Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3337831, 16 pages
https://doi.org/10.1155/2022/3337831

Research Article

ReLeC: A Reinforcement Learning-Based Clustering-Enhanced Protocol for Efficient Energy Optimization in Wireless Sensor Networks

Tripti Sharma,[1] **Archana Balyan**,[2] **Rajit Nair**,[3] **Paras Jain**,[1] **Shivam Arora**,[1] **and Fardin Ahmadi**[4]

[1]*IT Department, Maharaja Surajmal Institute of Technology, New Delhi, India*
[2]*ECE Department, Maharaja Surajmal Institute of Technology, New Delhi, India*
[3]*School of Computing Science & Engineering, VIT Bhopal University, Bhopal, Bhopal-Indore Highway Kothrikalan, Sehore, MP, India*
[4]*Lecturer of Computer Science Faculty, Rana University, Kabul, Afghanistan*

Correspondence should be addressed to Fardin Ahmadi; fardin.ahmadi@bcs.ru.edf.af

Received 2 May 2022; Accepted 21 July 2022; Published 29 August 2022

Academic Editor: Mohammad Farukh Hashmi

Wireless sensor networks (WSNs) are a widely studied area in the field of networked embedded computing. They are made up of several sensor nodes, which keep track of a variety of physical and environmental parameters, like temperature and humidity. The nodes are autonomous, self-configuring, and wireless. A significant problem in WSNs is that sensors in these networks consume a lot of energy. Energy consumption is a big issue when it comes to the deployment of sensor networks. The reason for this is the cost of operating a sensor node and the cost incurred due to energy consumption. Energy optimization is based on intelligent energy management. This paper presents a reinforcement learning-based and clustering-enhanced method. Reinforcement learning is a set of algorithms inspired by operant conditioning in animal behavior, and clustering-based methods have been extensively used for devising energy-efficient protocols. The proposed method is able to plan and schedule the nodes to ensure an extended network lifetime. In this work, we aim to assess and increase the efficiency of power consumption and reduce sensor node energy loss. The simulation results prove that the presented protocol effectively reduces the energy consumption of sensor nodes and ensures a prolonged lifetime of the sensor network.

1. Introduction

A wireless sensor network (WSN) is made up of sensors and sink nodes that operate in an ad hoc network to interpret, accumulate, and monitor events; sense physical and physiological parameters in the area they are deployed; and collaboratively transmit sensor information of conditions like temperature and humidity to multiple sink nodes.

In the previous two decades, WSN has been used in the medical area, structural health monitoring, habitat tracking, target detection in battles, disaster recovery, and chemical monitoring. WSNs achieved sustained development, especially after the advancement of the Internet of Things (IoT) [1].

Specific requirements are needed to support many devices in WSNs, including energy efficiency, complexity, delay, robustness, security, and sensor location. The architecture of a WSN may alter because a node may escape from an operating network owing to high battery exhaustion, and sensor nodes and sink nodes relocate in some situations. To preserve and calculate the energy efficiency, it is vital to run a fully working network for the longest extended time achievable, particularly for nodes deployed in severe conditions where battery charging and changing are difficult.

To improve the overall efficiency to extend the lifetime of the network operation has received great attention in research and has remained part of the objective, because sensor nodes may be put in difficult situations.

Various protocols have been established for the progress of routing protocols in WSNs. However, certain improvements are needed to be made. Routing protocols presume nodes of the same kind. In this instance, conventional methods such as the maximum available power, minimum number of hops route, and minimal energy route may perform sufficiently. Also, networks will begin to have limited network longevity and lower power efficiency in heterogeneous networks. That is why the diversity of nodes in terms of transmission of data and energy capabilities must be examined. Flat or hierarchical protocols are the protocols that are usually preferred. Flat routing is a multihop routing method in which each node is operational and the same tasks are assigned to all nodes. In flat network topology, all nodes help to achieve sensing tasks. The standard adopted lifetime definition is when the first node is dead regarding optimizing a lifetime. This time is not very significant because when an individual node is dead, the complete network remains to operate.

Machine learning systems have shown great promise and results in various areas including intrusion detection. Bhadoria et al. used different ensemble feature selection methods with different ML models for intrusion detection and found hybrid approaches like random forest with a support vector machine delivering improved real-time performance for intrusion detection in power systems [2].

In past years, there is an increasing interest in adding an artificial intelligence method called RL to different schemes in WSN to increase network performance. RL is a subfield of machine learning methods that reward desired behavior and punish undesired. The agent determines by taking sequential actions in its environment, examining the state of the environment and taking a reward. The agent must learn a policy approach to choose which action to take in any state. A reinforcement learning agent, in general, can detect and comprehend its surroundings, act, and learn through trial and error.

This method has caused dynamic routing and adaptive capability in data transmission related to standard routing methods [3].

A range of methods can be interpreted using reinforcement learning models, and lastly, different networking performances can be enhanced utilizing reinforcement learning algorithms.

WSNs consume a lot of energy, and this is a deterrent to their large-scale deployment and usage. The reason for this is the cost of operating a sensor node and the cost incurred due to energy consumption. The following work is proposed as a solution to this long-standing problem, by reducing energy used for communication by giving an optimal route.

The research work can be broken down into two parts, namely, clustering (which involves cluster-head selection and cluster formation) and the application of reinforcement learning. Routing protocols that employ clustering are found to have a higher stability period and longer network lifetimes. We use both clustering and RL to give an optimal routing protocol.

While conducting literature review, we made a startling observation that many authors view the routing problem through a partisan prism, either focussing too much on the residual energy; disregarding multihop communication, which is very prevalent in contemporary times; or laying excessive emphasis on the number of hop parameter, when both need equal consideration to work well in the real world. In addition, reinforcement learning has established itself as an exemplary approach for sequential decision making and has shown how probabilistic decision making works better in the longer run as compared to its counter, deterministic approaches. This research work is motivated by the aforementioned ideas.

We present a routing algorithm for WSNs that is essentially based on RL. This work is aimed at improving packet delivery and depreciating delivery time. The suggested ReLeC balances energy dissipation in WSN devices, thereby extending network lifetime and improving network scalability. ReLeC additionally offers efficient pathways utilizing a technique to share data as a reward, it is estimated by using hop count and remaining energy, and the hop count variable can lower the edge delay. To observe the overall results of ReLeC, we carried out computations, and the outcomes demonstrate that ReLeC performs that energy consumption is efficient, also prolongs the lifetime, and is adaptable for considerable WSNs.

2. Related Work and Existing Methodologies

Wang et al. [4] found that the different properties of WSNs cause optimization issues when creating energy-efficient routing methods. The majority of current routing strategies are designed to achieve one of several objectives. There are so many scholars who have conducted research on the optimization, the routing, and energy consumption algorithms and achieved certain results.

There are two kinds of clustering algorithms: centralized and distributed. Because the global knowledge of the network is required for a centralized clustering algorithm to select the number of cluster heads (CHs) that improved allocation, this proposed method is limited to wide-ranging networks. A distributed clustering algorithm, on the other hand, does not require access to the network's global data. Instead, the nodes in the system do the clustering operation separately based on local information, which reduces energy usage and is more appropriate for wide-ranging networks.

According to Heinzelman et al. [5], LEACH (Low-Energy Adaptive Clustering Hierarchy) is a hierarchical clustering algorithm that uses a randomized rotation of local CHs to evenly distribute the energy load across the nodes. LEACH can be utilised in dynamic networks with localised coordination, which allows for scalability, robustness, and a reduction in the amount of data sent to the sink node. The information is transmitted by the CH to all its neighbours to notify them that it became a CH. Wang et al. suggested that this protocol does not require large communication overhead [4].

```
for each node $i$, do
  set $D_{euclidean}$= Euclidean(node $i$, sink)
  set $N_H = (D_{euclidean}/\text{Transmission Range})$
  set $Q = (p \times (E_{residual}/E_{\max} - E_{\min})) + ((1-p) \times (N_H \times \log(1/N_H)))$
end
while len(CH$_{list}$)≤CH$_{total}$, do
  set $Q_{\max} = \max Q$
  for each node $i$, do
  if MIN$_{threshold}$≤$D_{euclidean}$ < MAX$_{threshold}$, then
    if CH$_{list}$ is empty, then
      add node $i$ to CH$_{list}$
        pop node $i$ from stack
    else
        for $head$ ⟵ 1 to len(CH$_{list}$), do
          dist = Euclidean(node $i$, CH $head$)
          if dist ≥ MIN$_{threshold}$, then
            set flag=1
          else
            set flag=0
            break
          end
    if flag==1, then
    add node $i$ to CH$_{list}$
    pop node $i$ from stack
end
```

Algorithm 1: Network set-up and cluster-head selection.

Smaragdakis et al., suggested that SEP (Stable Election Protocol) is used to achieve heterogeneity between nodes. Nodes holding the equivalent energy level are normal nodes, and some having higher energy than others are the advanced nodes [6].

In 2006, HCR was introduced, which uses optimization algorithms to improve the method. These algorithms choose the optimal solution among all the alternatives. To obtain the effective CHs, HCR (Intelligent Hierarchical Clustered Routing protocol) uses a genetic algorithm (GA).

The fitness value is calculated using energy consumption and node density, as discussed by Matin and Hussain [3].

Raghavendra [7] considered PEGASIS to be an improved form of LEACH. It is a routing protocol based on a chain that can save more energy than LEACH. The message can be aggregated along the chain and sent directly to the sink node by one random node in the chain. The fundamental flaw is that PEGASIS necessitates a global understanding of the entire network.

During CH selection, the HEED clustering procedure considers the residual energy as the primary parameter and the node's degree as a secondary parameter. Because the CHs are well dispersed, according to Chand et al. [8], it can minimise control overhead and increase network lifetime more than other clustering methods like LEACH. Importantly, no comprehensive network information is required, and all judgements are made by nodes. GSPR, introduced by Karp and Kung [9], are proximity-based routing methods that try to discover the shortest routing path.

Shah and Rabaey described a routing method that considers the sensor node residual energy and depicted that choosing the low-energy paths may not be optimum. They presented a novel system to maximize these measures called energy-aware routing that uses suboptimal paths occasionally to provide substantial gains [10]. Now, for balancing the load, the strategy of directed diffusion was taken into account. The strategy of reinforcement learning was taken into account to optimize all the goals together.

Littman and Boyan [11] were the first to propose the embedding of a learning module in each node of a given packet switched network. They concluded that the Q-routing approach performs well in comparison to the conventional shortest-path approaches in the case of high network traffic. Q-learning is a popular temporal-difference reinforcement learning algorithm which often explicitly stores state values using lookup tables they proclaimed in their seminal paper in 1992.

The fuzzy logic system and reinforcement learning are based on the nodes' remaining energies on the routes, the available bandwidth, and the distance to the sink, according to Akbari and Tabatabaei in a new method to find a high reliable route in IoT by using reinforcement learning and fuzzy logic [12]. Oddi et al. [13] presented a RL-based routing system, which assisted in optimising multihop communication, extending the lifetime of the devices, balancing their energy, and lowering network overhead.

2.1. Reinforcement Learning Overview. Reinforcement learning or RL is a branch of machine learning that tries to learn control policies using trial and error, without explicitly knowing the value of any state, action, or state-action combination concerned with how agents should act in a given environment to maximize the cumulative reward. This contrasts with model-based learning, where a value for the state,

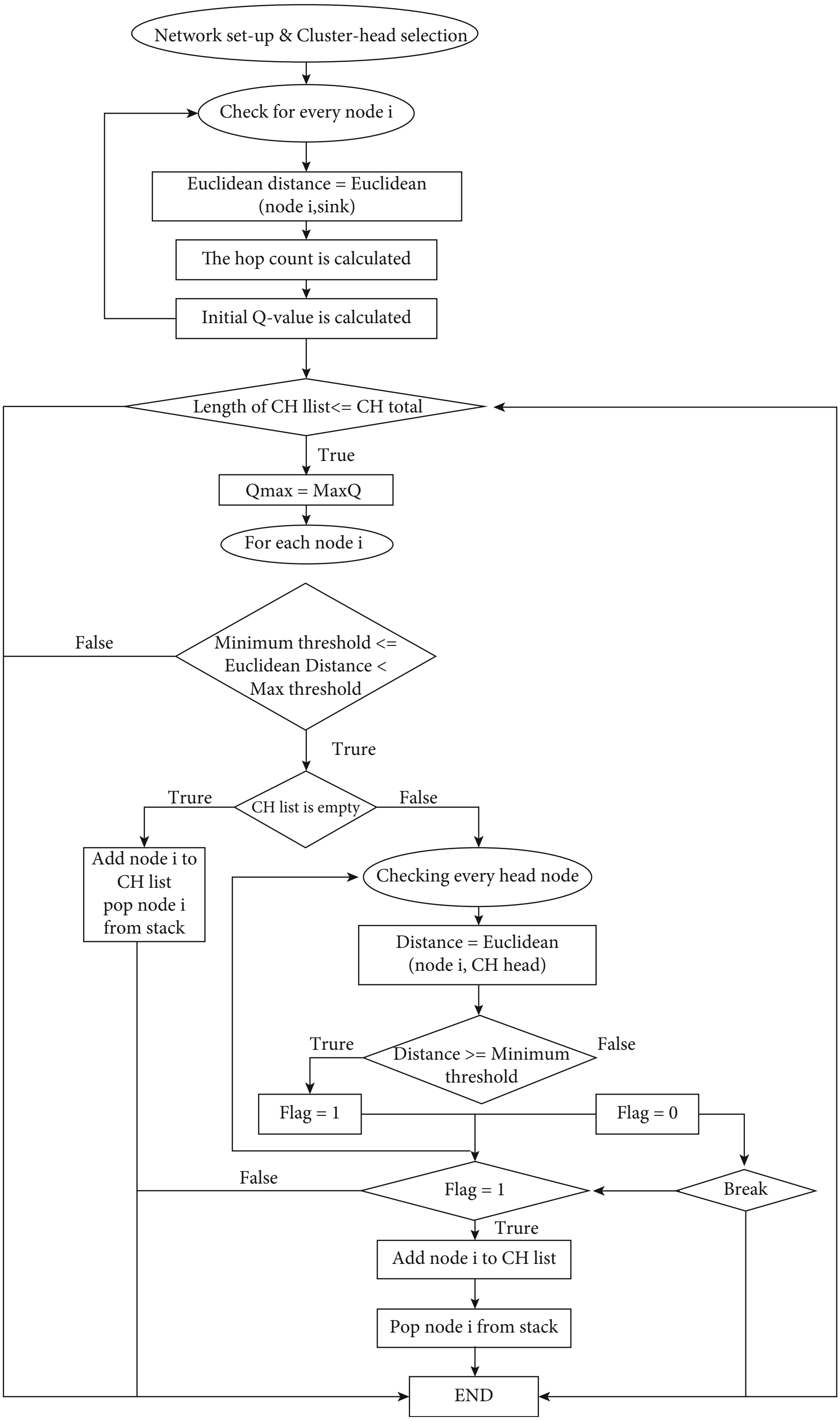

Network set-up & Cluster-head selection
Check for every node i
Euclidean distance = Euclidean (node i,sink)
The hop count is calculated
Initial Q-value is calculated
Length of CH llist<= CH total
True
Qmax = MaxQ
For each node i
False
Minimum threshold <= Euclidean Distance < Max threshold
Trure
Trure
CH list is empty
False
Add node i to CH list pop node i from stack
Checking every head node
Distance = Euclidean (node i, CH head)
Trure
Distance >= Minimum threshold
False
Flag = 1
Flag = 0
False
Flag = 1
Break
Trure
Add node i to CH list
Pop node i from stack
END

Figure 1: Flowchart of network set-up and cluster-head selection.

action, or state-action combination is predicted, and thus has a forward planning phase [14]. RL has been applied to games, robotics, artificial intelligence, and self-driving vehicles. In many cases, learning has been found to be more efficient than planning and has a potential for better generalization.

Reinforcement learning problems is often conceptualized as Markov Decision Processes or MDPs, in which the

```
for head in len($CH_{total}$), do
  for each node i, do
    set distCH(node i; CH head) =Euclidean (node $i$, CH $head$)
if distCH(node $i$, CH $head$) ≤ Transmission Range, then
      CH head invites node i
    end
end
for each node $j$, do
  if $D_{euclidean}$≤Transmission Range, then
    set node $j$'s destination = $bstn$
  else
    for $head$ in len($CH_{total}$), do
      if CH $head$ invites node $j$, then
        if distCH(node $j$, CH $head$) ≤
        min($distCH(node\,j,\ :)$), then
        set node $j$'s destination = $head$
          create $neighbour$
          add node $i$ to Cluster $j$
          end
    end
for each node $j$, do
```

Algorithm 2: Cluster formation.

state of an agent can be defined as the state of the world and the action as a choice between performing one of a set of actions at a given time. The actions and states are stochastic. The dynamics of the stochastic process are governed by a set of probabilistic rules and parameters known as a policy, which can be represented as a function that maps state to probability of action. The reward is a scalar that measures how successful the agent is at following the policy. The RL agent aims to find a policy that maximizes the expected reward over time. This is the standard RL setting, which is applicable in an array of problems. The state is not explicitly given during the learning process but is inferred by the agent by performing some action and observing its effect on the state, which is called an episode. The episode terminates when the agent reaches its goal or some maximum number of steps is reached. The most common RL framework is Q -learning, which learns a value for the state that is a function of the action [14].

The value function is incremented when the agent performs an action in the state, which decreases the probability of performing it again. The value function becomes a map of how valuable the state is. The current state value is compared to the current value function to select a new state: the new state is chosen if the value of the new state is lower than the current state and the action for which the current state is the goal. A state value function is defined where $f\,(s\,;a)$ is a function of the state, a is the action selected, s is the current state, and r is the reward and is a discount factor that controls how much weight is given to the future.

In the real world, not all states are immediately observable to the agent, which leads to a problem generally referred as the credit assignment problem. Since the goal is to maximize the reward, the policy is "discounted" for the future. If the state is not visited during an episode, the Q-learning is reset to some baseline value. Learning is achieved by using the Bellman equation, which calculates the Q-reward received from the state-action pair under the current policy and uses it to update the Q-value for the state-action pair received under the new policy. This is then used to update the policy to the new state-action pair. Since this process is repeated over time, the policy iteratively improves. Also, the state value functions must be initialized arbitrarily to avoid the problem of learning bias. The RL algorithm is to maximize a sum of state values discounted by the length of the episode. A learning rate is used to adjust the learning rate to the agent's current performance, and the learning rate diminishes as the agent becomes more successful [15]. This makes sure that the agent is always trying to improve but does not use an unrealistically large learning rate.

To effectively use RL for routing decision problems, we need to clearly define the essential components of canonical RL problem in terms of WSN. Each device in the network is mapped as an agent, and the state space for each device is mapped as the collection of possible routes through its surrounding devices to the sink. The action space is defined as the collection of all feasible neighbours via which packets to the sink can be relayed, and the way the devices in the net network or the way the agents behave is defined as a policy. Mutombo et al. [16] and Mutombo et al. [17] suggested that the policy iteration is then used to evaluate and improve the given policy which maps the state-action pair, maximizing the long-term reward to get the best policy.

3. Methodology

3.1. ReLeC Protocol. By sharing local information with the neighborhood, the proposed ReLeC protocol allows devices to make more accurate routing decisions, allowing them to improve next-hop selection and lower energy consumption.

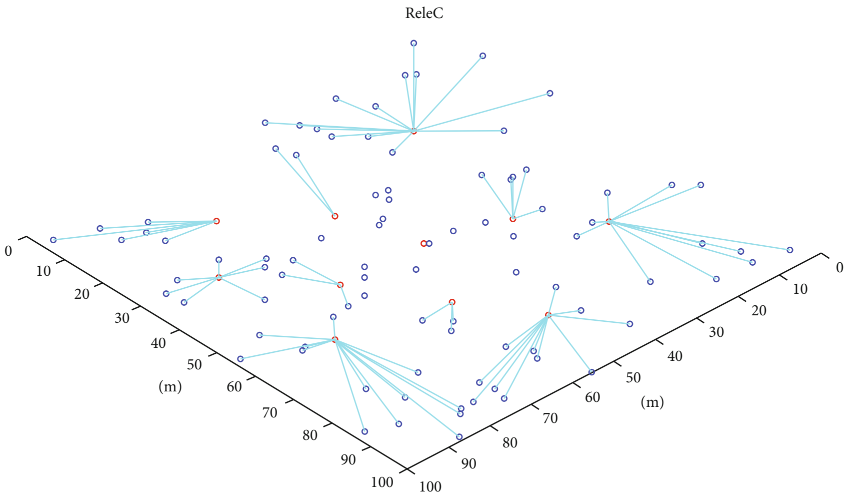


Figure 2: CH selection and cluster formation.

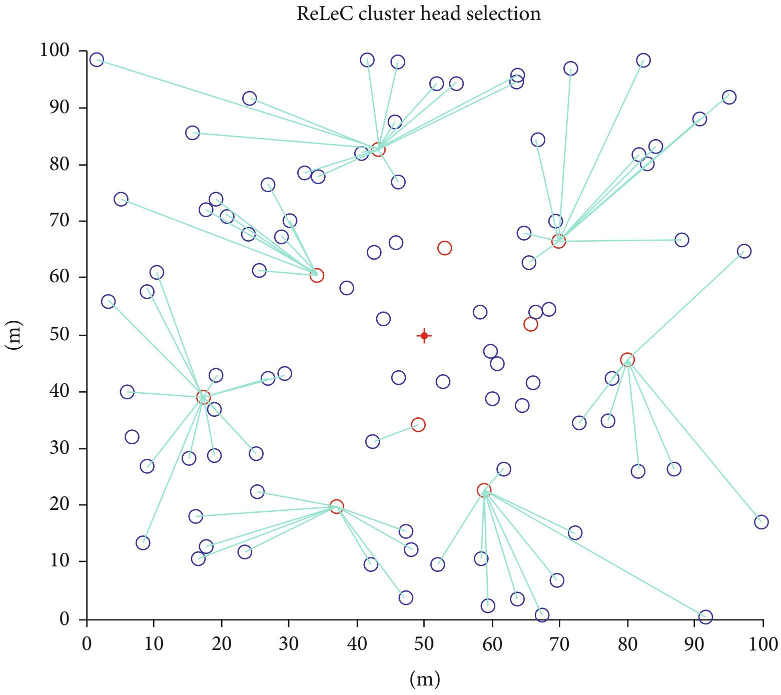


Figure 3: CH selection and cluster formation.

Routing tables of neighboring devices get modified as per the packet header contents of communicating device, whereas the sender inserts local data in the packet header. The local data sent includes ids, residual energy value, location coordinates, and N_H. Similar to other efficient, clustering-based routing protocols, ReLeC has three levels: network initialization, CH election and cluster creation, and communication phase.

3.2. Network Initialization. The set-up permits nodes of the network to determine the initial Q-value using local data. The base station then sends out a message to communicate its location coordinates. Individual nodes then keep the position of the base station after accepting the packet and use equations given to calculate the initial Q-value using residual, $E_{\min}$, $E_{\max}$, N_H, and probabilistic parameter p. This work proposes a modest extension to [16] by introducing Shannon-entropy inspired modification in finding initial Q -value. We also assume that all nodes have varying degrees of energy. To reduce network overhead, we establish a (distance) threshold, as a criterion between the cluster heads and base station, or sink and make it simpler for sensors placed distant from the base stations to identify a CH. Furthermore, a CH must not be on the network's edge to avoid connections diverging from a base station rather than converging, as this might result in energy waste due to the increased

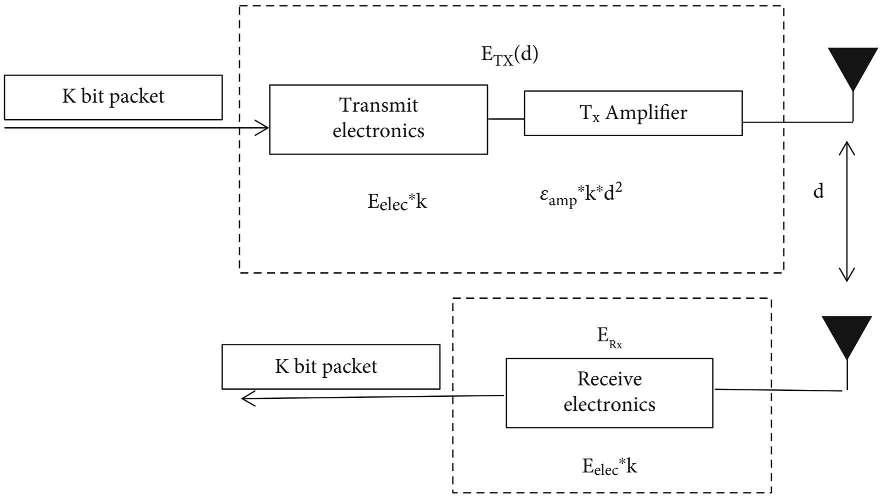


Figure 4: First-order radio model.

communication distance. The CH election procedure is represented by Algorithm 1. The initial Q-value is calculated as below.

$$Q = \begin{cases} \left(N_H \times \log\left(\frac{1}{N_H}\right)\right) & E_{\max} = E_{\min}, \\ \left(p \times \left(\frac{E_{\text{residual}}}{E_{\max} - E_{\min}}\right)\right) + \left((1-p) \times \left(N_H \times \log\left(\frac{1}{N_H}\right)\right)\right) & E_{\max} \neq E_{\min}. \end{cases} \tag{1}$$

The hop count (N_H) is approximated as the ratio of Euclidean distance to the transmission range, i.e., $N_H \approx D_{\text{euclidean}}$/transmission range, as discussed by Wang et al. [4].

3.3. Cluster-Head Election and Cluster Formation. Figure 1 has displayed the flowchart of network set-up and cluster-head selection.

Following the election phase, each CH transmits an invitation message to each and every device that falls into its transmission range, informing them that CH has been selected. The initial Q-value, the id, and the location coordinates are also included in the invitation. Every non-CH nodes determines the cluster it will join and sends a request to that particular CH, providing its data, depending on distance. Furthermore, if a node receives a number of invitations, i.e., if the situation rises where the node is at the joining point of multiple clusters, the node can choose the one with the closest CH [18]. Once all of the devices have sent requests, then it confirms that the membership is formed between clusters. The procedure of formation of clusters is explained in Algorithm 2.

Nodes in their transmission range with base station need not join a cluster; they communicate directly with the sink to conserve energy. ReLeC can also be used for different methods of communication such as intercluster and intracluster. Intercluster communication is concerned with multihop communication between CHs; for example, a CH located a far away from the sink can route the packets through CHs closer to the sink [19].

```
for each node i, do
  if E of node i >0, then
    set $Q_{max}$ = max(Q(node i; :))
    if $D_l$ Transmission Range, then
      if node i is to be the next hop, then
        collate data and send to bstn
      else
        send data to bstn
    else if node i is not a CH then
      if CH is within Transmission Range,
      then
        send data to CH
      else
        locate and send data to neighbour
    evaluate $R_{t+1}$
    update Q to $Q_{t+1}$ (s; a)
end
```

Algorithm 3: Data transmission.

When devices inside the cluster can transmit data either directly or via multihop to the CH, intracluster communication is comparable; a node can get connected with other devices in the cluster even if they are far away. Nodes have varying energy levels and transmission ranges, as previously indicated. The cluster formation is depicted in Figures 2 and 3.

3.3.1. Energy Consumption Model. To calculate the successive residual energy of each node after a round, the radio model is employed, given by Shah and Rabaey [10]. E_{elec} = 50 nJ/bit gets dissipated to power the transmitter/receiver, and $\varepsilon_{\text{amplification}}$ = 100 pJ/bit/m^2 is a proportionality constant for the power consumption in the message amplification

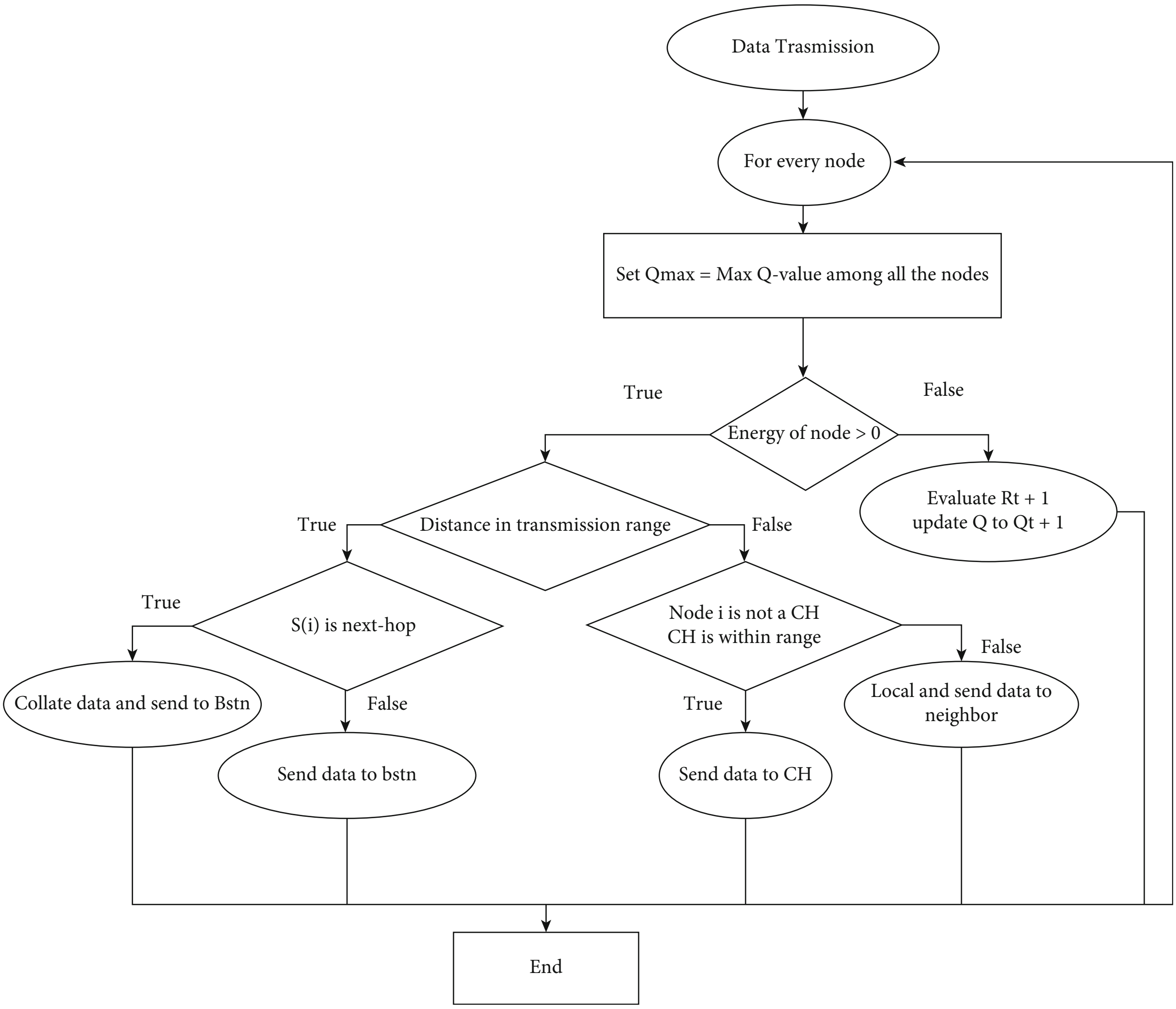


Figure 5: Flow chart of data transmission.

over a distance d. For transmission, the energy dissipated is computed with the following formula:

$$E_{\text{transmission}}(k, d) = E_{\text{elec}} \times k + \varepsilon_{\text{amplification}} \times k \times d^{w}, \quad (2)$$

where $w = 2$ or 4.

Similarly, for reception of the k-bit message over a distance d, $E_{\text{reception}}(k)$ is expended, which can be computed using the following formula:

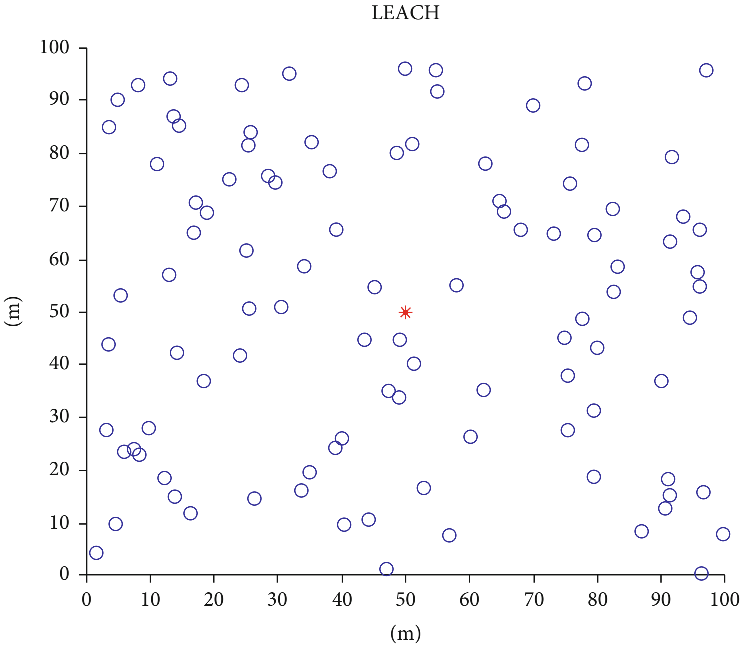
$$E_{\text{reception}}(k) = E_{\text{elec}} \times k. \quad (3)$$

First-order radio model is presented in Figure 4.

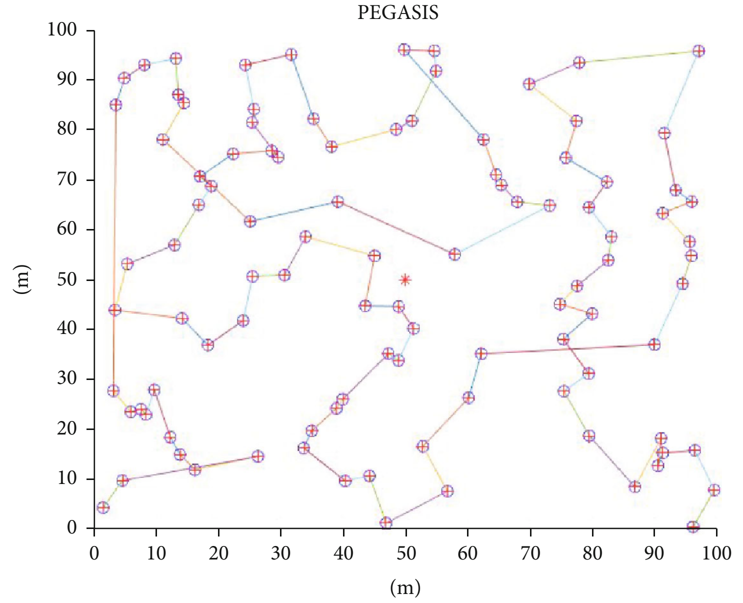
Table 1: Parameters for simulation.

Parameters	Values
Sensing field dimensions	100 × 100
Number of agents/devices	50-250
Transmission range	20 m
E_0	1-2 J
Data	4000 bits
E_{elec}	50×10^{-9} J/bit
E_{amp}	100×10^{-12} J/bit/m^2
γ	0.95
α	1

3.4. Communication Phase. The energy consumption model described above provides us with the updated value of E_{residual}; the residual energy of the node after subtracting the energy dissipated during package transmission. These updated values of E_{residual} and N_H are used to compute R_{t+1}, the next reward, using the reward function discussed at depth in the coming sections. Over time, each node, acting as an agent, learns as it updates the Q-value using rewards obtained by performing successive actions, to sequentially give a better routing strategy. The key equations to this are

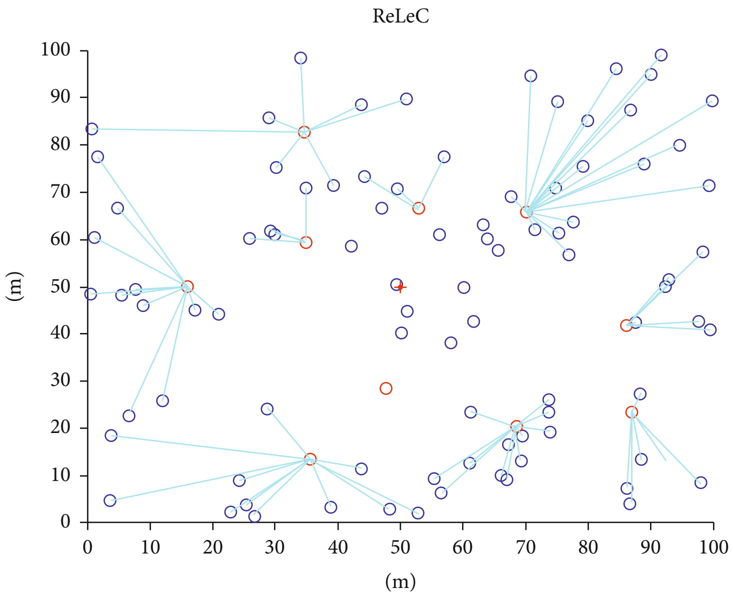


(a)



(b)

Figure 6: Continued.



(c)

Figure 6: (a) LEACH node deployment. (b) PEGASIS node deployment. (c) ReLeC cluster formation.

the reward function and the update rule described in this section.

3.4.1. Application of Reinforcement Learning. WSNs can be used to monitor temperatures in different areas of forests to prevent disasters like forest fires. The data can be analyzed using hybrid model, RFVR, a combination of support vector machine and random forest regression by Bhadoria et al. These new models have achieved high accuracy with low variance and can be used in conjunction with WSN to prevent forest fires [20].

Unlike the phase before, to calculate the hop count, D_l (between sender and neighbour nodes, respectively) is used instead of simple Euclidean distance, $D_{\text{euclidean}}$. D_l can be calculated as follows:

$$D_1 = D_{i,j} + D_{j,\text{bstn}}. \tag{4}$$

Here, D_l is the distance between the sender and the neighbour node.

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

$D_{i,j}$ is the distance between the ith and the jth node, and it is calculated using equation (5).

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

$D_{j,\text{bstn}}$ is the distance between the jth node and the base station; it is calculated using equation (6).

Further, the data transmission is done as per Algorithm 3 [19, 21].

The flow chart for data transmission is shown in Figure 5.

3.4.2. Reward Function and Update Rule. A key step in reinforcement learning is the selection of the reward function. We must define a function that captures the goal of the learning process. In our model, we use an energy and hop-count weighted sum as the reward function. p is the probabilistic parameter for E_{residual}, and q is the probabilistic parameter for the competing factor N_H: the trade-off between the two probabilistic parameters helps optimize enhancing the performance of the protocol as a greater p favours a node with higher residual energy to be next hop whereas q favours a closer node to be the next hop. The competing efforts result in the next hop to be closer to the current node and possessing high residual energy.

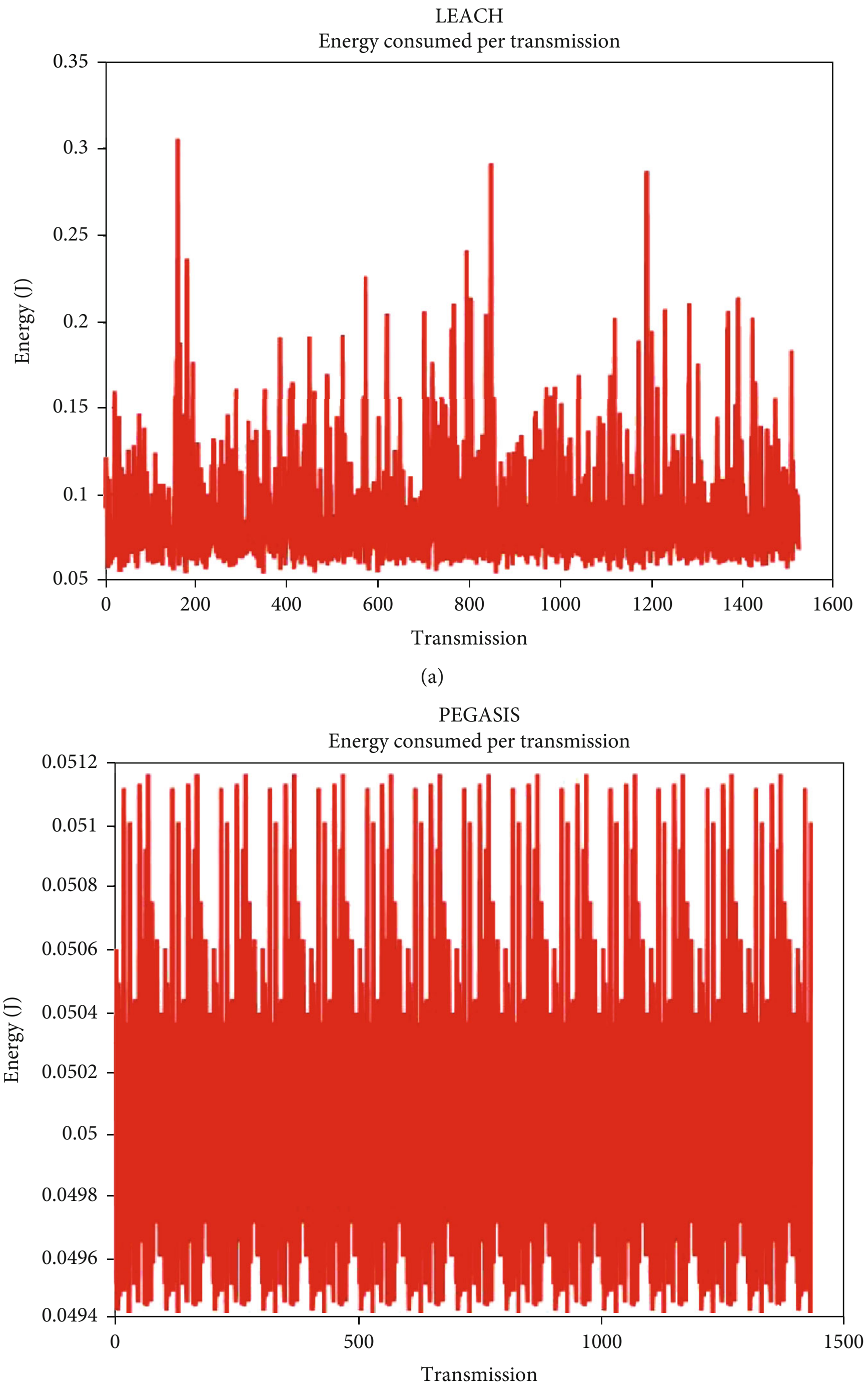
If E_{residual} is 0, a negative reward is assigned to the then prospective next hop and finds another next-hop node. The next hop follows the same procedure until the message is received at the base station while continuously sending feedbacks.

R_{t+1} (next reward) is calculated using the following equation:

$$R_{t+1} = \begin{cases} \left(N_H \times \left(\dfrac{1}{N_H}\right)\right) & E_{\max} = E_{\min}, \\ \left(p \times \left(\dfrac{E_{\text{residual}} - E_{\min}}{E_{\max} - E_{\min}}\right)\right) + \left(q \times \left(N_H \times \left(\dfrac{1}{N_H}\right)\right)\right) & E_{\max} \neq E_{\min}, \\ -100 & E_r \leq 0, \end{cases} \tag{7}$$

where $q = 1$ and $p = q - 1$.

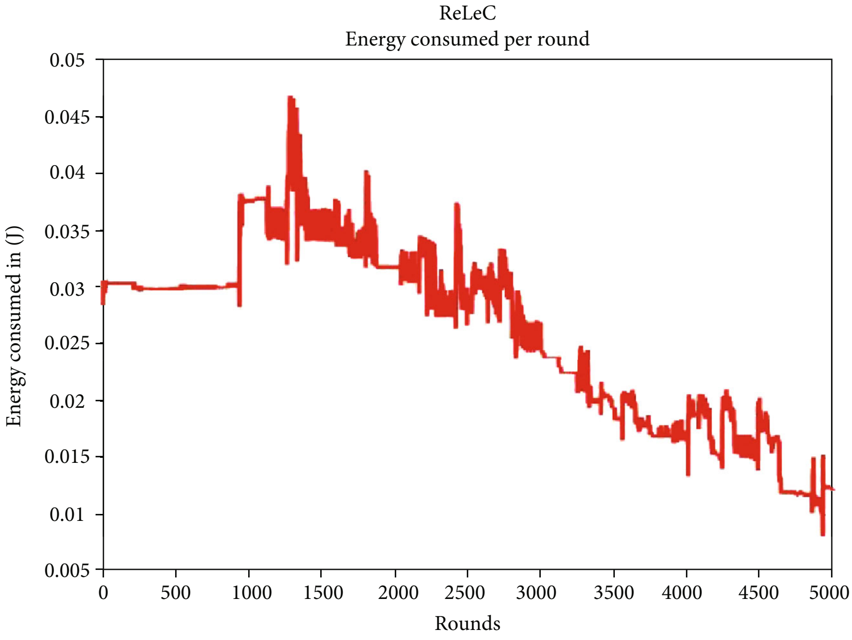
The updated Q-value is finally used as an argument in the policy function for updating the policy, which finds the



(a)

(b)

Figure 7: Continued.



(c)

Figure 7: (a) Energy consumption of LEACH protocol. (b) Energy consumption of PEGASIS protocol. (c) Energy consumption of ReLeC protocol.

best policy. It is calculated using equation (8). The above methods of policy and update are given below [14].

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

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

G_t calculates the discounted reward using equation (9), where k is the discount factor.

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

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

$$\begin{aligned} Q_{t+1}(s, a) &= (1 - \alpha) Q_t(s, a) \\ &\quad + \alpha\left(R_{t+1}(s, a) + \gamma \max Q\left(S', a\right)\right). \end{aligned} \tag{12}$$

Equations (10)–(12) show the equations used and update rule to get the best policy [14, 22].

The model, reward function, and update rule are updated with the calculated values, but the policy is updated with the best value at that point of time. The update is performed at the end of every step, and the new values are stored as the Q-values.

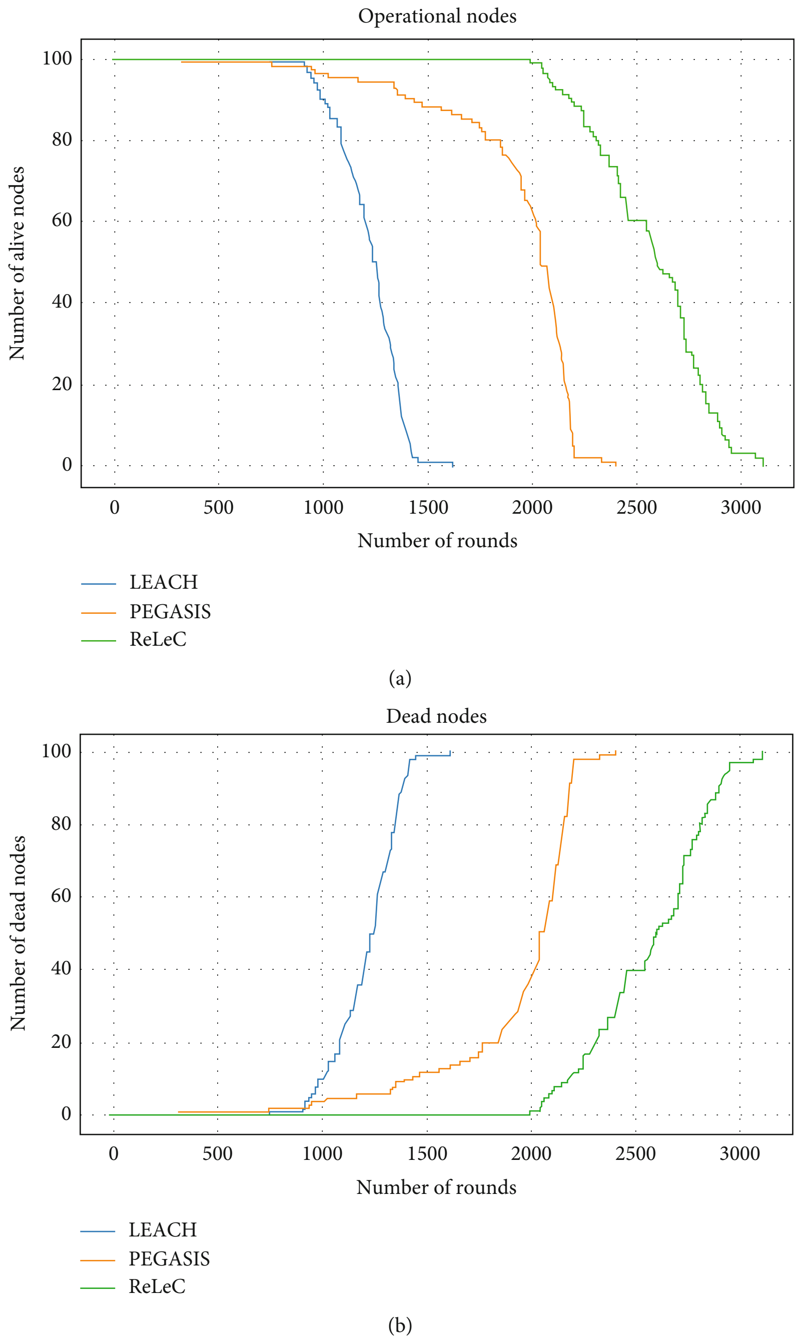
4. Performance Evaluation

To measure ReLeC's performance, we used MATLAB to run a simulation in which 100 nodes were dispersed over a field of 100*100 m in a randomized manner. With (50, 50) coordinates, in the sensing field, the base station was put in the centre. Furthermore, we expected that the network would be heterogeneous, with devices ranging in energy from 1 to 2 joules. Table 1 summarises the parameters taken.

4.1. Simulation Parameter Tuning. The proposed procedure, as previously stated, analyzes both numbers of hops and remaining energy, and probabilistic values such as p and q $= 1\,p$ have been allocated to both remaining energy and number of hops. A larger value of p gives more weightage to nodes having higher energy. A large value of q, on the other hand, enhances the chances of nodes with fewer hops to the base station being chosen. As a result, to improve the performance of ReLeC, we experimented with various values of these parameters to find the optimum ones. With varying p and q, the performance evaluation yielded slightly varied results. However, both the probabilistic parameters are equal, the network lifetime is improved while the energy balance remains favorable. p equals 0.4 and q equals 0.6 can also obtain similar outcomes in some cases.

4.2. Network Lifetime and Energy Efficiency Evaluation. The fundamental goal of the research is to improve the efficiency and network longevity. When the data transmission becomes no longer possible is termed as the lifetime of the network. By contrasting the proposed approach with existing clustering techniques like LEACH and PEGASIS, we assessed its energy efficiency and network longevity. For comparison, we used the following metrics:

(1) The number of sensor nodes that are alive each round; this parameter also helps examine the network's lifespan



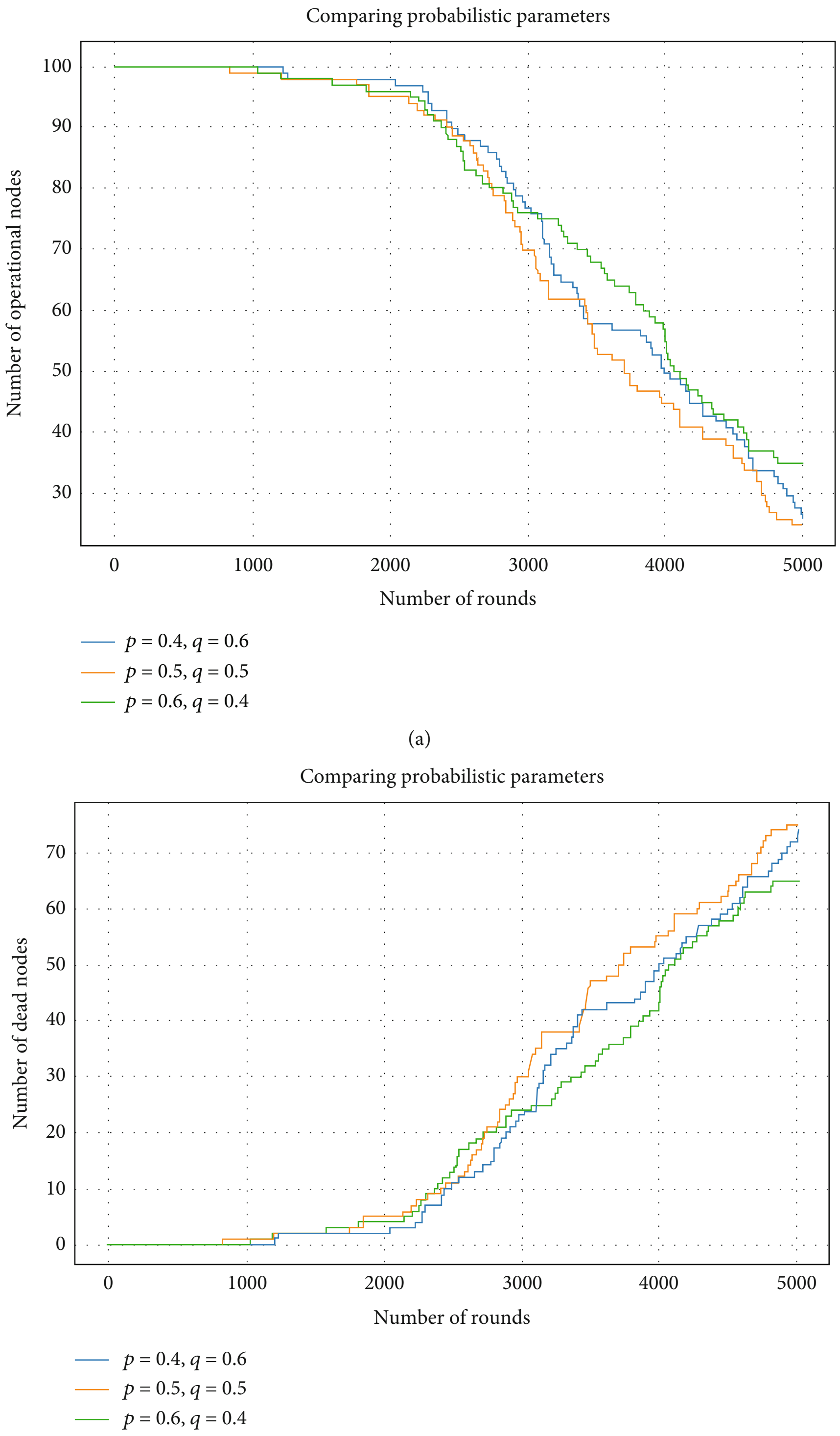
(a)

(b)

Figure 8: (a) Comparing the alive nodes for 3000 rounds. (b) Comparing the dead nodes for 3000 rounds.

Table 2: Comparison of LEACH, PEGASIS, and ReLeC.

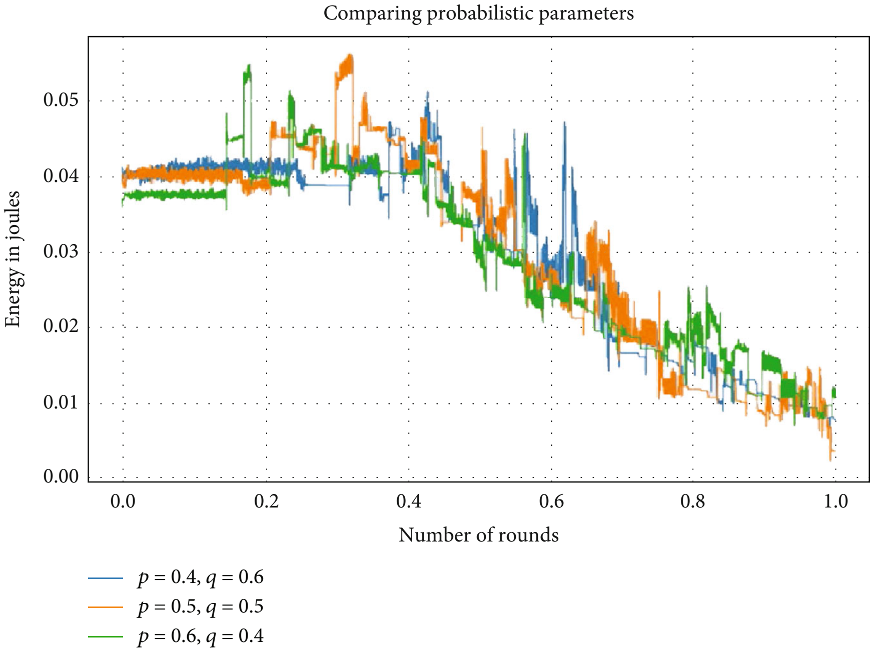
Parameters	LEACH	PEGASIS	ReLeC
Energy consumption range (J)	0.05 J-0.3 J	0.0494 J-0.0512 J	0.005 J-0.05 J
Network lifespan (rounds)	1662	2427	3130
Average stability period	825	1915	1150



(a)

(b)

Figure 9: Continued.



(c)

Figure 9: (a) Number of alive nodes at different rounds of iterations for ReLeC. (b) Number of dead nodes at different rounds of iterations for ReLeC. (c) Energy consumption at different rounds of iterations for ReLeC.

(2) The energy expended per round. It is the amount of all the devices' energy consumed per round

(3) The time frame in which the first node dies. These are the parameters that are used to estimate energy efficiency

Figure 6(a) displays the graph based on LEACH node deployment whereas Figure 6(b) shows the PEGASIS node deployment and Figure 6(c) has shown the cluster formation of ReLeC.

Figures 7(a)–7(c) show the energy consumption of LEACH, PEGASIS, and ReLeC. It is clear from the figures that ReLeC shows the best results in terms of energy consumption.

Using several sensor nodes, we measured the network lifetime of the proposed protocol in the figure mentioned below. We also compared LEACH and PEGASIS to show that the proposed protocol is effective. ReLeC outperforms LEACH and PEGASIS with both 30 and up to 100 devices. With a more extended network lifetime, the suggested protocol outperforms current protocols. In order to optimize network longevity, we considered both residual energy and hop count. When transmitting data, the power can be higher if the distance is too greater. In Figures 8(a) and 8(b), we compared the stability region; the suggested approach outperformed LEACH and PEGASIS in all test conditions. We can conclude from the foregoing that the protocol proposed performs better in a large-scale network than these canonical approaches. Still, the LEACH and PEGASIS protocol performs better in a small-scale network (less than 50 devices).

Table 3: ReLeC algorithm assessment at different values of probabilistic parameters.

Parameters	$p = 0.4$ $q = 0.6$	$p = 0.5$ $q = 0.5$	$p = 0.6$ $q = 0.4$
Operational nodes	15	13	35
Average stability period (rounds)	3850	4100	3985

Table 2 has shown the comparison of models based on various parameters. We have analyzed the number of alive nodes at different rounds of iterations, energy consumption, and number of dead nodes for ReLeC at different values of p and q.

Figure 9 shows the number of alive nodes at different rounds of iterations for ReLeC, and it is clear from the figure that $p = 0.5$ and $q = 0.5$ shows the best results.

Figure 9 shows the energy consumption for ReLeC, and evidently, $p = 0.6$ and $q = 0.4$ gives the best results.

Figure 9 depicts the number of alive nodes at different rounds of iterations for ReLeC and the values $p = 0.6$ and $q = 0.4$, respectively, lead to the desired results.

Table 3 represents the ReLeC algorithm assessment based on different values of probabilistic parameters.

5. Conclusion

This paper proposes ReLeC protocol, a reinforcement learning-based, clustering-enhanced strategy for energy-efficient routing in WSNs and potentially other spatially dispersed IoT networks. The protocol strives to find an effective data transmission route using clustering and RL. This was done in three phases: first being network initialization and preliminary setup followed by CH election on the basis of the initial energy of devices and hop count. The second phase consisted of cluster formation for efficient inter- and intracluster communication on transmission range-based invitation. The third and final phase is the data transmission or communication phase, which is learning-driven based on reinforcement learning. The proposed protocol ReLeC outperformed LEACH by the percentage of 88.32% and PEGASIS by the percentage of 28.9% in terms of network lifespan for 3000 rounds.

6. Future Work and Limitations

As mentioned under limitations section, careful testing followed by analysis needs to be done for assessing the performance and computational viability of the protocol. Additionally, to make the protocol more scalable, the protocol can be optimized to work on the "edge," to get a new generation of intelligent, edge-computing-enabled wireless sensor networks. The major limitation of the presented work is real-world application. For this study, we ran simulations with varying parameters like sensing field dimensions, range, and number of sensor nodes. Tests need to be performed to analyze the protocols' performance in the real world and find other areas of improvement.

Data Availability

The code has been implemented using MATLAB. The MATLAB code for the proposed work is available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

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

[2] R. S. Bhadoria, N. Bhoj, H. G. Zaini et al., "Artificial intelligence for creating low latency and predictive intrusion detection with security enhancement in power systems," *Applied Sciences*, vol. 11, no. 24, p. 11988, 2021.

[3] A. W. Matin and S. Hussain, "Intelligent hierarchical cluster-based rout-ing," *Life*, vol. 7, p. 8, 2006.

[4] S. Wang, J. Yu, M. Atiquzzaman, H. Chen, and L. Ni, "CRPD: a novel clustering routing protocol for dynamic wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 22, no. 3, pp. 545–559, 2018.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro-sensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, p. 10, Maui, HI, USA, 2000.

[6] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: a stable election protocol for clustered heterogeneous wireless sensor networks," in *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, Boston, MA, 2004.

[7] C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor in-formation systems stephanie lindsey," *Work*, vol. 310, pp. 336–1686, 2001.

[8] S. Chand, S. Singh, and B. Kumar, "Heterogeneous heed protocol for wireless sensor networks," *Wireless Personal Communications*, vol. 77, no. 3, pp. 2117–2139, 2014.

[9] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, Boston Massachusetts, 2000.

[10] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *2002 IEEE wireless communications and networking conference record. WCNC 2002 (cat. No.02TH8609)*, pp. 350–355, Orlando, FL, USA, 2002.

[11] M. Littman and J. Boyan, "A distributed reinforcement learning scheme for network routing," in *Proceedings of international workshop on applications of neural networks to telecommunication*, pp. 55–61, Psychology Press, 2013.

[12] 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.

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

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, 2018.

[15] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learn-ing: a survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[16] V. K. Mutombo, S. Lee, J. Lee, and J. Hong, "EER-RL: energy-efficient routing based on reinforcement learning," *Mobile Information Systems*, vol. 2021, Article ID 5589145, 12 pages, 2021.

[17] V. K. Mutombo, S. Y. Shin, and J. Hong, "EBR-RL: energy balancing routing protocol based on reinforcement learning for wsn," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing, Association for Computing Machinery*, New York, NY, USA, 2021.

[18] M. Hajjar, G. Aldabbagh, and N. Dimitriou, "Using clustering techniques to improve capacity of lte networks," in *2015 21st Asia-Pacific Conference on Communications (APCC)*, pp. 68–73, Kyoto, Japan, 2015.

[19] A. Hassan, Y. Zhao, L. Pu, G. Wang, H. Sun, and R. M. Winter, "Evaluation of clustering algorithms for dap placement in wireless smart meter network," in *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1085–1090, Kunming, China, 2017.

[20] R. S. Bhadoria, M. K. Pandey, and P. Kundu, "*RVFR* : random vector forest regression model for integrated & enhanced approach in forest fires predictions," *Ecological Informatics*, vol. 66, p. 101471, 2021.

[21] X. Liang, I. Balasingham, and S. S. Byun, "A multi-agent reinforcement learning based routing protocol for wireless sensor networks," in *2008 IEEE International Symposium on Wireless Communication Systems*, pp. 552–557, Reykjavik, Iceland, 2008.

[22] X. Liang, I. Balasingham, and S. S. Byun, "A reinforcement learning based routing protocol with qos support for biomedical sensor networks," in *2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies*, pp. 1–5, Aalborg, Denmark, 2008.