Energy-efficient deep Q-network: reinforcement learning for efficient routing protocol in wireless internet of things

Sampoorna Bhimshetty¹, Agughasi Victor Ikechukwu²

¹Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore, India
²Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore, Mysore, India

ABSTRACT

The internet of things (IoT) underscores pivotal real-world applications ranging from security systems to smart infrastructure and traffic management. However, contemporary IoT devices grapple with significant challenges pertaining to battery longevity and energy efficiency, constraining the assurance of prolonged network lifetimes and expansive sensor coverage. Many existing solutions, although promising on paper, are intricate and often impractical for real-world implementations. Addressing this gap, we introduce an energy-efficient routing protocol leveraging reinforcement learning (RL) tailored for wireless sensor networks (WSNs). This protocol harnesses RL to discern the optimal transmission route from the source to the sink node, factoring in the energy profile of each intermediary node. Training of the RL algorithm is facilitated through a reward function that includes energy outflow and data transmission efficacy. The model was compared against two prevalent routing protocols, LEACH and fuzzy C-means (FCM), for a comprehensive assessment. Simulation results highlight our protocol’s superiority with respect to the active node count, energy conservation, network longevity, and data delivery efficiency.

Keywords: Energy consumption, Energy-efficient routing, Network performance, Q-learning, Reinforcement learning

This is an open access article under the CC BY-SA license.

Corresponding Author:
Agughasi Victor Ikechukwu
Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore
Mysore, Karnataka, India
Email: victor.agughasi@gmail.com

1. INTRODUCTION

The internet of things (IoT) is a network of electronic devices that can gather, process, and share data, aiming to better current services [1]. There are many practical use cases of IoT, such as in healthcare, waste management, transportation, and more [2], [3]. Devices like radio-frequency identification (RFID) tags, mobile phones, and sensors gather data via a vast IoT network. However, these devices, often called sensor nodes, have limited computing power and short battery life. The existing approaches to direct data, or routing methods, in wireless sensor networks (WSNs) are complex and need resources like processing power and memory that many IoT devices do not have in surplus [4].

Devices in the IoT include smartphones, wireless sensors, and RFID tags, among others [5], [6]. A typical sensor has parts for power, sensing, processing, and communicating [7]–[10]. While the power part supplies energy to others, the other parts use very little energy [11]–[13]. This energy efficiency becomes paramount for devices deployed in challenging terrains and environment where frequent battery replacements or recharges are untenable [13], [14]. It is a widely accepted notion that energy-efficient routing algorithms can judiciously control the power consumption, thereby helping prolong the network’s operational longevity.

Given these challenges, there is a growing demand for energy management strategies that are resource-dependent yet robust to implement. Distributing the routing process evenly across devices is pivotal...
for such an energy-efficient network. This ensures balanced energy dissipation and increases the likelihood that more gadgets will keep working for longer.

In this regard, our research introduces a reinforcement learning-centric approach strategy for designing a low-power IoT routing protocol. The proposed method improves the longevity and scalability of an IoT network by balancing the energy produced in the network, and adeptly distributing energy among devices. The protocol identifies optimal transmission routes by incentivizing nodes to exchange locally pertinent data using residual energy as criteria. Furthermore, the hop count attribute was finetuned to minimize overall network latency. The following are our contributions:

– We repurpose energy optimization as a crucial part of routing concerns in the IoT landscape.
– We suggest a new routing plan that uses the least energy.
– We use a method called Q-learning to bolster energy savings further.
– Benchmarking the network efficiency against two established state-of-the-art methodologies.

The paper is in five sections: Section 2 discusses other works on power-saving routing for IoT-based WSNs. In section 3 explains the proposed methodology. The simulation setup and the ensuing results are presented in section 4. Section 5 summarizes the paper, offering concluding remarks and potential avenues for future exploration.

2. RELATED WORKS

2.1. Reinforcement learning

Reinforcement learning (RL) refers to a class of algorithms used to improve the efficiency of Markov property-based sequential decision-making [15], [16]. These algorithms operate by learning to make decisions through trial and error, essentially learning a strategy or policy that maps states of the world to the actions that should be taken in those states. The fundamental principle of RL is the concept of reward: algorithms learn to take actions that maximize cumulative reward over time. This approach makes RL especially suitable for complex environments where explicit programming of all possible scenarios is impractical. The RL paradigm can be divided into: i) model-based RL: when the model of the system is available and ii) model-free RL: when the system’s model is not known.

Central to RL is the agent’s interaction with the environment. As depicted in Figure 1, the agent makes decisions based on the state $S_t$ of the environment. The agent takes an action $a_t$, and as a result of the consequence of this action, the environment transitions to a new state $S_{t+1}$ and provides a reward $R_t$ to the agent.

![Figure 1. Reinforcement learning agent interaction](image)

In scenarios where the return reward is not instantaneous but rather delayed, meaning that the immediate reward does not accurately reflect the agent’s true performance. The cumulative reward $R_t$ is calculated as (1):

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \ldots = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$$

where $\gamma$ = factor, capturing the uncertainty of future rewards. The policy can never guarantee that the same future reward will be gained by the same behaviour because of the inherent uncertainty in the environment. The primary goal of RL algorithms is to ascertain the optimal action strategy, $\pi^*$, as described in (2):

$$\pi^* = \arg\max_{\pi} \mathbb{E}[R]$$

where $\mathbb{E}[R]$ is the expected return.
\[ \pi^+ = \arg \max_{\pi} \xi_{\pi} \left[ \sum_{t=0}^{t_{\text{max}}} \gamma^t r_t \right] \]  

(2)

### 2.2. Deep reinforcement learning with deep neural networks

Recent advances in deep learning (DL), particularly the use of deep neural networks (DNNs), have enabled the extraction of intricate patterns from high-dimensional data sources like images, audios, and videos [17]. DNNs in the context of RL, termed deep reinforcement learning (DRL), is depicted in Figure 2. DRL leverages DNNs to deduce optimal policies and extract relevant environmental data. DRL can determine either the Q-value (value-based) for every state-action combination or predict a probability distribution of potential actions (policy-based) [18], [19].

![DRL framework](image)

Figure 2. DRL framework

### 2.3. Wireless network protocols

Wireless networks face challenges in security, reliability, and energy efficiency. Researchers have proposed different protocols to address these issues [20]–[22]. Recent advancements have also seen the integration of machine learning and game theory to factor-in the optimal routing approaches for wireless networks [23], [24]. WSNs have seen the evolution of diverse routing protocols. Kooshari et al. [25] classifies these into three primary categories:

- Flat routing: data-centric protocols without any constraints on data origin [26], [27].
- Location-based routing: these assign addresses to sensor nodes based on their geographical locations. Techniques include relative distance calculations from signal strengths and global positioning system (GPS)-based location tracking [28].
- Hierarchical routing: these protocols prioritize energy conservation. They distribute routing responsibilities based on the capabilities of the devices, ensuring optimal energy use [29]–[31]. Table 1 summarizes the findings of relevant literature.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL basics</td>
<td>Algorithms for decision-making based on Markov property</td>
<td>[15], [16]</td>
</tr>
<tr>
<td>DRL with DNNs</td>
<td>Integration of deep learning with RL for high-dimensional data</td>
<td>[17]–[19], [32], [33]</td>
</tr>
<tr>
<td>Wireless network protocols</td>
<td>Protocols addressing security, reliability, and energy efficiency</td>
<td>[20]–[24]</td>
</tr>
<tr>
<td>WSN routing types</td>
<td>Classification into flat, location-based, and hierarchical routing</td>
<td>[25]–[31]</td>
</tr>
</tbody>
</table>

### 3. METHOD

In addressing energy efficiency challenges within IoT routing, this research employs the adaptability of RL. With RL’s potential to optimize decisions based on environmental feedback, it offers a solution for dynamic, resource-limited IoT networks. Central to the approach is an agent interacting with the IoT network, observing network states such as residual energy, data queue lengths, and traffic. To manage the high dimensionality in large-scale IoT networks, DRL is introduced, enabling the agent to extract patterns via DNNs for better policy learning. Building on this, the energy-efficient deep Q-network (EEDQN) routing protocol is proposed in Figure 3. EEDQN employs a Q-network that, using the network state, estimates Q-values for potential actions, emphasizing energy conservation. This ensures decisions that maximize long-term rewards, adapting to IoT network dynamics.
3.1. Network setup and cluster formation phase

The process primarily focuses on establishing the network and selecting the cluster head, which is divided into two segments. Initially, devices determine their preliminary Q-values using inherent data. The central station disseminates a periodic signal detailing its geographical coordinates. On intercepting this signal, every device registers the central station’s location and applies (1) and (2) to deduce its starting Q-value, considering the energy reserve and number of hops. It’s presumed that each device possesses a unique energy reserve. A set distance boundary is maintained between the cluster heads (CHs) and the central station to streamline the network and aid distant sensors in locating a CH. To ensure efficient energy usage and direct connections towards the central station, CHs are strategically positioned away from the network’s periphery, minimizing extended communication paths. The entire election process is described in Algorithm 1. Q-value computation based on energy level and hop count, as presented in (3) and (4):

\[
Q(x, y) = Q_{prev}(x, y) + \beta (r + \gamma \cdot \text{highest}(Q_{next}(x', y')) - Q(x, y)) 
\]

3. Energy level update

\[
E(t+1) = E(t) - ETx(k) - ERx(k) - EDA
\]

where: \( Q(x, y) \) = Q-value of a state-action pair; \( x \)= learning rate; \( r \)= reward for taking action \( y \) in state \( s \); \( \gamma \)= discount factor; \( \text{max}(Q(x', y')) \)= maximum Q-value over all possible actions \( x' \) in the next state \( y' \). \( E(t) \)= current energy level; \( ETx(k) \)= energy used to transmit a packet of length \( k \). \( ERx(k) \)= energy used to receive a packet of length \( k \). \( EDA \) is the energy consumed by the device for data aggregation.

Algorithm 1. The entire election processes

1. Network setup and cluster head election:
   - For each device do:
     1a. SELECT a random backoff time and wait for it to expire
     1b. ASSIGN the cluster head to the device with maximum energy
     1c. BROADCAST a group head announcement message to inform other devices of its status.
   - End For

2. Cluster Formation:
   - For each device that detects a group head announcement do:
     2a. JOIN the group with the strongest signal strength.
     2b. SEND a message to the identified cluster head.
     2c. UPDATE its routing table using the local information from the cluster head.
   - End For

3. Data Transmission:
   - WHILE! Destination Node do:
     3a. SELECT the next-hop device using its routing table, prioritizing residual energy and position coordinates.
3b. TRANSMIT local information to the selected next-hop device.
3c. EXTRACT information from the received packet header and UPDATE its routing table.
End While

Extract the information encapsulated in the packet header and UPDATE its routing table.
4. Reinforcement learning:
4a. EVALUATE the energy efficiency of routing decisions based on energy consumption and packet transmission distance.
4b. UPDATE routing decisions using a Q-learning algorithm to optimize future choices.

3.2. Data transmission phase
In the phase of data transmission, each device harnesses RL to maximize its routing decisions. This adaptive decision-making process involves recalculating the Q-value, considering both immediate rewards from recent actions and the anticipated cumulative rewards. The methodology encompasses three integral components: a model estimating energy usage, a function that determines rewards, and a mechanism to refine the Q-value. The energy model gauges the remaining energy, adjusting for energy expended during packet transfers. Using this updated energy data alongside the hop count, the system computes the reward. This reward subsequently informs the Q-value recalibration, as detailed in (5) to (7). Equation for updating Q-value:

\[ Q(s, a) = Q(s, a) + \alpha (r + \gamma \max(Q(s', a')) - Q(s, a)) \]  

(5)

in (6) for energy consumption model:

\[ E(t + 1) = E(t) - ETx(k) - ERx(k) - EDA \]  

(6)

in (7) for reward function:

\[ r = \text{Reward}(E(t + 1), h) \]  

(7)

where:
\[ Q(s, a) = \text{Q-value of a state-action pair} \]
\[ \alpha = \text{learning rate} \]
\[ r = \text{the immediate reward obtained from taking action a in state s} \]
\[ \gamma = \text{the discount factor} \]
\[ \max(Q(s', a')) = \text{the maximum Q-value over all possible actions a' in the next state s'} \]
\[ E(t) = \text{the current energy level} \]
\[ ETx(k) = \text{the energy used to transmit a packet of length k} \]
\[ ERx(k) = \text{the energy used to receive a packet of length k} \]
\[ EDA = \text{the energy consumed by the device for data aggregation} \]
\[ h = \text{the hop count} \]

3.2. Reinforcement learning
3.2.1. Reward function
This metric is obtained by evaluating the remaining energy, denoted as Lr, and the aggregate of hops, represented by Nhops:

\[ \text{Reward} = g(Lr, Nhops) \]  

(8)

where \( g \) = reward function. In this stage, the proximity between the transmitting device and its neighboring devices is also taken into account. The span between a device, labeled Si, and the central station through a connecting device, Sj, is labeled as Dlink and is determined as per (9):

\[ Dlink = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \]  

(9)

where \( x_i, y_i, \) and \( z_i \) = the coordinates of device Si and \( x_j, y_j, \) and \( z_j \) are the coordinates of device Sj. The estimated hop count is computed as (10):

\[ Nh = Dlink/(TXrange) \]  

(10)
where $TX\text{range}$ is the transmission range. The reward function is then computed as:

$$\text{Reward} = Er - a \times Nh$$  \hfill (11)

where $a$ is a constant that determines the trade-off between energy consumption and hop count.

### 3.2.2. Q-value update

To ascertain the true worth of a specific action, it is essential to calculate the action-value function. This function gauges the efficacy of executing a certain action in a designated state while adhering to a strategy denoted by $\pi$. The function is calculated as a discounted total of rewards received by the agent post specific action in a given state. This can be mathematically represented in (12):

$$Q(s, a) = E[R_t + 1 + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots | S_t = s, A_t = a, \pi]$$  \hfill (12)

where:

- $Q(s,a)$: the action-value function for taking action ‘a’ in state ‘s’
- $E$: the expectation operator
- $R_t$: the reward received at time ‘t’
- $\gamma$ (gamma): the discount factor used to balance immediate and future rewards
- $S_t$: the current state of the agent
- $A_t$: the action taken by the agent in the current state
- $\pi$: the policy being followed by the agent

### 4. RESULTS AND DISCUSSION

#### 4.1. Model training and implementation

Training the EEDQN network entails using past data from the IoT network alongside simulated scenarios. This ensures robustness in the learned policies. The model is trained iteratively, with each episode involving. Observing the current state, taking an action based on the Q-network, receiving a reward from the environment, and adjusting the Q-network based on the observed reward using backpropagation.

We performed simulations in Python 3.10 to validate the effectiveness of the model. 120 devices were randomly distributed over a sensing field of 150 m×150 m. The sub station was positioned at the center of the sensing field with coordinates of (75, 75). The network was designed with diversity in mind, featuring devices that have energy capacities spanning between 1 and 2 joules. The simulation parameters are detailed in Table 2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>1.00</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.92</td>
</tr>
<tr>
<td>$h$</td>
<td>20</td>
</tr>
<tr>
<td>Field size</td>
<td>150 m×150 m</td>
</tr>
<tr>
<td>Sum of devices</td>
<td>120</td>
</tr>
<tr>
<td>Initial energy</td>
<td>1.5 joules</td>
</tr>
</tbody>
</table>

The suggested protocol incorporates two crucial metrics: the hop count and the remaining energy. Additionally, probabilistic values, denoted as ‘a’ and ‘b’, play a role. The choice of a device hinges on its energy status and hop count. When ‘a’ has a larger value, devices with abundant energy are more likely to be chosen. Conversely, a greater ‘b’ value favors devices that have fewer hops to the primary station. Figure 4 provides the results of the initialization of nodes using the RL approach, while Figure 5, showcasing the simulation results with parameters set at $\alpha=0.3$, and $b=0.7$ demonstrates the protocol’s optimal performance.

Remarkably, the best performance was reached with fewer than 35 operating nodes, highlighting the efficiency of our approach. This is particularly evident when assessing the energy consumption metrics, as depicted in Figures 6 and 7. The cumulative energy consumed tends to decrease as the sum of nodes increases, this effect is captured in Figure 7.
Figure 4. RL-based node initialization

Figure 5. Effects on performance ($a=0.3$, $b=0.7$)

Figure 6. Cluster-based energy consumption
An assessment was conducted against established clustering protocols. Key performance indicators included energy consumption per operational round and the duration of sustained network activity before energy depletion, using state of the art (SoTA) clustering protocols like LEACH, HEED, and smart-BEE [34]. We used two metrics for comparison: i) network area per round which also helps evaluate the network lifetime and ii) energy consumed by all devices each round. In Figure 8, we assessed the network efficiency by varying the number of sensor devices used.

To improve the network lifetime and avoid the challenges associated with a cold start, a cluster-based routing protocol with RL was incorporated [35]. The learning rate $\alpha$ set to 1 accelerates the learning process, which resulted in a significant reduction in energy consumption per round. Additionally, by setting the discount factor $\gamma$ to 0.92, we were able to prioritize future rewards, which contributed to better balancing of energy consumption over time, minimizing energy consumption per device, and ultimately extending the network lifetime, as demonstrated in Figure 7.

The consequent energy distribution, as illustrated in Figure 8, underscores the protocol’s efficiency, yielding enhanced energy efficiency and a more extended network operational span. Such advancements underscore the potential benefits for enterprises and researchers, highlighting avenues for optimizing IoT network efficiency and reduce energy costs. Table 3 provides a comparative analysis of the proposed approach.

### Table 3. Comparative analysis with state-of-the-art protocols

<table>
<thead>
<tr>
<th>Metrics (average over 50 rounds)</th>
<th>Proposed protocol</th>
<th>LEACH</th>
<th>HEED</th>
<th>Smart-BEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network area per round (m²)</td>
<td>11.250</td>
<td>9.750</td>
<td>10.500</td>
<td>9.800</td>
</tr>
<tr>
<td>Energy consumed per round (Joules)</td>
<td>1.1</td>
<td>1.5</td>
<td>1.4</td>
<td>1.6</td>
</tr>
</tbody>
</table>
From Table 3, the following implications and observations can be drawn: the proposed protocol demonstrates superior area coverage, registering 11,250 m² per round and outpacing LEACH, HEED, and smart-BEE by significant margins. It also excels in energy efficiency, consuming only 1.1 Joules per round, notably less than its counterparts. This efficiency translates to extended device battery life and network longevity. Moreover, the protocol's expansive coverage and low energy consumption highlight its operational efficacy. It promises scalability, maintaining performance as network size grows [36]. When compared to the established protocols like LEACH, HEED, and Smart-BEE, the proposed protocol's advantages become evident, underscoring the benefits of integrating reinforcement learning with a cluster-based routing approach [37].

5. CONCLUSION

This research introduced an RL-based routing protocol for IoT aimed at optimizing energy consumption and extending network lifespan. By leveraging initial energy and hop count, the protocol effectively determined the optimal route for data transmission. Through a multi-phased approach, including cluster formation and data transmission, reinforcement learning was employed to prioritize energy-efficient routing. Comparison with existing protocols like LEACH, smart-BEE, and HEED validated the superior energy efficiency and longevity of the proposed method. The result further showcased the protocol's suitability for modern IoT networks, emphasizing its balance between sustainability and reach. An extension of this work will explore additional parameters, like network traffic and node mobility, and the possibility of incorporating other deep learning techniques to refine the protocol.

REFERENCES

BIographies of authors

Sampoorna Bhimshetty received her Bachelor of Engineering (B.E.) degree from VTU University, Belgaum in Computer Science in 2014, and currently pursuing her M.Tech from Computer Science from Bangalore University-India. From 2014 to 2016, she was a Junior Software developer in Tech Narayana Software Pvt Ltd. Bangalore-India. From 2019 to 2020, she was a faculty in KLE S.NJalingappa College in Bangalore-India. Since 2022, she is working as a faculty in AVK group of Institutions in Bangalore-India. Her research interest includes efficient routing protocol in wireless IoT, computer vision and deep learning. She can be contacted at email: sampoornabhim23@gmail.com.

Agughasi Victor Ikechukwu received his Bachelor of Science (B.Sc.) degree from the Michael Okpara University of Agriculture Umudike-Nigeria in 2010, and Masters (M.Sc.) degree from Bangalore University, India in 2016 and currently pursuing his Ph.D. in Computer Science from the University of Mysores-India. He has authored 7 research articles, conducted 12 workshops in deep learning, and has an educational YouTube channel (Vactor’s Portal) where he shares his ideas on python for machine learning. His research interest includes medical image processing, computer vision and deep learning. He was a recipient of the best graduating student award of 2016 batch and holds a gold medal in computer science during his master’s degree, a student member of IEEE, and a recipient of the best paper award at the international conference on computing system and its applications in 2021. He can be contacted at email: victor.agughasi@gmail.com.