Hindawi
Mobile Information Systems
Volume 2021, Article ID 5589145, 12 pages
https://doi.org/10.1155/2021/5589145

Research Article

EER-RL: Energy-Efficient Routing Based on Reinforcement Learning

Vially Kazadi Mutombo, Seungyeon Lee, Jusuk Lee, and Jiman Hong

Department of Computer Science and Engineering, Soongsil University, Seoul 06978, Republic of Korea

Correspondence should be addressed to Jiman Hong; jiman@ssu.ac.kr

Received 20 February 2021; Revised 13 March 2021; Accepted 2 April 2021; Published 20 April 2021

Academic Editor: Hoon Ko

Wireless sensor devices are the backbone of the Internet of things (IoT), enabling real-world objects and human beings to be connected to the Internet and interact with each other to improve citizens' living conditions. However, IoT devices are memory and power-constrained and do not allow high computational applications, whereas the routing task is what makes an object to be part of an IoT network despite of being a high power-consuming task. Therefore, energy efficiency is a crucial factor to consider when designing a routing protocol for IoT wireless networks. In this paper, we propose EER-RL, an energy-efficient routing protocol based on reinforcement learning. Reinforcement learning (RL) allows devices to adapt to network changes, such as mobility and energy level, and improve routing decisions. The performance of the proposed protocol is compared with other existing energy-efficient routing protocols, and the results show that the proposed protocol performs better in terms of energy efficiency and network lifetime and scalability.

1. Introduction

The emergence of wireless technologies and information systems and mobile technologies has opened up a new era for the Internet of things (IoT). The latter has become the backbone for ubiquitous computing while enabling the environment to be smart through recognition, identification of objects, data generation, transmission, and retrieval [1, 2]. IoT allows real-world things and people to be connected and be part of the virtual world of the Internet through wireless communication. Initially, IoT has been targeted to the network of RFID tags and later it has been broadly extended to various devices and applications with the goal to first make objects capable of learning and understanding their environment and interact with it [3]. Through wireless communication, these objects can interact with each other and enable the system to be remotely controlled via Internet connection [2–5]. Due to its implications to various fields, IoT has recently received much attention, and it has been applied to a wide range of applications such as smart cities, smart healthcare systems, smart homes, object tracking, disaster management, and environmental monitoring [6, 7].

IoT consists of the interconnection of heterogeneous wireless devices including smartphones, wireless sensors, actuators, identification by radio frequency (RFID) tags, and real-world things with sensing capabilities [8, 9]. Generally, a sensor device comprises four units, namely, power unit, sensing unit, processing unit, and communication unit [10–13]. The sensing unit is responsible for sensing data from the surrounding environment, whereas the processing unit carries out the computation tasks. The communication unit is in charge of sending packets across the network. Finally, the power unit consists of a small battery that supplies power to the remaining three modules. Logically, the power unit does not consume any energy but supplies energy to other modules, the sensing module and processing module also consume negligible energy, whereas the communication module is the most energy-consuming [14–16].

However, to accommodate a large number of devices in an IoT, several requirements are needed including energy efficiency, scalability, interoperability, security, and flexibility [2]. Energy efficiency is crucially important to maintain a fully operational network for the most prolonged time possible [16, 17], especially for devices deployed in a harsh environment where recharging and replacing the battery are impossible. Hence, energy-efficient routing protocols are known to manage the consumption of devices' available energy and extend the lifetime of the network [6, 13].

Reinforcement learning is a subfield of machine learning that solves the problem of an agent that takes actions in an unknown environment and improves over time through a sequence of trial-and-error interactions with the environment [18]. In other words, the agent interacts with the environment by performing actions and gets rewards, which can be either positive when the action performed was right or negative otherwise. This approach has brought dynamism in data routing and adaptation capability in network communication compared to static routing approaches [19–21]. In IoT networks, RL can be used to deal with problem such as network topology change due to mobility of devices, energy level, and other transmission parameters such as distance, signal strength, and bandwidth, which can change over time and influence the network performance.

In this paper, we propose EER-RL, an energy-efficient routing protocol for IoT based on reinforcement learning. The proposed EER-RL balances the energy dissipation between devices in an IoT network and extends the network lifetime and improves the network scalability. EER-RL also provides optimal paths using a feedback mechanism to share local information as a reward, the latter is computed using the residual energy and hop count to the sink, and the hop count parameter can reduce the end-to-end delay. To evaluate the performance of EER-RL, we carried out simulations and the results show that EER-RL achieves an efficient energy consumption, extends the network lifetime, and is more scalable for large-scale IoT networks. EER-RL is also compared to LEACH [22] and PEGASIS [23], the comparison results show that EER-RL outperformed them by providing a better energy balance and extending the network lifetime.

The remainder of this paper is organized as follows: in Section 2, we give an overview of RL and its application in routing. Section 3 discusses the existing solutions on routing protocols using RL. Section 4 describes our proposed solution. The performance evaluation of the proposed solution is presented in Section 5, followed by the conclusion remarks and future work in Section 6.

2. Overview on Reinforcement Learning

RL problems are formalized as Markov decision processes (MDP) with a tuple (S, A, P, R), where S represents a set of states an agent can be in at a given time t; A is a set of possible actions an agent can take. The transition probability that an agent at a given time t, and from a given state $s(t)$ which performs an action $a(t)$ to enter in state $s(t+1)$, is denoted as P, and R is the reward obtained by the agent for the action performed [18]. Applying RL to routing protocols requires defining the main components of an RL model, such as agent and environment, state and action, and reward. First, the agent is the decision-maker of an RL model while the environment is what the agent observes and reacts to it. In IoT networks, every device is considered an agent; for the whole network, multiagent RL is required. Secondly, a state is any useful information about the environment at a given time, whereas an action is the agent's reaction in a given state. The state space for an agent is the available routing information from all available neighbouring devices. A state can be a tuple of decision-making factors such as residual energy, hop count, and signal strength, depending on the factors taken into account while designing the protocol. On the other hand, an action refers to selecting the next-hop to route the packet towards the base station. Thus, the action space represents the set of all available routes through neighbours at a given time. Thirdly, the cost of the action performed by the agent in a given state is called a reward.

The following definitions are considered for the implementation of the proposed protocol: (1) every device in the network is considered an agent, and (2) for each device, the set of available routes through its neighbouring devices to the base station is the state space. (3) The set of all available neighbours through which packets can be sent to the base station is denoted as the action space. (4) The agent's behaviour is denoted as a policy.

A policy maps the state-action pair; it can be stochastic or deterministic and improves over time. The goal of every RL model is to find an optimal policy that maximizes the long-term reward of each state-action pair [18, 19]; this goal can be reached using the policy iteration process, which includes evaluating and improving the given policy. The policy evaluation evaluates the policy from the results while policy improvement improves the policy towards the best one [18, 20]. Figure 1 depicts a simple RL model.

3. Related Work

RL in network routing protocol was first introduced by Littman and Boyan in [24], who proposed the Q-routing algorithm, a delivery time-optimal solution that aimed to increase the packet delivery ratio while minimizing the average delivery time. Its experimental results outperformed the shortest path routing protocol in packet delivery time, especially in heavy network loads. However, it could suffer from poor learning policy, as the agents did not update their information about the environment. Consequently, the selected routes could not be reliable; also, Q-routing did not consider devices' residual energy. Thus, it did not guarantee energy efficiency.

Recently, significant research efforts have been made to apply RL to wireless sensor networks (WSN) and IoT. Several approaches were proposed, including the cooperative approach, which is one of the most used in many works, such as SSAR in [8], FROMS in [25], OPT-Q-Routing in [26], EQR-RL in [27], and the work by [28]. This approach is deemed to be suitable for multiagent RL models where agents are required to cooperate and work together for the same purpose. For instance, for self-organized wireless

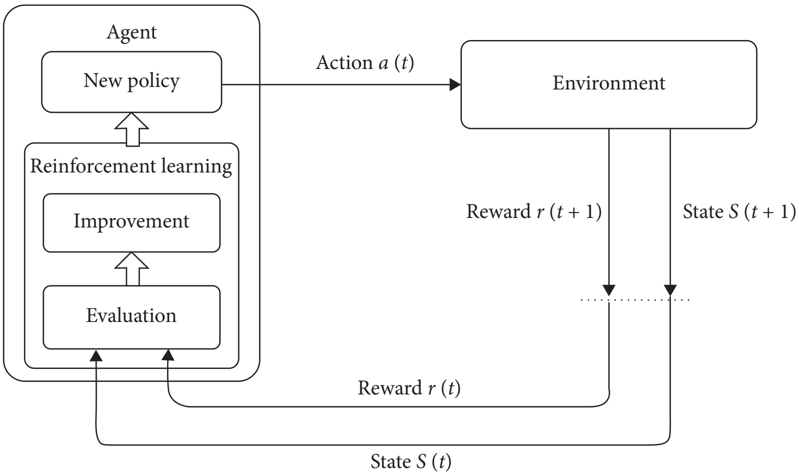


Figure 1: Simple reinforcement learning model.

networks [29, 30], it allows devices to communicate with each other by sharing local information such as position coordinates, hop count to the base station, initial energy, residual energy, transmission range, and signal strength. These considerations, taken alone or combined based on some policies, allow devices to make better decisions on the next-hop to route the data. However, due to devices' resource-constrained nature, the residual energy is a crucial factor for many energy-efficient routing protocols; it is usually combined with others to optimize the routing protocol's performance.

In [25], feedback routing for optimizing multiple sinks (FROMS) used a feedback mechanism to share local information as reward, i.e., the receiver of a packet shares its local information as feedback to the sender without extra network overhead. FROMS also provided a recovery mechanism after device failure to deal with packet loss. However, this protocol could suffer from packet loss in sink mobility, especially in high-speed mobility, and this problem triggers routing errors and extra energy consumption.

In [26], OPT-Q-Routing, an energy balancing routing protocol based on RL, optimized the multihop communication and extended the network lifetime by balancing the devices' energy dissipation and reducing the network overhead. However, they considered only the residual energy of devices; thus, the proposed protocol did not ensure a good balance for multihop communication in the long run. In [27], devices periodically broadcasted heartbeat packets that include the delivery ratio estimate and the sender's residual energy using EQR-RL. This information enables each device to compute the next-hop using a probability function. The simulation results showed a good delivery ratio and low end-to-end delay while supporting sink mobility. However, network isolation is likely to happen due to the avoidance of some devices.

In [8], the authors proposed a smart and self-organizing routing algorithm (SSAR), which selects the best route based on some communication parameters, namely, the distance between nodes, the stability of links, and residual energy of IoT devices. These parameters have an impact on the stability and energy-efficient routing. SSRA achieved a good QoS expressed in packet delivery and end-to-end delay; it also extended the network lifetime compared to other existing algorithms. In [28], the authors added the bandwidth to the three parameters considered in [8] and used fuzzy logic and RL to train network devices for optimal route selection. Their proposed algorithm performed better in terms of energy efficiency and network lifetime.

4. EER-RL Protocol

This section describes the proposed EER-RL protocol, a cluster-based energy-efficient routing protocol for IoT wireless networks using RL. EER-RL allows devices to learn how to make better routing decisions by sharing local information with the neighbourhood to optimize the next-hop selection and minimize energy consumption. The sender includes its local information in the packet header, every device in the neighbourhood that can overhear the packet, extracts the information encapsulated in the packet header, and updates its routing table. The local information shared includes device id, residual energy, position coordinates, and hop count. Like other cluster-based routing protocols, EER-RL consists of three steps: network set-up and cluster head election, cluster formation, and data transmission.

4.1. Network Set-Up and Cluster Head Election. This phase consists of two steps: first, the network set-up allows devices to compute the initial Q-value based on their local information. Initially, the base station broadcasts a heartbeat message where it shares its position coordinates. Every device, after receiving the packet from the base station, saves the latter's position and computes the initial Q-value using the initial energy level and the hop count using equations (1) and (2). Furthermore, we assume that all the devices have different energy levels and we also define a distance threshold between cluster heads (CHs) and the base station to mitigate network overhead and facilitate sensors far from the base stations to find a CH easily. Besides, to avoid links to diverge to the base station rather than converging, a CH cannot be at the edge of the network as this can result in energy wastage due to the extension of the communication distance. Algorithm 1 describes the CH election process.

$$Q = \begin{cases} \dfrac{1}{N_h}, & \text{if } E_{\min} = E_{\max}, \\ p \times \left(\dfrac{E_r - E_{\min}}{E_{\max} - E_{\min}} \right) + (1 - p) \times \dfrac{1}{N_h}, & \text{if } E_{\min} \neq E_{\max}, \end{cases} \tag{1}$$

$$N_h \cong \frac{D_{\text{link}}}{\text{TX}_{\text{range}}}. \tag{2}$$

After the cluster heads' election phase, every CH sends an invitation message to inform all the devices within its transmission range that it has been elected as CH, the invitation also includes the CH Id, its initial Q-value, and its location coordinates. Every non-CH device that overhears packets from CHs decides which cluster to join depending on the distance and sends a membership request to the designed CH, including its local information. Additionally, if a device receives more than one invitation, i.e., the device is at the intersection of different clusters, it can decide to join the one whose CH is the closest. Once all the devices send membership requests to CHs, every CH confirms the membership and forms the cluster. Algorithm 2 describes the cluster formation process.

Devices with the base station in their transmission range do not need to join any cluster [4], but they can communicate directly with the base station and save more energy. Another objective of the EER-RL protocol is to provide an energy-efficient multihop scheme for intercluster and intracluster communication. Intercluster communication is about managing the multihop communication between CHs; for instance, a CH far from the base station can send packets through an intermediate CH, or if there is a powerful device next to the base station, it can also be used to aggregate data from that CH. Thus, CHs far from the base station can communicate in multihop through either other CHs or powerful devices next to the base station.

Similarly, for intracluster communication, devices within the cluster can also send packets directly to the CH or in multihop if they are far from the CH; in other words, a device far away from the CH can communicate with the latter through another device in the same cluster. As mentioned earlier, IoT devices have different energy levels and transmission range; thus, we assume that devices with high energy levels also have a more extended transmission range than those with low energy levels. Figure 2 depicts the cluster formation.

4.2. Data Transmission: Application of RL.

The application of RL intervenes at this phase, where every device behaves like a learning agent and learns to make better routing decisions. The learning process consists of updating the Q-value using the immediate reward obtained from the action performed and finding the best policy that optimizes the long-term reward. We use three functions, namely, the energy consumption model, the reward function, and the update function for Q-value. The energy consumption model allows updating the residual energy by subtracting the energy dissipated after every packet transmission; the updated residual energy and the hop count are then used to evaluate the reward. Finally, the reward is then used as an argument in the Q-value function to update the Q-value.

4.2.1. Energy Consumption Model.

After a packet transmission, the sender and receiver both consume energy; unlike the receiver, the sender consumes more energy to send packets over the network and amplify the signal over the distance. The energy consumption model computes the energy which is dissipated at packet transmission or reception and updates the residual energy. The energy consumption model is presented in the following equation [22]:

$$\begin{cases} E_{\text{Tx}}(k, d) = E_{\text{elec}} \times k + E_{\text{amp}} \times k \times d^m, \\ E_{Rx}(k) = E_{\text{elec}} \times k, \end{cases} \tag{3}$$

where $E_{\text{Tx}}(k, d)$ and $E_{\text{Rx}}(k)$ are the energy consumed by the transmitter and receiver, respectively.

At each transmission (or reception), E_{elec} estimated at $50\, nJ/bit$ is the energy consumed to run the transmitter or receiver circuit, whereas amplification energy estimated at $100\, pJ/bit/m2$ is the energy consumed to amplify the signal over the distance, with $m = 2$ or 4 depending upon the distance.

4.2.2. Reward Function.

As mentioned earlier, the learning agent gets compensation denoted as a reward for every action performed; the reward is the cost of an action helping the agent to know whether the action performed was good or bad. In EER-RL, the action refers to selecting a neighbour as the next-hop to route the packet. The reward function is computed using the residual energy (E_r) and hop count N_h, unlike the link distance used in the previous phase to compute the hop count, the distance between the sender and each neighbour is also considered at this phase. For instance, the distance of a device S_i to the base station via an intermediate device S_j is denoted as D_{link}, computed as in equations (4)–(6). This distance D_{link} can also equivalent to $N_h \times \text{TX}_{\text{range}}$, where N_h denoted as the hop count and TX_{range} is the transmission range [9]. From the above, the estimated hop count is computed as in equation (2), and the reward is computed as in equation (7).

```
For $i \longleftarrow n$, do
    $S(i)$.dist $\longleftarrow$ Euclidean($S(i)$, sink)
    $S(i)$.hop $\longleftarrow$ ($S(i)$.dist/$\mathrm{TX}_{\mathrm{range}}$)
    $S(i).Q \longleftarrow p \times (S(i).E - E_{\min}/E_{\max} - E_{\min}) + (1 - p) \times (1/S(i).\mathrm{hop})$
End for
While length $(\mathrm{CH}_{\mathrm{table}}) \leq \mathrm{CH}_{\mathrm{tot},}$ do
    $\mathrm{Q}_{\max} \longleftarrow \max(S.Q)$
    For $i \longleftarrow 1$ to $n$, do
        If $\mathrm{Min}_{\mathrm{Thres}} \leq S(i)$.dist $< \mathrm{Max}_{\mathrm{Thres}}$, then
            If length ($\mathrm{CH}_{\mathrm{Table}}$ == 0), then
                $\mathrm{CH}_{\mathrm{Table}}$.append ($S(i)$)
                $S$.pop ($S(i)$)
            Else
                For $h \longleftarrow 1$ to length ($\mathrm{CH}_{\mathrm{Table}}$), do
                    dts = Euclidean ($S(i)$, CH ($h$))
                    If dts $\geq \mathrm{Min}_{\mathrm{Thres}}$, then
                        $C \longleftarrow$ True
                    Else
                        $C \longleftarrow$ False
                        Break
                    End if
                End for
                If $C$ = = True, then
                    $\mathrm{CH}_{\mathrm{Table}}$.Append ($S(i)$)
                    $S$.pop ($S(i)$)
                End if
            End if
        End if
    End for
End while
```

Algorithm 1: Network set-up and cluster head election algorithm.

```
For $i \longleftarrow 1$ to $\mathrm{CH}_{\mathrm{tot},}$ do
    For $j \longleftarrow 1$ to $n$, do
        dtsCh ($j, i$) $\longleftarrow$ Euclidean ($S(j)$, CH ($i$))
        If dtsCh ($j, i$) $\leq$ CH ($i$).$\mathrm{TX}_{\mathrm{range}}$, then
            CH ($i$).send Invitation ($S(j)$)
        End if
    End for
End for
For $j \longleftarrow 1$ to $n$, do
    If $S(j)$.dts $\leq \mathrm{TX}_{\mathrm{range}}$, then
        $S(j)$.dest $\longleftarrow$ sin$k$
    Else
        For $i \longleftarrow 1$ to $\mathrm{CH}_{\mathrm{tot},}$ do
            If Invitation ($j, i$) $\neq$ Emply, then
                If dtsCh ($j, i$) $\leq$ min (dtsCh ($j$, : )), then
                    $S(j)$.dest $\longleftarrow$ CH ($i$).$i\ d$
                    Create Neigh (j,:)
                    Cluster ($i$).append ($S(j)$.Id)
                End if
            End if
        End for
    End if
End for
```

Algorithm 2: Cluster formation.

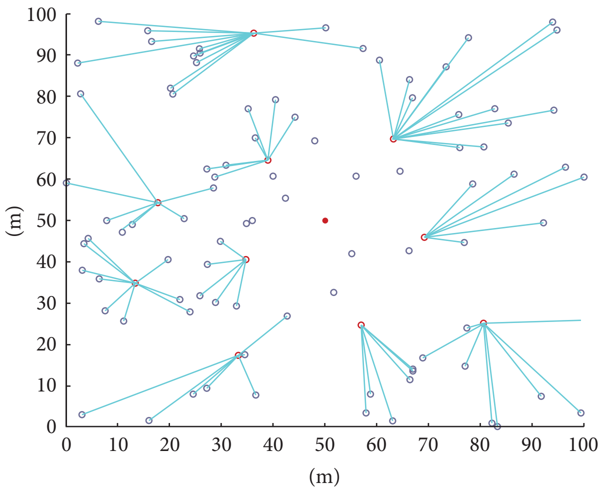


Figure 2: Cluster formation using 100 devices scattered along a sensing field of 100 × 100 m of size.

$$D_{\text{link}} = D_{i,j} + D_{j,\text{sink}}, \tag{4}$$

$$D_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{5}$$

$$D_{j,\text{sink}} = \sqrt{\left(x_j - x_{\text{sink}}\right)^2 + \left(y_j - y_{\text{sink}}\right)^2}, \tag{6}$$

$$r_{t+1} = \begin{cases} \dfrac{1}{N_h}, & \text{if } E_{\min} = E_{\max}, \\ p \times \left(\dfrac{E_r - E_{\min}}{E_{\max} - E_{\min}}\right) + (1-p) \times \dfrac{1}{N_h}, & \text{if } E_{\min} \neq E_{\max}, \\ -100, & \text{if } E_r < 0, \end{cases} \tag{7}$$

where $0 f p f 1$ is the probabilistic variable, which defines the impact of the E_r in contrast to N_h.

A high value of p gives a high probability to the neighbour with a high energy level to be selected as next-hop, whereas a high value of $q = 1 - p$ gives to the closest neighbour a high probability to be selected. Therefore, a trade-off between p and q is required to optimize the performance of the protocol; if E_r is null(0) in this case, S_i will assign a negative reward to S_j and select another next-hop device. Similarly, S_j also repeats the same process when forwarding the packet to the next-hop and sends feedback to S_i, and the process goes on until the packet reaches the destination. The reward is encapsulated in the packet header to avoid the network overhead, all neighbourhood devices can overhear the packet forwarded and update their routing table. For instance, if S_i overhears a packet that it sent earlier, it will also extract the reward as feedback.

4.2.3. *Function for Updating Q-Value.*

To learn the real cost of an action, we have to compute the action-value function, which defines how good it is to perform an action from a given state following a policy π [18, 19]. The action-value function is the expectation of the discounted sum of returns given a state and an action as presented in the following equation:

$$Q_\pi(s,a) = \mathbb{E}\left[G_t \mid S_t = s,\ A_t = a\right], \tag{8}$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k \times R_{t+k+1}. \tag{9}$$

In [31], Watkins and Dayan proposed an approach to estimate the action-value function (or Q-function). His approach is a model-free (the opposite of model-based, model-free system does not need any environment model at all, i.e., agents cannot think about how their environments will change in response to a single action [18]) learning technique called Q-learning. The action-value function approximation proposed by Watkins depends on the policy followed by the agent, which makes Q-learning easy to implement and applicable in many situations [19]. Thus, we have

$$Q_{\pi *} = Q^*(s,a), \tag{10}$$

$$V^*(s,a) = \max_a (Q(s,a)). \tag{11}$$

In Q-learning, the initial Q-value can be set to a random fixed value or computed using some arguments, and this is implementation-dependent. In this work, the initial Q-value is calculated using the initial energy and hop count as described in equation (1). Then, for each action performed, the agent gets a reward and uses it to update the value of Q using the following equation:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left(r_{t+1}(s,a) + \gamma \max_a Q(S',a)\right), \tag{12}$$

where α is the learning rate, which in most cases is set to 1 to speed up the learning process, and $r_{t+1}(s,a)$ is the immediate reward computed using equation (7).

The policy used is such that the sender selects the neighbour with the highest Q-value denoted as $\max_a Q(S',a)$, to maximize the reward, and end up in a state S'. The discount factor γ varies between 0 and 1; it defines the importance of the long-term rewards against the immediate one. If γ approaches 1, it means that the agent emphasizes the future reward rather than the immediate reward. Therefore, most RL-based protocols set a value that approaches 1 to give high importance to future reward. In the EER-RL as well, we use $\gamma = 0.95$. However, a discount factor of zero (0) means that the agent is more concerned only with maximizing the immediate reward.

Algorithm 3 describes the data transmission process. First, devices with less than or equal to zero residual energy are considered dead, thus cannot transmit data. However, devices with the highest Q-value in their neighbourhood and devices with the base station in their transmission range can communicate directly with the base station without any intermediate device. On the other hand, devices far from the base station send packets through the CH, but if the CH is also far, they can send to the CH through another closest device member of the same cluster.

```
For $i \leftarrow 1$ to $n$, do
    If $S(i).E > 0$, then
        $\max Q = \max(Q(i,:))$
        If $S(i).d \leq \mathrm{TX}_{\mathrm{range}}$
            If $S(i)$ is next-hop, then
                Aggregate data
                Send data to sink
            Else
                Send data to sink
            End if
        Else if $S(i).\mathrm{role} == 0$, then
            If CH within $\mathrm{TX}_{\mathrm{range}}$, then
                Send data to CH
            Else
                Find closest neighbour in the cluster
                Send data to closest neighbour
            End if
        End if
        Compute reward
        Update Q-value
    End if
End for
```

Algorithm 3: Data transmission.

5. Performance Evaluation

To evaluate the performance of the proposed EER-RL, we conducted simulation using MATLAB and deployed 100 devices distributed randomly over a sensing field of 100 × 100 m following the normal distribution [32]. The base station was placed in the middle of the sensing field with (50, 50) coordinates. Furthermore, we assumed the network to be heterogeneous, including devices with different energy levels ranging from 1 to 2 joules. The simulation parameters are presented in Table 1.

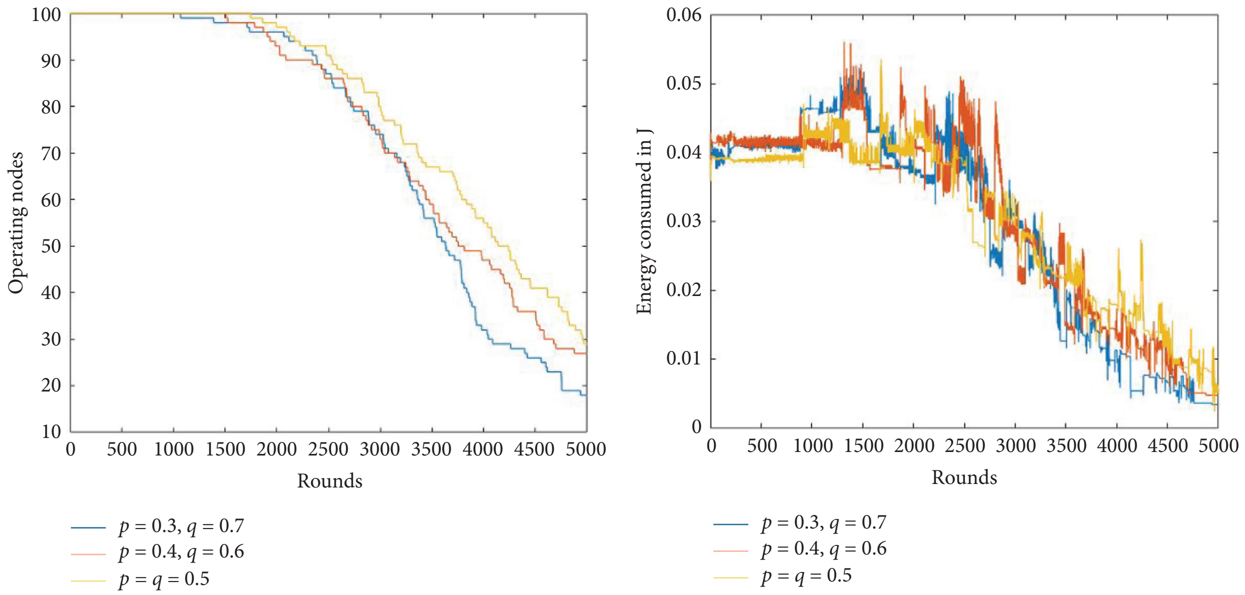
Table 1: Simulation parameters.

Parameters	Values
Sensing field size	100×100 m
Number of devices	[30–100]
Transmission range	20 m
Initial energy	[1-2] joules
Data size	4000 bits
$\mathrm{E}_{\mathrm{elec}}$	50×10^{-9} joules/bit
E_{amp}	100×10^{-12} joules/bit/m^2
α	1
γ	0.95

5.1. Parameter Tuning. As mentioned above, the proposed protocol takes the hop count and residual energy into consideration, and some probabilistic parameters such as p and $q = 1-p$ were assigned to both residual energy and hop count, respectively. A high value of p gives the devices with high energy levels a high probability to be selected. Similarly, a high value of q increases the probability of devices with less hop count to the base station to be selected. Therefore, to optimize the performance of EER-RL, we tested different values of these parameters to select the best ones. The performance evaluation showed slightly different results with different values of p and q. However, with p and q equal to 0.5, the network lifetime was extended more while keeping a good energy balance. In some cases, $p = 0.4$ and $q = 0.6$ can also give similar results. Figure 3 shows the performance results of EER-RL using different values of p and q.

5.2. Comparison between EER-RL and FlatEER-RL. Through comparison, we highlight the difference between the proposed cluster-based protocol EER-RL and its flat-based version denoted as FlatEER-RL. The first difference is that the flat-based protocol consumes more energy at the beginning until the learning process finishes. After the learning process, the next-hop selection is then optimized. In contrast, the cluster-based protocol assigns to devices far from the base station, the CH as the next-hop by default, which speeds up the learning process and optimizes the energy consumption from the beginning. However, in the cluster-based protocol, CHs consume more energy for data aggregation, resulting in high-energy consumption per round. While with the flat-based, after the learning process, the energy consumed per round is balanced between the devices. Thus, the time until the first device dies is slightly extended compared to the FlatEER-RL. Therefore, to balance the energy consumption in EER-RL, we set an energy threshold for CHs, and when a CH reaches the threshold, it needs to be replaced before it gets completely depleted. This approach extends the network lifetime when using EER-RL compared to FlatEER-RL. Figure 4 shows the results of both EER-RL and FlatEER-RL.

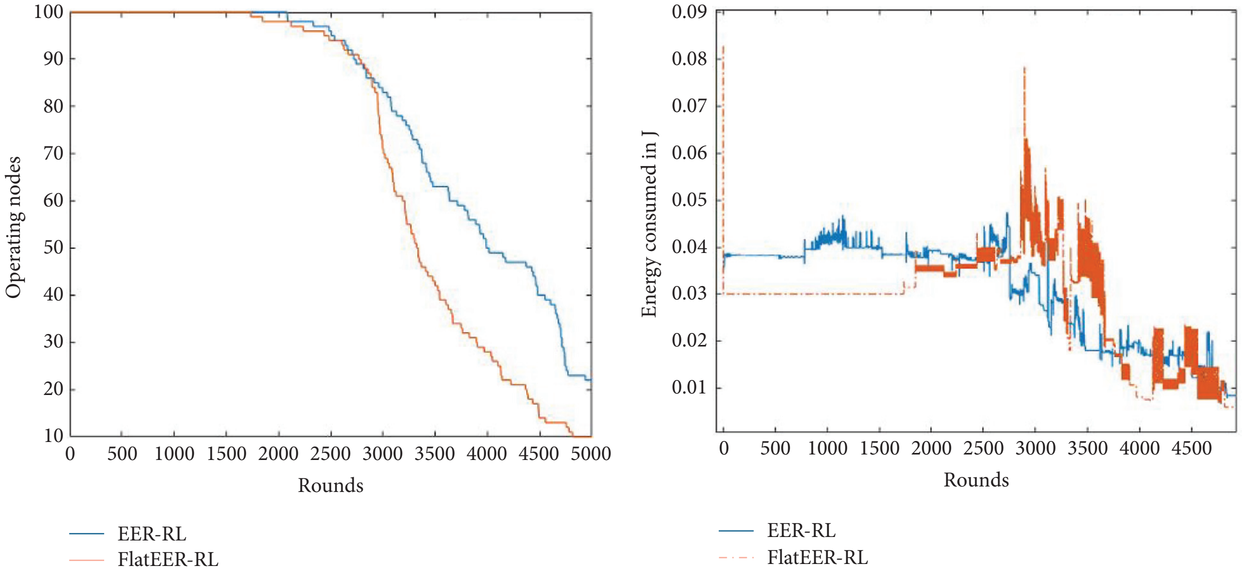
5.3. Energy Efficiency and Network Lifetime Evaluation. The major objective of this paper is to enhance energy efficiency and extend the network lifetime. Various definitions



(a)

(b)

Figure 3: Performance evaluation of EER-RL using different values of p and q.



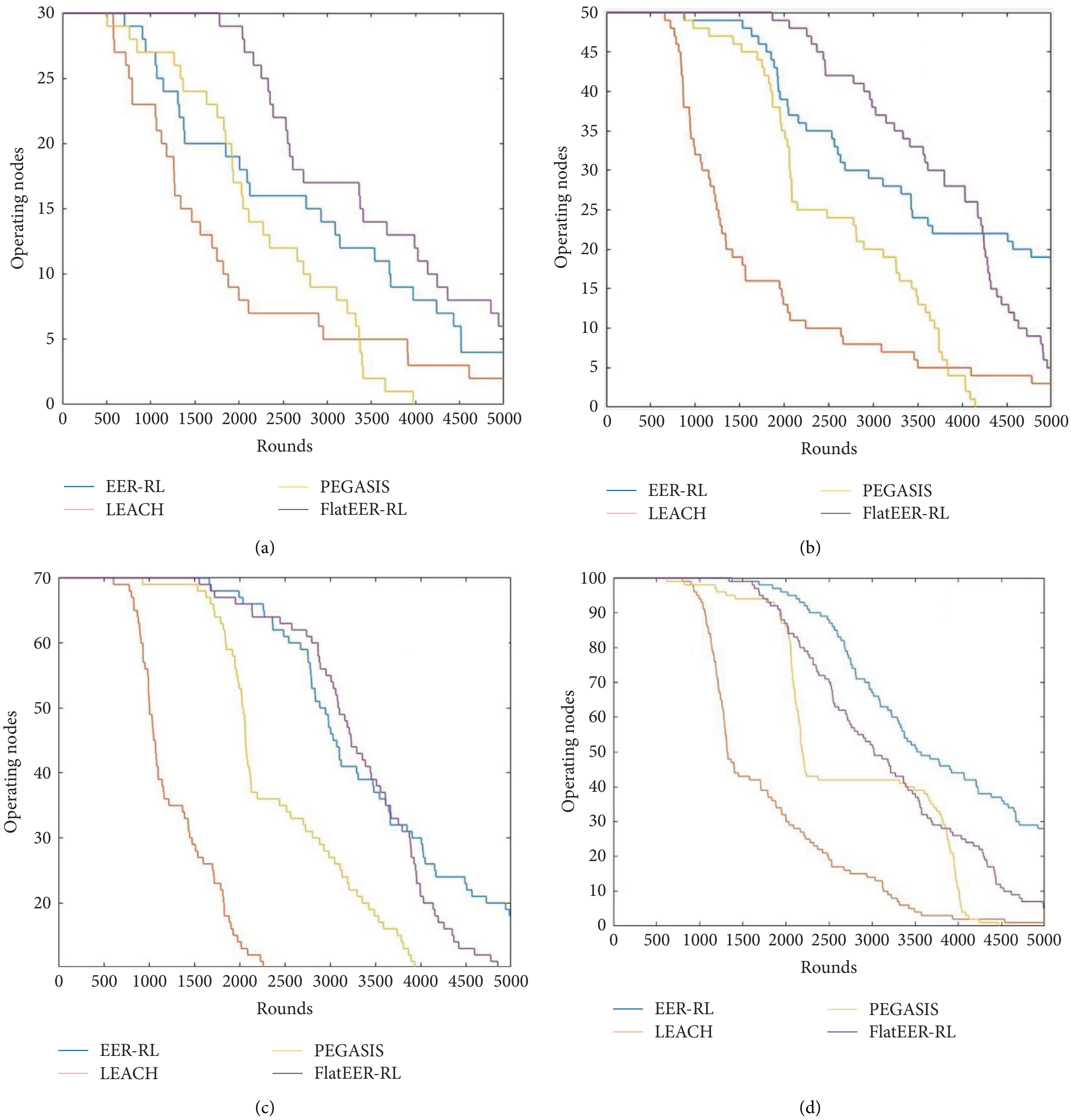
(a)

(b)

Figure 4: Comparison between cluster-based EER-RL and flat-based EER-RL protocols.

of network lifetime were presented in the literature. In this paper, we define network lifetime as the time until data transmission is no longer possible. We evaluated the energy efficiency and network lifetime of the proposed protocol by comparing it with existing clustering protocols such as LEACH [22] and PEGASIS [23]. We used the following metrics for comparison: (1) number of alive devices per round; this metric also helps to evaluate the network lifetime. (2) Energy consumed per round. It is the sum of the energy consumed by all the devices each round, and (3) time until the first device dies. These metrics are used to evaluate energy efficiency.

In Figure 5, we evaluated the proposed protocol's network lifetime using a different number of sensor devices. Also, we made a comparison with LEACH and PEGASIS to prove the efficiency of the proposed protocol. With 30 and 50 devices as shown in Figures 5(a) and 5(b), respectively, FlatEER-RL performs better than EER-RL and EER-RL slightly performed better than LEACH and PEGASIS. However, in Figure 5(c), with 70 devices, FlatEER-RL and



(a)

(b)

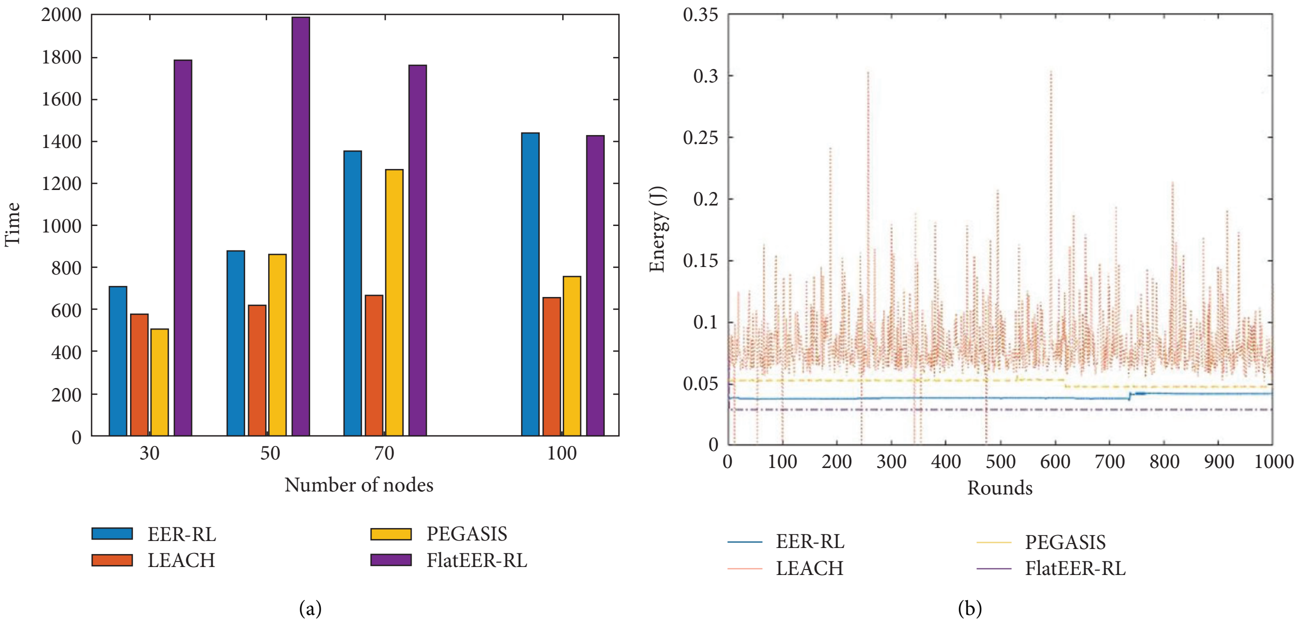
(c)

(d)

Figure 5: Network lifetime and energy consumption evaluation. We conducted the experiment in 4 phases: (a) with 30 devices, (b) with 50 devices, (c) with 70 devices, and (d) with 100 devices.

EER-RL showed similar results, yet both outperformed LEACH and PEGASIS, while with 100 devices, EER-RL outperformed much better than FlatEER-RL. The proposed protocol shows better results with long network lifetime compared to the existing protocols. We equally considered the residual energy and the hop count to maximize the network lifetime because the energy consumed when transmitting the data can be high if the distance is too large. Furthermore, we also compared the time until the first device dies in Figure 6(a); in all test scenarios, the proposed protocol outperformed LEACH and PEGASIS. From the above, we can note that the cluster-based version of the routing protocol proposed in this paper works better in a large-scale network, while for a small network(less or equal to 50 devices), the flat-based version can be preferred.

On the other hand, energy consumption mainly affects the network performance; this is to say, the usage of devices' energy can have a good or bad influence on the overall network performance [33–36]. Therefore, it is crucial to consider the energy efficiency in a routing protocol. Figure 4(b) shows the energy consumption evaluation. The performance results show that the proposed protocol



(a)

(b)

Figure 6: Network lifetime and energy consumption evaluation.

consumes less energy and outperforms LEACH and PEGASIS. We used a cluster-based routing protocol with RL to avoid the cold start with a flat-based protocol and enhance the network lifetime. Moreover, to speed up the learning process, we set the learning rate α to 1, and after the learning process, the energy consumed per round was considerably reduced. In addition, we set up the discount factor γ to 0.95 to focus more on the future reward. This played a major role in balancing energy consumption in the long run, minimizing the energy consumed by each device, and extending the network lifetime, as shown in Figure 6.

6. Conclusions and Future Considerations

In this paper, we proposed a cluster-based energy-efficient routing protocol for IoT using Reinforcement Learning, named EER-RL. The objective of this work was to optimize energy consumption and prolong the network lifetime by finding an optimal route for data transmission. We produced two versions of the same algorithm, one is cluster-based (EER-RL) and the other one flat-based (FlatEER-RL), and through comparison, we proved that the cluster-based routing protocol proposed is more scalable than the flat-based one. However, it is preferred to use a flat-based version of the proposed work in a small network.

EER-RL was designed in three phases, such as network set-up and CH election. We considered the hop count factor and initial energy to compute the initial Q-value used for the CH election in this phase. The second phase was to form clusters, every CH sends an invitation to all the devices in its transmission range, and every device far from the base station joined the cluster whose CH was the closest. Finally, the data transmission phase characterized by the learning provided an energy-efficient routing with both the residual energy of devices and hop count considered for making routing decisions. Moreover, an energy threshold was set for CH replacement. The simulation results showed that EER-RL achieved better energy consumption and network lifetime than LEACH and PEGASIS.

In this paper, we used a lightweight RL to reduce the protocol runtime and minimize energy consumption. In the future, we would like to consider other parameters for a more optimal routing protocol.

Data Availability

The source code used in this manuscript is available from the following author upon request: Mutumbo (mutombo.kazadi@gmail.com).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D81B01016073).

References

[1] J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 122–128, 2015.

[2] I. Yaqoob, E. Ahmed, I. A. T. Hashem et al., "of things architecture: recent advances, taxonomy, requirements, and

open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10–16, 2017.

[3] A. Imran, S. Sendra, J. Lloret, and A. Oumnad, "Internet of things for measuring human activities in ambient assisted living and e-health," *Network Protocols and Algorithms*, vol. 8, no. 3, pp. 15–28, 2016.

[4] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, "Software-defined internet of things for smart urban sensing," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 55–63, 2015.

[5] V. A. Dhtore, A. R. Verma, and S. B. Thigale, "Energy efficient routing protocol for IoT based application," in *Proceedings of the 2nd International Conference on Advanced Technologies for Societal Applications*, pp. 197–204, Maharashtra, India, November 2018.

[6] M. S. Obaidat and S. Misra, *Inside a Wireless Device: Structure and Operations*, Cambridge University Press, Cambridge, UK, 2014.

[7] C. A. Suescun and M. Cardei, "Anchor-based routing protocol with dynamic clustering for internet of things WSNs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 130, pp. 1–12, 2019.

[8] S. Hamrioui, C. A. M. Hamrioui, J. Lioret, and P. Lorenz, "Smart and self-organised routing algorithm for efficient IoT communications in smart cities," *IET Wireless Sensor Systems*, vol. 8, no. 6, pp. 305–312, 2018.

[9] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[10] D. Kandris, A. Alexandridis, T. Dagiuklas, and E. Panaousis, "Applications multiobjective optimization algorithms for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2020, no. 1, pp. 1–5, 2020.

[11] S.-H. Vergados, *Hardware Design for WSNs*, Springer, London, UK, 2014.

[12] Internet of Things, *Wireless Sensor Networks*, IEC, Geneva, Switzerland, 2014.

[13] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, "Communication protocols for wireless sensor networks: a survey and comparison," *Heliyon*, vol. 5, no. 5, p. e01591, 2019.

[14] I. Shallari, S. Krug, and M. O'Nils, "Communication and computation inter-effects in people counting using intelligence partitioning," *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 1869–1882, 2020.

[15] M. Pavelic, V. Bajt, and M. Kusek, "Energy efficiency of machine-to-machine protocols," in *Proceedings of the 41st International Convention on Information and Communication Technology*, pp. 361–366, Electronics and Microelectronics (MIPRO), Opatija, Croatia, May 2018.

[16] J. Shin and J. Hwang, "Intelligent energy information service based on a multihome environment," in *Proceedings of the 3rd International Conference on Ambient Systems, Networks and Technologies (ANT)*, pp. 197–204, Warsaw, Poland, December 2012.

[17] L. Xu, G. M. P. O'Hare, and R. Collier, "A smart and balanced energy-efficient multi-hop clustering algorithm (smart-BEEM) for MIMO IoT systems in future networks," *Sensors*, vol. 17, no. 7, p. 1574, 2018.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, UK, 2nd edition, 2020.

[19] Z. Mammeri, "Reinforcement learning based routing in networks: review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916–55950, 2019.

[20] C. Savaglio, P. Pace, G. Aloi, A. Liotta, and G. Fortino, "Lightweight reinforcement learning for energy efficient communications in wireless sensor networks," *IEEE Access*, vol. 7, pp. 29355–29364, 2019.

[21] D. A. Dugaev, I. G. Matveev, E. Siemens, and V. P. Shuvalov, "Adaptive reinforcement learning-based routing protocol for wireless multi-hop networks," in *Proceedings of the XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pp. 209–218, Novosibirsk, Russia, October 2018.

[22] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 1–10, Maui, Hawaii, January 2000.

[23] S. Lindsey and C. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference*, pp. 1125–1130, Big Sky, MA, USA, May 2002.

[24] M. Littman and J. Boyan, "A distributed reinforcement learning scheme for network routing," in *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications*, pp. 1–6, Seattle, WA, USA, March 1993.

[25] A. Forster and A. L. Murphy, "FROMS: Feedback routing for optimizing multiple sinks in wsn with reinforcement learning," in *Proceedings of the 3rd. International Conference on Intelligent Sensors*, pp. 371–376, Sensor Networks and Information, Melbourne, Australia, August 2007.

[26] G. Oddi, A. Pietrabissa, and F. Liberati, "Energy balancing in multi-hop wireless sensor networks: an approach based on reinforcement learning," in *Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp. 262–269, Leicester, UK, July 2014.

[27] S. Z. Jafarzadeh and M. H. Y. Moghaddam, "Design of energy-aware QoS routing algorithm in wireless sensor networks using reinforcement learning," in *Proceedings of the 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 722–727, Mashhad, Iran, October 2014.

[28] Y. Akbari and S. Tabatabaei, "A new method to find a high reliable route in Iot by using reinforcement learning and fuzzy logic," *Wireless Personal Communications*, vol. 112, no. 2, pp. 967–983, 2020.

[29] P. Costa, M. Cesana, S. Brambilla, L. Casartelli, and L. Pizziniaco, "A cooperative approach for topology control in wireless sensor networks: experimental and simulation analysis," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–10, Sydney, Australia, June 2008.

[30] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[31] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3/4, pp. 279–292, 1992.

[32] C. F. Gauss, "Normal distribution," 2001.

[33] Y. H. Robinson, E. G. Julie, S. Balaji, and A. Ayyanar, "Energy aware clustering scheme in wireless sensor network using neuro-fuzzy approach," *Wireless Personal Communications*, vol. 95, no. 2, pp. 703–721, 2017.

[34] P. Padmaja and G. V. Marutheswar, "Optimization of wireless sensor network," in *Proceedings of the International Conference on Electrical, Electronics, and Optimization*

Techniques (ICEEOT), pp. 161–166, Chennai, India, March 2016.

[35] M. A. Mahmud, A. Abdelgawad, and K. Yelamarthi, “Energy efficient routing for internet of things (IoT) applications,” in *Proceedings of the IEEE International Conference on Electro Information Technology (EIT)*, pp. 442–446, Chicago, IL, USA, July 2017.

[36] N. Nasser, L. Karim, A. Ali, M. Anan, and N. Khelifi, “Routing in the internet of things,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM 2017)*, pp. 1–6, Anaheim, CA, USA, December 2017.