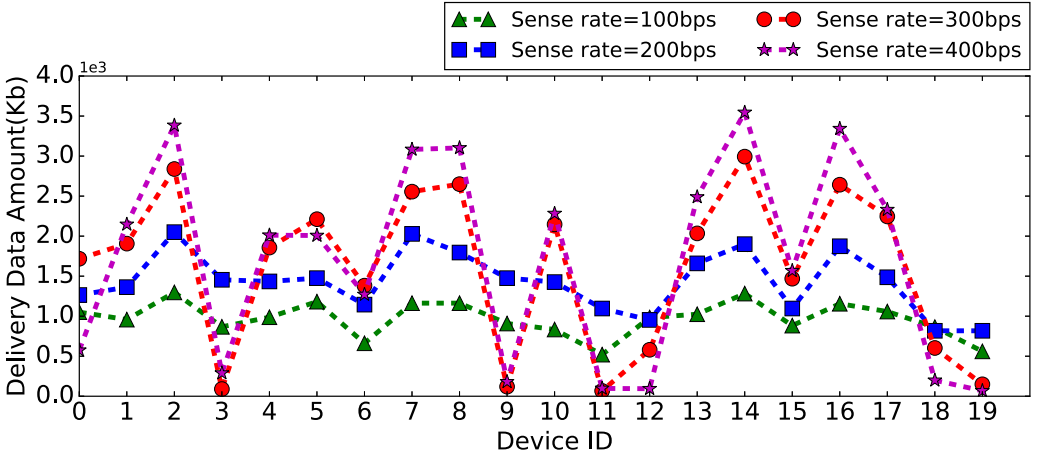


Fig. 18. Fairness: Distribution for four sensing rates.

$\mathcal{T}_i = 0$, it means there is no joint energy allocation. All devices will execute operations as long as there is residual energy. Figure 17 indicates a jump gain of data delivery from $\mathcal{T}_i = 0$ to $\mathcal{T}_i = [0, 0.5]$. Without energy allocation, the devices can not dynamically fine-tune sensing and transmission operations to balance the sensing and communication, which leads to inefficient energy utilization. Once there is energy allocation, with the granularity of $\mathcal{T}_i$ increasing, the growth of data delivery to sink gradually increases. In the following experiments, we set $\mathcal{T}_i = [0, 0.25, 0.5, 0.75]$ as it achieves the best performance.

6.2.3 Overall Comparison. We compare DeepIoT Routing with three baselines, including Q-table, ESDSRAA, and GRUL from *Efficiency*, *Effectiveness*, *Fairness*, and *Learning Performance* perspectives.

Efficiency. We first evaluate the efficiency by measuring the delivery rate, the amount of sensed data, the amount of data delivery to the sink, and the execution time (computation overhead) as listed in Table 3. We define the delivery rate as the amount of data delivery to sink divided by the amount of data generated by all devices. The proposed DeepIoT Routing algorithm agent achieves the highest delivery rate (80.02%) compared with Q_table (41.31%), ESDSRAA (69.56%), and GRUL (24.06%), respectively. However, one concern is that EH IoT devices can also “play a trick” if we only measure the delivery rate. For example, the device can generate fewer sensed data to achieve high data transmission and delivery rate. Therefore, we jointly evaluate the received data size,

Table 3. The Efficiency Evaluation (Daily Average) in Solar-Powered Scenario

Algorithm	Delivery Rate (%)	Sensed Data (Mb)	Delivery Data (Mb)	Execution Time (s)
<i>DeepIoTRouting</i>	80.02	35.38	28.31	0.12
Q_Table	41.31	30.26	12.50	0.11
ESDSRAA	69.56	29.34	20.41	0.024
GRUL	24.06	52.82	12.71	0.024

Table 4. The Efficiency Evaluation (Daily Average) in Wind-Powered Scenario

Algorithm	Delivery Rate (%)	Sensed Data (Mb)	Delivery Data (Mb)	Execution Time (s)
<i>DeepIoTRouting</i>	93.50	189.45	177.12	0.12
Q_Table	86.98	187.60	163.18	0.11
ESDSRAA	46.52	338.59	157.50	0.024
GRUL	34.08	459.12	156.49	0.024

the sensed data size, and the delivery rate to measure the energy efficiency. As listed in Table 3, although *DeepIoTRouting* maintains a similar sensed data level compared with the other three baselines, it achieves $\sim 2\times$ the delivered data amount on the sink (~ 28 Mb) than that of the Q_table and GRUL. While compared with ESDSRAA, *DeepIoTRouting* achieves 38.71% improvement on the amount of data delivery to the sink. Our goal is to maximize this term. In contrast, GRUL senses the most data (52.82Mb) while the sink of GRUL only receives 12.71Mb data. When devices sense more data, they are expected to transmit more data to the sink, but much less data than expected are delivered to the sink of GRUL. This indicates that the agent senses more data that may not have sufficient energy to transmit data or prevent the routing loop. Therefore, the packet may be dropped due to the expiration caused by the routing loop or exceeding the hop budget. In addition, ESDSRAA performs better than GRUL. The only difference between ESDSRAA and GRUL is that the former algorithm has the energy allocation, which further proves the importance of energy allocation. The results from Table 3 indicate that the efficiency of *DeepIoTRouting* outperforms the three baselines on both energy allocation and routing selection.

The execution time in Table 3 is measured when the MCU frequency is 16MHz. The execution time listed in Tables 3 and 4 refers to the inference time of our neural network model. These execution times reflect the computational cost of making decisions at runtime. As shown in Table 3, the execution time of *DeepIoTRouting* at the inference stage, 0.12s, is the greatest among the four algorithms, which is slightly longer than Q_Table but about $5\times$ of the execution time of the ESDSRAA and GRUL. Such an overhead is due to the execution of the DRL model. Nevertheless, the data delivery on EH IoT system is not expected to achieve the real-time performance due to the intermittent nature of EH power. Therefore, the impact of such time overhead is negligible. Besides, what is even more appealing is that compensating for the execution overhead is the significant improvement of energy utility. Specifically, under the same energy harvesting scenario, *DeepIoTRouting* achieves 38.71% and 122.74% more data delivery to the sink compared with ESDARAA and GRUL, respectively.

Effectiveness. We synthetically measure the residual energy, the amount of data delivery to the sink, and the amount of sensed data to evaluate the algorithm's effectiveness. We measured these data on an hourly basis in order to accurately and deeply analyze the algorithm. Figures 19 and 20 show the hourly received data and sensed data from 8:00 AM to 18:00 PM. As shown in Figures 19 and 20, *DeepIoTRouting* has the dominant hourly data delivery at the sink while the sensed data

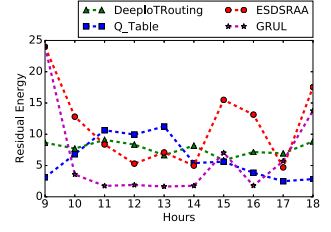
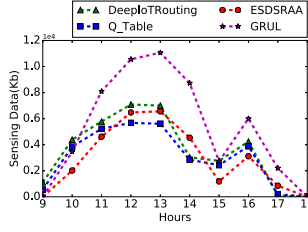
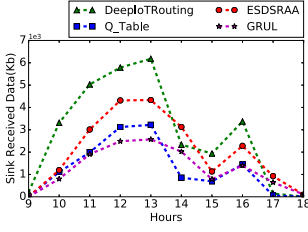


Fig. 19. Data delivery to sink in hours. Fig. 20. Sensed data in hours. Fig. 21. Residual energy in hours.

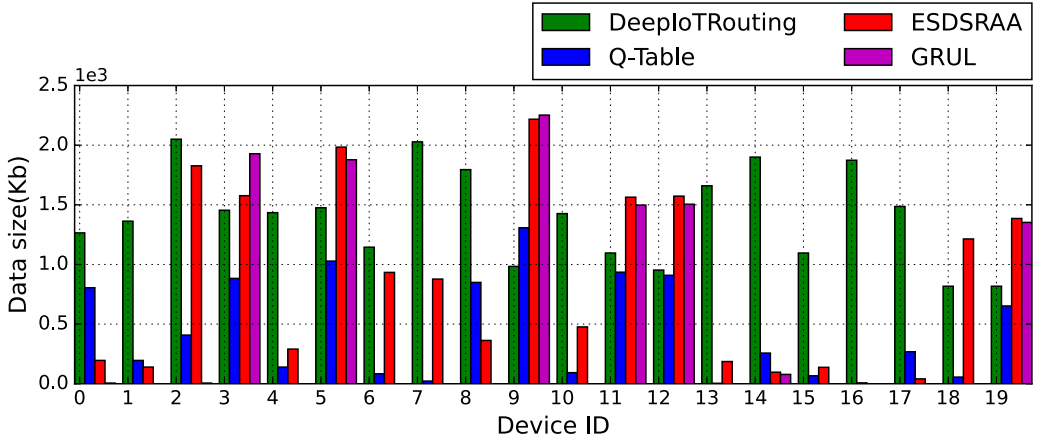


Fig. 22. Distribution of data delivery to sink (Daily).

does not maintain the top level. It is worth noting that the trend of Figure 20 and Figure 19 are similar to the trend of harvesting power trace. From 8:00 AM to 12:00 PM, the EH IoT devices sense more data with the increased solar power and the sink receives an increasing amount of data. After that, the power trace curve drops as well as the curve of sensing data and data delivery at the sink. Between 14:00 PM and 18:00 PM, all of them experience another jump and drop.

The energy level of EH IoT devices should vary with different tasks including sensing, transmitting, and receiving. Figure 21 shows the energy variation of four algorithms. *DeepIoT Routing* can maintain the energy at a stable level, unlike ESDSRAA which suffers from a dramatic fluctuation. Moreover, *DeepIoT Routing* has a lower level of residual energy, which means that it discharges more and performs more tasks, and thus it can make good use of the recharging opportunities. Although ESDSRAA maintains enough energy to balance sensing and transmission, it misses the opportunity to recharge due to the limited battery capacitance. There is no energy allocation in the GRUL; therefore, it depletes the energy and then harvests energy for the next task. The residual energy of the Q-table also remains at a stable level as *DeepIoT Routing*, but due to its limited routing capability, it reaches a local optimum early. We will analyze this issue in the evaluation of learning performance.

Fairness. In EH IoT multi-hop system, our goal is to not only maximize the received data on the sink but also ensure that the received data originates from all IoT devices. In this way, the data delivery to the sink can have a balanced distribution with a promising delivery rate. Thus, to evaluate the fairness, in Figure 22 and Figure 23, we show the amount of data delivery to the sink of each device and the delivery rate achieved by each device, respectively. As we can observe, for ESDSRAA and GRUL algorithms, each local agent achieves a large amount of data delivery to

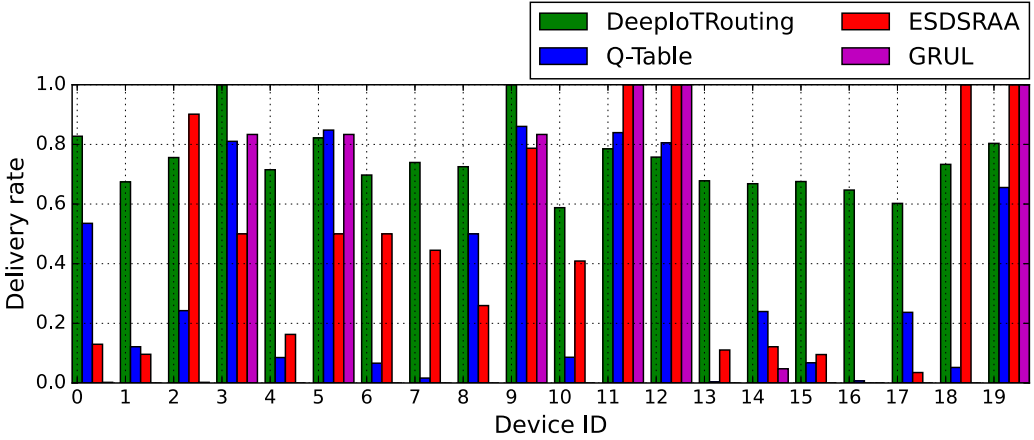


Fig. 23. Distribution of delivery rate (Daily).

the sink and a high delivery rate when devices are close to the sink. However, when the device is located at the edge of the network, the packets are difficult to be transmitted to the sink, which indicates poor data distribution. For *DeepIoTRouting*, it achieves a balanced distribution. The sink can receive evenly distributed data from most devices when *DeepIoTRouting* is adopted. Notice that, a slightly smaller amount of data are received from device 8 and device 9 than other devices because device 8 and device 9 consume energy for forwarding packets to its neighbors and sense fewer data, which will be proved in Figure 26. For devices having more neighbors that can not transmit data packets to the sink in one hop, we can increase their energy capacity or storage capacity when we deploy the sensor network. The sink can receive data from most devices in Q-table. However, the number of data delivery to sink and delivery rate from each device with Q-table algorithm is lower than the performance of *DeepIoTRouting*. Therefore, *DeepIoTRouting* has an outstanding performance in both data distribution and efficiency.

It is worth noting that, in an energy-hungry scenario, sensors that are closer to the sink should sense fewer data than sensors at the boundary, because all sensed data by sensor devices located at the boundary are forwarded by their next-hop neighbors that are closer to the sink. Therefore, considering the limited harvesting energy budget, devices that are closer to the sink should sacrifice their individual data sensing rate and allow more energy to be used for the relay in order to achieve a global optimal data delivery. We observe that even with the reduced sensing rate, those devices can still yield a high delivery rate with *DeepIoTRouting*. The results in Figure 22 and Figure 23 show that *DeepIoTRouting* align with this pattern and balances network workload well. The three baseline algorithms for comparison receive few data packets from devices 14, 15, 16, and 17, which proves the packet drop in a set of areas. Three baseline algorithms fail to balance network workload and fail to achieve desirable data delivery rates.

Learning Performance. As shown in Figure 24, we measure the daily obtained reward $r_{i,t}$ to evaluate the learning performance of the agent. The Q-table can converge quickly on around 10th day. Although *DeepIoTRouting* converges gently at 40th day, it gains a $\sim 2\times$ rewards over Q-table, which further shows the huge performance gap between *DeepIoTRouting* algorithm and Q-table when encountering large-scale multi-hop routing. When implementing the Q-table, we have to balance the discrete density and the number of states. Increasing the discrete density raises the number of states in the Q-table, leading to more complicated searching for the optimal actions. Decreasing the discrete density reduces the number of states, further making the search easier; however, the Q-table cannot accurately capture the environment states. This article simplifies the

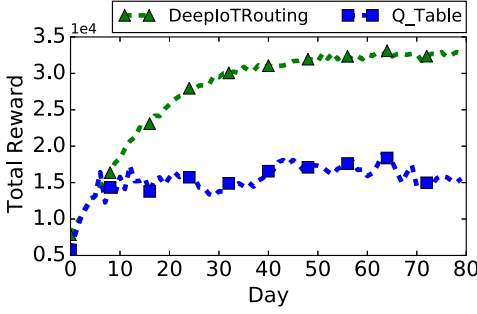


Fig. 24. Daily reward.

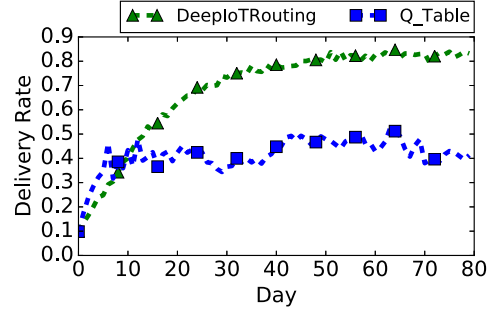
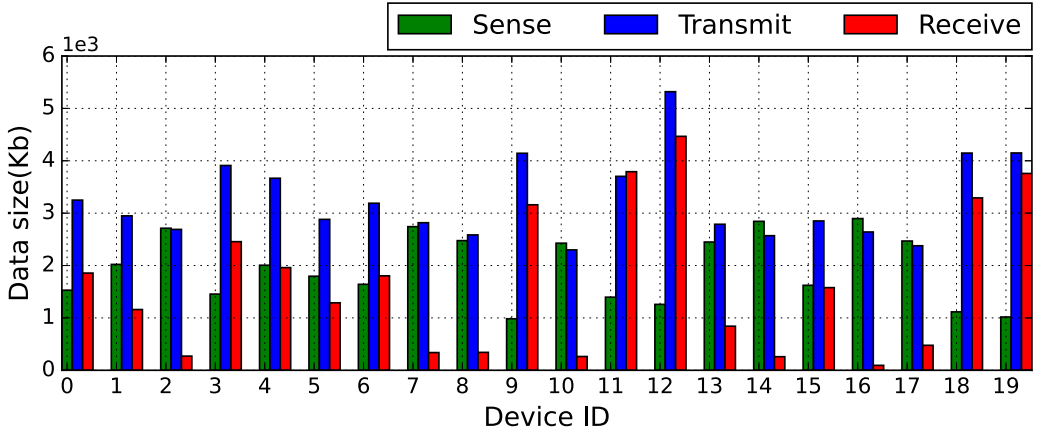


Fig. 25. Daily delivery rate.

Fig. 26. Daily routing data distribution for *DeepIoTRouting*.

neighbor states in Q_table , where the status of $\mathcal{E}_i, Q_i$ are represented by a binary value. We also remove $\mathcal{I}_i$ from Q_table since it leads to narrowed-down action space that can dramatically reduce the complexity of Q_table optimal searching. Furthermore, in case the agent obtains a high reward but without a high performance in the network, we also measure the daily received data amount by the sink (see Figure 25). The trends of reward for the *DeepIoTRouting* are similar to the curve trend in Figure 25, which further proves that the reward setting in Equation (14) is suitable for our EH IoT multi-hop routing system.

6.2.4 Network Performance. We also provide details on the sensing, transmission, receiving, and routing action of each device to indicate how *DeepIoTRouting* works at each device. The transmitted data of each EH IoT device comes from two sources, including sensed data on its own and received (relay) data from its neighbors. Figure 26 indicates the amount of the sensed, transmitted, and received data on each EH IoT device. Ideally, the amount of sensed and received data together should be equal to the amount of transmitted data on each EH IoT device. However, Figure 26 indicates that the amount of sensed and received data together is slightly larger than the transmitted data on most EH IoT devices. This means that a few data packets are dropped due to the routing loop or packets expiration.

Besides, the sensed data of devices 9, 18, and 19 are around 1Mb in Figure 26. All of those devices are one-hop neighbors of the sink. Since they also have many neighbors that are far away from the sink, devices 9, 18, and 19 need to relay data to the sink for their neighbors. This reduces the

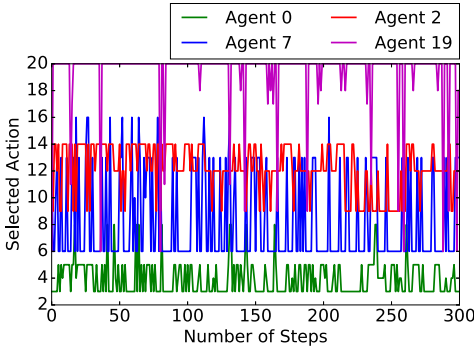


Fig. 27. Routing actions on four agents for *DeepIoTRouting*.

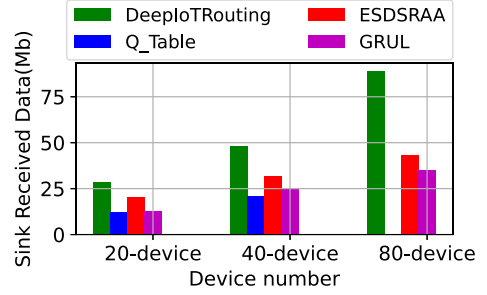


Fig. 28. Design space exploration.

energy budget on its own sensing and transmitting. Although devices 0 and 5 are also one-hop neighbors of the sink, their sensed data are slightly more than devices 3 and 12. This is because devices 3 and 12 complete most forwarding tasks so that devices 0 and 5 can have more energy to perform their own sensing and transmitting. For devices 2, 7, 8, 10, 14, 16, and 17, their received data is less than $\sim 0.5\text{Mb}$, since devices 8 and 16 are located at the corner of the network, so they rarely relay for others. Because of their unique topological location, they should not receive data. The unexpected received data on devices 2, 7, 8, 10, 14, 16, and 17 implies the dissipation of transmission energy and receiving energy. This can also cause the potential routing loop and deplete the hop budgets due to the wrong routing selection.

Moreover, we print out 300 consecutive actions for 4 EH IoT devices as shown in Figure 27 where the y-axis represents the EH IoT device ID of the destination device. The results in this figure show that the actions of the *DeepIoTRouting* agent can change in real-time. However, ESDSRAA makes decisions at the beginning of each hour, which means it cannot make decisions dynamically in real-time as the environment changes, which however can be fulfilled by *DeepIoTRouting*. *DeepIoTRouting* is able to dynamically fine-tune the amount of sensed data to prevent the network from being overloaded or having unbalanced sensing and communication. Therefore, it dramatically improves the energy efficiency and data delivery rate of the EH IoT system.

6.2.5 *DeepIoTRouting* Design Space Exploration. This section discusses the impacts of network scaling on the different algorithms. Figure 28 gives a glance at the amount of received data of the sink for four different algorithms at three different network scales. The EH IoT devices have more neighbors when the network scale, leading to a more intensive data relaying. Given the large-scale complicated topology and the more intensive data relaying, the *DeepIoTRouting* algorithm can achieve the sink delivery rate at 71.26%. This can be reflected by the great volume of received data 88.79 Mb for the 80-device EH IoT network, as shown in Figure 28. Besides, with the scaling of the network, the performance of *DeepIoTRouting* is superior compared to the other three algorithms. Figure 28 shows that when the size of the EH IoT system increases to 80 devices, the Q-table cannot learn the large-scale network because the number of states increases exponentially. Such a huge number of states can significantly hinder its convergence. In contrast, *DeepIoTRouting* has excellent scalability.

6.3 Experimental Analysis on Wind-Powered IoT Network

We also simulate the wind-powered IoT network under the same parameters setting, where the wind power traces are acquired from Scada Systems [2]. Because the average wind power intensity

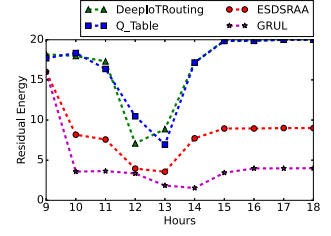
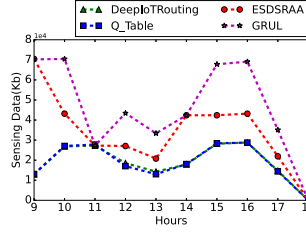
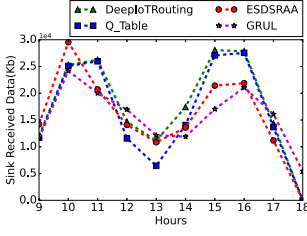


Fig. 29. Data delivery to sink in hours. Fig. 30. Sensed data in hours. Fig. 31. Residual energy in hours.

utilized in this article is greater than that of solar, we increase the sensing rate from 200 bps to 400 bps to enable the intermittent work pattern of EH IoT devices. Then, we evaluate the wind powered EH IoT network performance from the following three critical perspectives, (1) efficiency, (2) effectiveness, and (3) fairness.

Efficiency. We evaluate the proposed algorithm efficiency by measuring the delivery rate, the amount of sensed data, the amount of data delivery to the sink, and the execution time (computation overhead). Table 4 shows the wind-powered EH IoT network efficiency in one day after the *DeepIoTROUTING* being well-trained. As we can observe, the *DeepIoTROUTING* still can achieve the best performance (177.12 Mb Delivery Data) compared with all three baselines. Note that, since the average wind power intensity utilized in this article is stronger than that of solar, the network performance is improved under the wind power supply. Because the architecture of the neural network remains the same, the corresponding computation overhead is unchanged.

Effectiveness. We measure the residual energy, the amount of data delivered to the sink, and the amount of sensed data to evaluate the algorithms' effectiveness under the wind power scenarios. Figures 29–31 indicate the hourly network performance on the amount of data delivery to sink, the amount of sensed data, and the residual energy. Compared with the measurements in solar power scenarios, the data sensing and data forwarding operation activity in wind-powered IoT networks also follow energy fluctuation. Note that, even with a stronger wind power supply, in Figure 29, the *DeepIoTROUTING* can still outperform all baselines in terms of data delivery to the sink. What's more, in Figure 30, the amount of sensed data using ESDSRAA and GRUL are dramatically more than that of the *DeepIoTROUTING* and Q-table. This means ESDSRAA and GRUL waste even more harvested energy on "effortless" transmission where the majority of the sensed data packets are lost which leads to a low delivery rate. In practice, the wasted energy could be used to support other runtime operations. Since wind power is stronger than solar, as shown in Figure 31, after 15 : 00, the residual energy of each EH IoT device keeps at the full state because of the strong power supply.

Fairness. To evaluate the fairness, we measure the distribution of the amount of delivered data to sink from all EH IoT devices in Figure 32. Unlike Figure 22 with solar power, due to a stronger wind power supply, Figure 32 shows a more even distribution. As the green and blue bars indicated in Figure 32, the data distribution completed by Q-table is even better than the *DeepIoTROUTING*, yet the total amount of delivery data to the sink and the delivery rate completed by the Q-table is less than that of *DeepIoTROUTING*.

Together with experiments under solar power scenarios, we observe that the promising adaptability of *DeepIoTROUTING* outperforms all state-of-the-art related techniques under different practical power harvesting scenarios. In particular, when the ambient source power is weak such as the solar power utilized in this article, *DeepIoTROUTING* shows significant improvements over baseline methods. When the ambient source power is good such as the wind power utilized in this paper, *DeepIoTROUTING* can still show a marginal advantage over baseline methods.

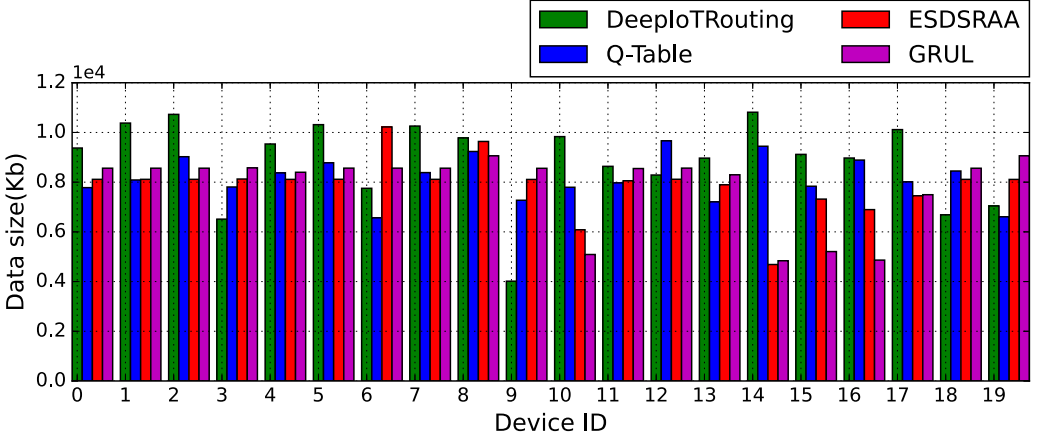


Fig. 32. Distribution of data delivery to sink (Daily).

7 CONCLUSION

In this article, we first mathematically formulated the problems and analyzed the topology influence and constraints on the overall performance of the multi-hop energy harvesting powered networks. Then, we developed a topology design and validation algorithm for better node deployment. After that, we investigated a network of EH IoT devices at different scales and jointly consider their routing and energy allocations. The experimental evaluation of the proposed distributed and scalable DRL-based approach *DeepIoTRouting* showed that *DeepIoTRouting* can significantly surpass the existing solutions by 126.48%, 38.71%, and 122.74% compared with *Q_table*, *ESDSRAA*, and *GRUL* in terms of data delivery to the sink. Besides, through the experiments, the proposed methods also demonstrate excellent scalability for being implemented in large IoT networks.

APPENDIX

A APPENDIX

The following is a proof of Theorem 1.

PROOF. Once the hardware of self-powered devices are selected, $\frac{(E_t+E_r)v_s T}{E_T+(E_r-E_s)v_s T}$ is a constant. Let $\Pi = \frac{(E_t+E_r)v_s T}{E_T+(E_r-E_s)v_s T}$, $\Pi \in [0, 1]$. We prove (7) as follows:

Based on (6), we have $\frac{M_h}{\sum_{l=1}^h M_l} \geq \Pi$. Thus,

$$M_h \geq \frac{\Pi}{1-\Pi}(M_1 + M_2 + \dots + M_{h-1}). \quad (16)$$

While $h = 2$, $M_2 \geq \frac{\Pi}{1-\Pi}M_1$.

While $h = 3$, $M_3 \geq \frac{\Pi}{(1-\Pi)}(M_1 + M_2)$.

Because we are computing the lower boundary of M_h , we can substitute M_2 with its minimum value, where $M_2 = \frac{\Pi}{1-\Pi}M_1$.

While $h = 3$, $M_3 \geq \frac{\Pi}{(1-\Pi)^2}M_1$ is true.

Similarly, for $M_h \geq \frac{\Pi}{(1-\Pi)}(M_1 + M_2 + \dots + M_{h-1})$, we substitute M_2 to M_{h-1} with the related M_1

term. The calculation can be given by (17) and (18).

$$M_h \geq \frac{\Pi}{1 - \Pi} \left(M_1 + \frac{\Pi}{(1 - \Pi)} M_1 + \frac{\Pi}{(1 - \Pi)^2} M_1 + \cdots + \frac{\Pi}{(1 - \Pi)^{h-2}} M_1 \right) \quad (17)$$

$$M_h \geq \frac{\Pi}{1 - \Pi} \left(\frac{1}{(1 - \Pi)} M_1 + \frac{\Pi}{(1 - \Pi)^2} M_1 + \cdots + \frac{\Pi}{(1 - \Pi)^{h-2}} M_1 \right)$$

$$M_h \geq \frac{\Pi}{1 - \Pi} \left(\frac{1}{(1 - \Pi)^2} M_1 + \cdots + \frac{\Pi}{(1 - \Pi)^{h-2}} M_1 \right)$$

$$M_h \geq \frac{\Pi}{(1 - \Pi)^{h-1}} M_1 \quad (18)$$

Substitute $\Pi = \frac{(E_t + E_r)v_s T}{E_T + (E_r - E_s)v_s T}$, $\Pi \in [0, 1]$ into (18). We can obtain (7) that approximates the number of devices in each layer. $\square$

REFERENCES

- [1] [n. d.]. Measurement and Instrumentation Data Center (MIDC). <https://midcdmz.nrel.gov/apps/sitehome.pl?site=ORNL>.
- [2] [n. d.]. The Wind Power Traces used to Power the Energy Harvesting IoT Devices. <https://www.kaggle.com/code/chittalpatel/wind-turbine-power-analysis>
- [3] Jamal N. Al-Karaki and Ahmed E. Kamal. 2004. Routing techniques in WSN: A survey. *IEEE Wireless Communications* 11, 6 (2004), 6–28.
- [4] Avik Banerjee, Anal Paul, and Santi Prasad Maity. 2017. Joint power allocation and route selection for outage minimization in multihop cognitive radio networks with energy harvesting. *IEEE Transactions on Cognitive Communications and Networking* 4, 1 (2017), 82–92.
- [5] Charles H. Bennett, Igor Devetak, Aram W. Harrow, Peter W. Shor, and Andreas Winter. 2014. The quantum reverse Shannon theorem and resource tradeoffs for simulating quantum channels. *IEEE Transactions on Information Theory* 60, 5 (2014), 2926–2959.
- [6] Mohammed Sulaiman BenSaleh, Raoudha Saidia, Yessine Hadj Kacem, and Mohamed Abid. 2020. Wireless sensor network design methodologies: A survey. *Journal of Sensors* 2020 (2020).
- [7] Toby Berger. 2003. Rate-distortion theory. *Wiley Encyclopedia of Telecommunications* (2003).
- [8] Petros S. Bithas, Emmanouel T. Michailidis, Nikolaos Nomikos, Demosthenes Vouyioukas, and Athanasios G. Kanatas. 2019. A survey on machine-learning techniques for UAV-based communications. *Sensors* 19, 23 (2019), 5170.
- [9] Jo Bito, Ryan Bahr, Jimmy G. Hester, Syed Abdullah Nauroze, Apostolos Georgiadis, and Manos M. Tentzeris. 2017. A novel solar and electromagnetic energy harvesting system with a 3-D printed package for energy efficient Internet-of-Things wireless sensors. *IEEE Transactions on Microwave Theory and Techniques* 65, 5 (2017), 1831–1842.
- [10] Guanrong Chen. 2022. Searching for best network topologies with optimal synchronizability: A brief review. *IEEE/CAA Journal of Automatica Sinica* 9, 4 (2022), 573–577.
- [11] GaoJun Fan and ShiYao Jin. 2010. Coverage problem in wireless sensor network: A survey. *Journal of Networks* 5, 9 (2010), 1033.
- [12] Kai-Wei Fan, Zizhan Zheng, and Prasun Sinha. 2008. Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks. In *6th ACM Conference on Embedded Network Sensor Systems*. 239–252.
- [13] Hannes Frey. 2004. Scalable geographic routing algorithms for wireless ad hoc networks. *IEEE Network* 18, 4 (2004), 18–22.
- [14] Neha Garg and Ritu Garg. 2017. Energy harvesting in IoT devices: A survey. In *2017 International Conference on Intelligent Sustainable Systems (ICISS '17)*. IEEE, 127–131.
- [15] Michael Gimelfarb, Scott Sanner, and Chi-Guhn Lee. 2020. $\{\epsilon\}$ -BMC: A Bayesian ensemble approach to epsilon-greedy exploration in model-free reinforcement learning. *arXiv preprint arXiv:2007.00869* (2020).
- [16] Drew Gislason. 2008. *Zigbee Wireless Networking*. Newnes.
- [17] Deniz Gunduz, Kostas Stamatou, Nicolo Michelusi, and Michele Zorzi. 2014. Designing intelligent energy harvesting communication systems. *IEEE Communications Magazine* 52, 1 (2014), 210–216.
- [18] Xiaoli He, Hong Jiang, Yu Song, Chunlin He, and He Xiao. 2019. Routing selection with reinforcement learning for energy harvesting multi-hop CRN. *IEEE Access* 7 (2019), 54435–54448.
- [19] Jae Hyun Kim and Jong Kyu Lee. 1999. Capture effects of wireless CSMA/CA protocols in Rayleigh and shadow fading channels. *IEEE Transactions on Vehicular Technology* 48, 4 (1999), 1277–1286.

- [20] Peng-Yong Kong and Yujae Song. 2019. Joint consideration of communication network and power grid topology for communications in community smart grid. *IEEE Transactions on Industrial Informatics* 16, 5 (2019), 2895–2905.
- [21] Hojoong Kwon, Hanbyul Seo, Seonwook Kim, and Byeong Gi Lee. 2009. Generalized CSMA/CA for OFDMA systems: Protocol design, throughput analysis, and implementation issues. *IEEE Transactions on Wireless Communications* 8, 8 (2009), 4176–4187.
- [22] Tim Leinmüller, Elmar Schoch, Frank Kargl, and Christian Maihöfer. 2005. Influence of falsified position data on geographic ad-hoc routing. In *European Workshop on Security in Ad-hoc and Sensor Networks*. Springer, 102–112.
- [23] Mo Li and Baijian Yang. 2006. A survey on topology issues in wireless sensor network. In *ICWN*. Citeseer, 503.
- [24] Xiaofang Li, Yingchi Mao, and Yi Liang. 2008. A survey on topology control in wireless sensor networks. In *2008 10th International Conference on Control, Automation, Robotics and Vision*. IEEE, 251–255.
- [25] Zitong Liu, Guoru Ding, Zheng Wang, Shilian Zheng, Jiachen Sun, and Qihui Wu. 2020. Cooperative topology sensing of wireless networks with distributed sensors. *IEEE Transactions on Cognitive Communications and Networking* 7, 2 (2020), 524–540.
- [26] Ting Lu, Guohua Liu, and Shan Chang. 2018. Energy-efficient data sensing and routing in unreliable energy-harvesting wireless sensor network. *Wireless Networks* 24, 2 (2018), 611–625.
- [27] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. 2019. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3133–3174.
- [28] Xiaolong Ma, Geng Yuan, Sheng Lin, Caiwen Ding, Fuxun Yu, Tao Liu, Wujie Wen, Xiang Chen, and Yanzhi Wang. 2020. Tiny but accurate: A pruned, quantized and optimized memristor crossbar framework for ultra efficient DNN implementation. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC '20)*. IEEE, 301–306.
- [29] Goutam Mali and Sudip Misra. 2015. TRAST: Trust-based distributed topology management for wireless multimedia sensor networks. *IEEE Trans. Comput.* 65, 6 (2015), 1978–1991.
- [30] Jaco Marais, Reza Malekian, Ning Ye, and Ruchuan Wang. 2016. A review of the topologies used in smart water meter networks: A wireless sensor network application. *Journal of Sensors* 2016 (2016).
- [31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
- [32] Miguel Morales. 2020. *Grokking Deep Reinforcement Learning*. Manning Publications.
- [33] Thien Duc Nguyen, Jamil Yusuf Khan, and Duy Trong Ngo. 2018. A distributed energy-harvesting-aware routing algorithm for heterogeneous IoT networks. *IEEE Transactions on Green Communications and Networking* 2, 4 (2018), 1115–1127.
- [34] Andrea Ortiz, Hussein Al-Shatri, Xiang Li, Tobias Weber, and Anja Klein. 2017. Reinforcement learning for energy harvesting decode-and-forward two-hop communications. *IEEE Transactions on Green Communications and Networking* 1, 3 (2017), 309–319.
- [35] S. Pallavi, Jayashree D. Mallapur, and Kirankumar Y. Bendigeri. 2017. Remote sensing and controlling of greenhouse agriculture parameters based on IoT. In *2017 International Conference on Big Data, IoT and Data Science (BIG '17)*. IEEE, 44–48.
- [36] Chen Pan, Mimi Xie, Song Han, Zhi-Hong Mao, and Jingtong Hu. 2019. Modeling and optimization for self-powered non-volatile IoT edge devices with ultra-low harvesting power. *ACM Transactions on Cyber-Physical Systems* 3, 3 (2019), 1–26.
- [37] Chen Pan, Mimi Xie, and Jingtong Hu. 2018. 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* 37, 11 (2018), 2440–2450.
- [38] Chen Pan, Mimi Xie, Yongpan Liu, Yanzhi Wang, Chun Jason Xue, Yuangang Wang, Yiran Chen, and Jingtong Hu. 2017. A lightweight progress maximization scheduler for non-volatile processor under unstable energy harvesting. *ACM SIGPLAN Notices* 52, 5 (2017), 101–110.
- [39] Patricia Ruiz, Bernabé Dorronsoro, Pascal Bouvry, and Lorenzo Tardón. 2012. Information dissemination in VANETs based upon a tree topology. *Ad Hoc Networks* 10, 1 (2012), 111–127.
- [40] Ashutosh Sabharwal, Ahmad Khoshnevis, and Edward Knightly. 2007. Opportunistic spectral usage: Bounds and a multi-band CSMA/CA protocol. *IEEE/ACM Transactions on Networking* 15, 3 (2007), 533–545.
- [41] Stanislav Safaric and Kresimir Malaric. 2006. ZigBee wireless standard. In *ELMAR 2006*. IEEE, 259–262.
- [42] Claude E. Shannon. 1961. Two-way communication channels. In *4th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, Vol. 4. University of California Press, 611–645.
- [43] Navin Sharma, Jeremy Gummesson, David Irwin, and Prashant Shenoy. 2010. Cloudy computing: Leveraging weather forecasts in EH sensor systems. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10)*. IEEE, 1–9.

- [44] Dimitrios Sikeridis, Eirini Eleni Tsiropoulou, Michael Devetsikiotis, and Symeon Papavassiliou. 2018. Energy-efficient orchestration in wireless powered internet of things infrastructures. *IEEE Transactions on Green Communications and Networking* 3, 2 (2018), 317–328.
- [45] Dimitrios Sikeridis, Eirini Eleni Tsiropoulou, Michael Devetsikiotis, and Symeon Papavassiliou. 2018. Wireless powered public safety IoT: A UAV-assisted adaptive-learning approach towards energy efficiency. *Journal of Network and Computer Applications* 123 (2018), 69–79.
- [46] Mehdi Sookhak, Adnan Akhundzada, Alireza Sookhak, Mohammadreza Eslaminejad, Abdullah Gani, Muhammad Khurram Khan, Xiong Li, and Xiaomin Wang. 2015. Geographic wormhole detection in wireless sensor networks. *PLoS One* 10, 1 (2015), e0115324.
- [47] Gyorgy D. Szarka, Bernard H. Stark, and Stephen G. Burrow. 2011. Review of power conditioning for kinetic energy harvesting systems. *IEEE Transactions on Power Electronics* 27, 2 (2011), 803–815.
- [48] Marc Tafforeau, Marie-Claire Verdu, Vic Norris, Glenn J. White, Mike Cole, Maurice Demarty, Michel Thellier, and Camille Ripoll. 2004. Plant sensitivity to low intensity 105 GHz electromagnetic radiation. *Bioelectromagnetics: Journal of the Bioelectromagnetics Society, The Society for Physical Regulation in Biology and Medicine, The European Bioelectromagnetics Association* 25, 6 (2004), 403–407.
- [49] Shengda Tang and Liansheng Tan. 2016. Reward rate maximization and optimal transmission policy of EH device with temporal death in EH-WSNs. *IEEE Transactions on Wireless Communications* 16, 2 (2016), 1157–1167.
- [50] Omkar Thakoor, Jugal Garg, and Rakesh Nagi. 2019. Multiagent UAV routing: A game theory analysis with tight price of anarchy bounds. *IEEE Transactions on Automation Science and Engineering* 17, 1 (2019), 100–116.
- [51] Zengfu Wang, Qing Wang, Bill Moran, and Moshe Zukerman. 2020. Optimal submarine cable path planning and trunk-and-branch tree network topology design. *IEEE/ACM Transactions on Networking* 28, 4 (2020), 1562–1572.
- [52] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning* 8, 3-4 (1992), 279–292.
- [53] Paul Wright, Andrew Lord, and Savvas Nicholas. 2012. Comparison of optical spectrum utilization between flexgrid and fixed grid on a real network topology. In *OFC/NFOEC*. IEEE, 1–3.
- [54] Aaron Wyner. 1974. Recent results in the Shannon theory. *IEEE Transactions on Information Theory* 20, 1 (1974), 2–10.
- [55] Yang Xiao, Jun Liu, Jiawei Wu, and Nirwan Ansari. 2021. Leveraging deep reinforcement learning for traffic engineering: A survey. *IEEE Communications Surveys & Tutorials* (2021).
- [56] Mimi Xie, Mengying Zhao, Chen Pan, Jingtong Hu, Yongpan Liu, and Chun Jason Xue. 2015. Fixing the broken time machine: Consistency-aware checkpointing for energy harvesting powered non-volatile processor. In *P 52nd Annual Design Automation Conference*. 1–6.
- [57] Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. 2004. On greedy geographic routing algorithms in sensing-covered networks. In *5th ACM International Symposium on Mobile ad hoc Networking and Computing*. 31–42.
- [58] Xinyu You, Xuanjie Li, Yuedong Xu, Hui Feng, Jin Zhao, and Huaicheng Yan. 2020. Toward packet routing with fully distributed multiagent deep reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2020).
- [59] Wen Zhang, Tao Liu, Mimi Xie, Jun Zhang, and Chen Pan. 2021. Sac: A novel multi-hop routing policy in hybrid distributed iot system based on multi-agent reinforcement learning. In *22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 129–134.
- [60] Wen Zhang, Tao Liu, Mimi Xie, Longzhuang Li, Dulal Kar, and Chen Pan. 2022. EH aware multi-hop routing policy in distributed IoT system based on multi-agent reinforcement learning. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC '22)*. IEEE, 562–567.

Received 23 September 2022; revised 10 July 2023; accepted 20 December 2023