



University of Bahrain
College of Engineering
Department of Electrical and Electronics Engineering

Reinforcement Learning for Intelligent Wireless Networks

A Thesis Submitted in Partial Fulfillment of the Requirements for the
degree of Master of Science in Artificial Intelligence System

Submitted by: Najeeba Mohammed
Student ID: 20112695

Supervised by Dr. Mohab Mangoud

May 2024



University of Bahrain
College of Engineering
Department of Electrical and Electronics Engineering
Master of Science in Artificial Intelligent Systems

Defense Committee's Approval

Student **Najeeba Mohammed Jaafar** ID number **20112695** has defended her thesis titled “**Reinforcement Learning for Intelligent Wireless Networks**” by which she has fulfilled the requirement for the degree of Master of Science in Artificial Intelligent Systems on **Thursday, May 30, 2024**.

The committee recommends awarding of M.Sc. Degree with the grade of: **Pass.**

(Defense Committee)

Name		Signature
Prof. Mohab Mangoud	Supervisor	
Dr. Wael Elmedany	Internal examiner	
Dr. Ahmed Yousif	External examiner	<i>Ahmed Hassan Goussef</i>

Abstract

The growing presence of Internet of Things (IoT) devices requires efficient and reliable data transmission protocols. However, traditional routing protocols struggle in dynamic and resource constrained IoT networks due to factors like decentralization, limited energy, and frequent topology changes. This thesis proposes a novel routing protocol, modified Energy-Efficient Routing based on Reinforcement Learning (mEER-RL), specifically designed to address these challenges and enhance performance in both fixed and mobile IoT deployments. mEER-RL is built over the foundation of EER-RL, a cluster-based, RL-driven protocol focused on energy efficiency. mEER-RL contains a significant modification to address node mobility, a crucial factor often neglected in existing cluster-RL based protocols for IoT. By integrating mobility considerations into the RL agent's reward function, mEER-RL dynamically adapts its routing decisions to network conditions, improving Packet Delivery Ratio (PDR) and overall network reliability. This thesis presents a survey of most types of cluster-based routing protocols built for IoT networks. RL techniques are also reviewed to be applied for the routing protocol. The thesis then comprehensively evaluates mEER-RL's performance through simulations, comparing it against the established base protocol LEACH and EER-RL. The evaluation focuses on crucial metrics like network lifetime, energy consumption, packet delivery ratio, and the impact of node mobility. The results demonstrate that mEER-RL effectively balances energy efficiency, reliability, and adaptability, making it a promising solution for routing in complex and dynamic IoT networks. The key contributions of this work include enhanced reliability, increased PDR, enhanced adaptability, and improved energy efficiency for both fixed and mobile IoT environments by optimizing routing paths and minimizing data transmission through improved multi-hop communication techniques. For an advanced performance, compared to LEACH and EER-RL, mEER-RL exhibits superior performance in terms of network lifetime, energy consumption, and PDR.

Dedication

To my family and friends, whose unwavering belief and proud eyes fueled my determination to see this project through. Also, to my own resilience, for enduring the challenges and anxieties that come with such a significant accomplishment.

Acknowledgment

This thesis would not have been possible without the precious support of several individuals and resources. I would like to express my sincere gratitude to my supervisor Prof. Mohab for his assistance with securing the resources necessary for this research. His arrangements for a powerful device and insightful recommendations proved to be helpful in the successful completion of this project.

Furthermore, I am deeply grateful to my management in Batelco for their understanding and flexibility in allowing me to reschedule my duties to manage my time. This enabled me to dedicate the necessary focus and effort to this research.

Finally, I would like to acknowledge the work done on the EER-RL protocol, which served as a valuable foundation for the development of my proposed mEER-RL protocol. The initial code and research on EER-RL provided a strong starting point for my own exploration.

List of Content

Defense Committee’s Approval.....	B
Abstract	C
Dedication.....	D
Acknowledgment.....	E
List of Content.....	F
Abbreviations	1
List of Tables	4
List of Figures	6
Chapter1: Introduction	10
1.1 Motivation and Objectives	11
1.1.1 The Growth of Complexity in IoT Networks	12
1.1.2 Reinforcement Learning for Adaptive Routing in IoT	13
1.2 Contribution of This Work.....	14
1.3 Thesis Structure.....	15
1.4 Chapter Summary.....	16
Chapter 2: Background and Literature Review	18
2.1 Architecture and Components of IoT System	19
2.1.1 IoT Network Layer and Routing Protocols Classification	23
2.1.2 IoT Routing Challenges.....	28
2.1.3 Mobility Challenges	31
2.1.4 Data Routing Metrics	32
2.2 Clustering Methods in IoT Networks.....	34
2.2.1 Network Clustering Advantages.....	36
2.2.2 Network Clustering Approaches	37
2.3 Reinforcement Learning Overview	42
2.3.1 Reinforcement Learning Applications in IoT: A Review	42
2.3.2 Reinforcement Learning Technique	45

2.4 Cluster-Based Routing Protocols for IoT	52
2.4.1 Classical Approaches.....	53
2.4.2 Optimization Approaches.....	56
2.4.3 Machine Learning Approaches.....	60
2.5 Chapter Summary.....	64
Chapter 3: Comparative Performance Evaluation of Existing Protocols	66
3.1 Performance Evaluation Metrics.....	67
3.2 LEACH Model	68
3.2.1 Energy Consumption Model.....	68
3.2.2 Advertisement and Cluster Set-Up Phase.....	69
3.2.3 Schedule Creation and Transmission Phase	70
3.3 EER-RL Model	70
3.3.1 Network Set-Up and CH Election Phase.....	70
3.3.2 Cluster Formation Phase.....	72
3.3.3 Data Transmission Phase.....	72
3.4 Protocols Simulation and Evaluation	75
3.5 Chapter Summary.....	81
Chapter 4: mEER-RL: A Modified EER-RL Protocol for Enhanced Fixed and Mobile IoT Networks Performance	82
4.1 CH Election Method.....	83
4.2 Consideration of Distance Threshold Between CHs.....	85
4.3 Alive and Dead Nodes Parameters Update	89
4.4 Re-clustering Phase Rearrangement.....	90
4.5 Intra and Inter-Cluster Multihop Communication	93
4.6 Dynamic Learning.....	100
4.7 Modified Q-function and Reward: Balancing Energy and Reliability.....	102
4.8 Node Mobility Consideration in mEER-RL.....	103
4.9 Chapter summary	108
Chapter 5: mEER-RL Simulation Results and Discussion.....	109
5.1 mEER-RL Simulation Setup	110
5.2 mEER-RL Simulation Results and Performance Evaluation.....	112

5.2.1 CH Election and Nodes Parameters Update	112
5.2.2 CH Distance Threshold Formula	118
5.2.3 Mitigating Packet Drops and Duplications in mEER-RL-v2	127
5.2.4 Modified Q-function and Reward: Balancing energy Efficiency and Reliability in mEER-RL	130
5.2.5 Dynamic Learning	133
5.2.6 Mobile mEER-RL	140
5.3 Performance Comparison: Fixed vs. Mobile mEER-RL and Existing Protocols	147
5.3.1 RL-based Protocol Parameters	147
5.3.2 Network Setup and Clustering.....	148
5.3.3 Network Lifetime and Energy Consumption Evaluation	151
5.3.4 Distance Traveled Evaluation.....	156
5.3.5 Packet Delivery Evaluation	158
5.3.6 Results Summary and Discussion of mEER-RL Performance.....	161
5.4 Chapter summary	162
Chapter 6: Conclusion and Future Work.....	164
References	168

Abbreviations

IoT	Internet of Things
WSN	Wireless Sensor Network
AI	Artificial Intelligence
RL	Reinforcement Learning
mEER-RL	modified Energy-Efficient Routing based on Reinforcement Learning
EER-RL	Energy-Efficient Routing based on Reinforcement Learning
LEACH	Low-Energy Adaptive Clustering Hierarchy
CH	Cluster Head
BS	Base Station
PDR	Packet Delivery Ratio
RFID	Radio Frequency Identification
IP	Internet Protocol
SoA	Service-oriented Architecture
QR	Quick Response
NFC	Near-Field Communication
QoS	Quality of Service
CM	Cluster Member
EAs	Evolutionary Algorithms
GA	Genetic Algorithm
LBCA	Load-Balanced Clustering Algorithm
GLBCA	Greedy Load-Balanced Clustering Algorithm
BFS	Breadth-First Search
SI	Swarm Intelligence
ACO	Ant Colony Optimization
FA	Firefly Algorithm

ABC	Artificial Bee Colony
PSO	Particle Swarm Optimization
BFO	Bacterial Foraging Optimization
ML	Machine Learning
SL	Supervised Learning
DT	Decision Tree
RF	Random Forest
ANN	Artificial Neural Network
DL	Deep Learning
SVM	Support Vector Machine
UL	Unsupervised Learning
QC	Query Controller
QP	Query Processors
DG	Distributed Generation
EES	Electrical Energy Storage
MDP	Markov Decision Processes
POMDP	Partially Observable MDP
Alpha (α)	Learning rate
Gamma (γ)	Discount factor
SARSA	State, Action, Reward, State, Action
WBSN	Wireless Body Sensor Networks
MCH	Main Cluster Head
SDN	Software Defined Networks
EC	Energy Consumption
EED	End-to-End Delay
FND	First Node Dies
HNA	Half of the Nodes Alive
LND	Last Node Dies
AE	Average Energy

NS	Near-Sink
MCE	Maximum Consumed Energy
ACE	Average Consumed Energy
Epsilon (ϵ)	Decreasing factor

List of Tables

Table 1 The layers of the most common IoT architectures	21
Table 2 Summary of metrics for WSN according to Yuan, Kanhere & Hollick [32]	33
Table 3 Classical cluster-based routing protocols in IoT networks	55
Table 4 Optimization approaches for cluster-based routing protocols in IoT networks.....	59
Table 5 ML approaches for cluster-based routing protocols in IoT networks	63
Table 6 Parameters of LEACH & EER-RL simulation setup	75
Table 7 Simulation results of LEACH & EER-RL	78
Table 8 Results of different network input parameters to calculate CH distance threshold using equation (4.2)	89
Table 9 Simulation parameters for mEER-RL evaluation	110
Table 10 Protocols performance for evaluating CH election and distance threshold consideration while accurately updating the parameters, 5000 rounds	118
Table 11 Protocols performance for evaluating mEER-RL CH Dthreshold formula, 5000 rounds	121
Table 12 Protocols performance for evaluating mEER-RL CH fixed and Dthreshold formula compared to original EER-RL, 5000 rounds and 15% CH	124
Table 13 Protocols performance for evaluating updated mEER-RL performance for Multihop advanced and packet duplication prevention, 5000 rounds	130

Table 14 Probabilistic values of mEER-RL Q-value and Reward functions including PDR	131
Table 15 Protocols performance results for evaluating mEER-RL-v3 performance for the new Q-function and reward, 5000 rounds.....	133
Table 16 ϵ values for dynamic learning simulation	134
Table 17 Protocol performance results for evaluating mEER-RL using different ϵ values for dynamic learning effect	134
Table 18 Protocols performance results for evaluating mEER-RL Alpha reset strategies for dynamic learning effect	137
Table 19 Protocol performance results for evaluating mobile mEER-RL using different ϵ values for dynamic learning effect.....	144
Table 20 Protocols performance results for evaluating mEER-RL Alpha reset strategies for dynamic learning effect	146
Table 21 Simulation parameters for RL based protocols	148
Table 22 Network lifetime and energy evaluation metrics for fixed and mobile mEER-RL performance comparison with existing protocols....	154
Table 23 Traveled distance evaluation metric for fixed and mobile mEER-RL performance comparison with existing protocols	158
Table 24 Packet delivery evaluation metric for fixed and mobile mEER-RL performance comparison with existing protocols	160

List of Figures

Figure 1 Basic IoT 3-layer structure.....	20
Figure 2 Basic sensor network connected to internet forming IoT.....	23
Figure 3 IoT Routing Protocols Classification	25
Figure 4 IoT routing challenges	28
Figure 5 Clustering vs Non-clustering nodes to BS communication in sensor-based networks	35
Figure 6 CH selection methods in clustering.....	36
Figure 7 Clustering advantages for IoT routing optimization	37
Figure 8 Reinforcement Learning state-action-reward model [8]	48
Figure 9 Network setup and cluster formation using both LEACH (a) and EER-RL (b) protocols	77
Figure 10 Simulation observations of LEACH and EER-RL in 5000 rounds	79
Figure 11 CH re-election process during communication phase.....	91
Figure 12 Simplified cluster formation process in mEER-RL	93
Figure 13 mEER-RL Intra-cluster Multihop communication	95
Figure 14 mEER-RL Inter-cluster Multihop communication	97
Figure 15 mEER-RL CH within BS range communication	99
Figure 16 Modified inter-cluster communication condition for mobile mEER-RL	106
Figure 17 mEER-RL mobile nodes location and distance re-clustering trigger	107

Figure 18 Protocols cluster formation during the setup phase comparing mEER-RL CH election method.....	113
Figure 19 Operating and dead nodes per round for evaluating CH election and distance threshold consideration while accurately updating the parameters, 5000 rounds	115
Figure 20 Consumed energy and traveled distance per round for evaluating CH election and distance threshold consideration while accurately updating the parameters, 5000 rounds	116
Figure 21 mEER-RL performance with fixed and CH Dthreshold formula, 5000 rounds and 10% CH	119
Figure 22 Operating nodes and energy consumed per round for evaluating mEER-RL CH Dthreshold formula, 5000 rounds	120
Figure 23 Total sent packets per round for evaluating mEER-RL CH Dthreshold formula, 5000 rounds	121
Figure 24 mEER-RL performance with fixed and Dthreshold formula compared to original EER-RL, 5000 rounds and 15% CH	123
Figure 25 Operating nodes and energy consumed per round for evaluating mEER-RL CH fixed and Dthreshold formula compared to original EER-RL, 5000 rounds and 15% CH	125
Figure 26 Total sent packets per round for evaluating mEER-RL CH fixed and Dthreshold formula compared to original EER-RL, 5000 rounds and 15% CH	126
Figure 27 Operating nodes and total sent packets per round for evaluating updated mEER-RL performance for Multihop advanced and packet duplication prevention, 5000 rounds	128

Figure 28 Energy consumption and traveled distance per round for evaluating updated mEER-RL performance for Multihop advanced and packet duplication prevention, 5000 rounds	129
Figure 29 Operating nodes and total sent packets per round for evaluating mEER-RL-v3 for the new Q-function and reward performance, 5000 rounds	132
Figure 30 Operating nodes and total sent packets per round for evaluating mEER-RL using different ϵ values for dynamic learning effect	135
Figure 31 Energy consumption per round for evaluating mEER-RL using different ϵ values for dynamic learning effect	136
Figure 32 Operating nodes and total sent packets per round for evaluating mEER-RL Alpha reset strategies for dynamic learning effect	138
Figure 33 Energy consumption per round for evaluating mEER-RL Alpha reset strategies for dynamic learning effect	139
Figure 34 First and last round nodes location for mobile mEER-RL, with different ϵ values.....	141
Figure 35 Operating nodes and total sent packets per round for evaluating mobile mEER-RL using different ϵ values for dynamic learning effect	142
Figure 36 Energy consumption per round for evaluating mobile mEER-RL using different ϵ values for dynamic learning effect.....	143
Figure 37 Operating nodes and total sent packets per round for evaluating mobile mEER-RL Alpha reset strategies for dynamic learning effect ..	145
Figure 38 Energy consumption per round for evaluating mobile mEER-RL Alpha reset strategies for dynamic learning effect.....	146

Figure 39 Formed clusters by each protocol in network setup phase, for fixed and mobile mEER-RL performance comparison with existing protocols	150
Figure 40 Operating nodes per round for fixed and mobile mEER-RL performance comparison with existing protocols	151
Figure 42 Energy consumption per round for fixed and mobile mEER-RL performance comparison with existing protocols	153
Figure 43 Traveled distance per round for fixed and mobile mEER-RL performance comparison with existing protocols	157
Figure 44 Total number of sent packets per round for fixed and mobile mEER-RL performance comparison with existing protocols.....	159

Chapter1: Introduction

1.1 Motivation and Objectives

1.1.1 The Growth of Complexity in IoT Networks

1.1.2 Reinforcement Learning for Adaptive Routing in IoT

1.2 Contribution of This Work

1.3 Thesis Structure

1.4 Chapter Summary

The presence of wireless devices and applications has fundamentally become a part of our lives, constantly evolving to offer even wider benefits. At the lead of this transformation lies the Internet of Things (IoT) [1], which is mainly the point of focus to deliver the experience of being connected and have live interactions with remote devices. IoT networks includes Wireless Sensor Networks (WSN) [2], where these networks connect a group of smart “things” that can interact with the physical world and transmit data. Together, WSNs and the internet form the backbone of the IoT infrastructure

The integration of Artificial Intelligence (AI) with IoT has opened exciting possibilities in diverse fields like smart homes, cities, healthcare, buildings, social networks, traffic management, the military, environmental monitoring, energy, transportation, agriculture, education, and more [3]. This merging has led to a significant increase in the number of devices connected to wireless networks and the volume of data transmissions, particularly within sensor based IoT networks. These networks enable real-time remote monitoring and control of "things" – everyday objects equipped with sensing capabilities that connect to the internet, forming a connected ecosystem which is capable of representing the required vision.

1.1 Motivation and Objectives

This section highlights the challenges of routing in complex and dynamic IoT networks encouraging the necessity of the improvement of the applied routing protocol. By introducing Reinforcement Learning (RL) as a promising approach for developing adaptive and intelligent routing protocols that can address these challenges and empower a truly connected world of IoT devices.

1.1.1 The Growth of Complexity in IoT Networks

As IoT has revolutionized our world, it is promoting a network of interconnected devices generating and transmitting vast amounts of data. This rapid growth presents significant challenges for traditional routing protocols designed for simpler network architectures, as these protocols are based on assumptions and no longer sufficient to suit conditions of modern IoT networks. These conditions and factors include:

- **Decentralization and Automation:** IoT networks often lack central control and unmanageable, requiring individual devices "things" to make autonomous routing decisions. This is further complicated by the dynamic nature and uncertainty of the environment, with devices constantly joining, leaving, or changing location [4].
- **Resource Constraints:** Sensor-based devices in IoT networks typically have limited processing power and battery life. The communication is known to be the most energy consuming in the networking node. Therefore, packet routing is one of the important parts of improvement to reduce power consumption in order to keep the node operational as long as possible. Energy-efficient routing is critical to ensure network longevity and maximize device uptime [5].
- **Dynamic Network Topology:** IoT networks exhibit constant changes in topology due to device mobility, varying signal strengths, and fluctuating node energy levels. Traditional routing protocols struggle to adapt to these rapid shifts, as routing protocols in these networks must adapt to changes in network topology, signal strength, residual node energy, and link quality [6].

These challenges necessitate a shift towards intelligent routing solutions that can learn and adapt in real-time, where RL appears as a powerful tool to achieve this goal.

1.1.2 Reinforcement Learning for Adaptive Routing in IoT

Reinforcement Learning (RL) [7] allows machines to learn through trial-and-error interactions with their environment. In the situation of IoT, an RL agent (a learning algorithm) interacts with the network, observing the current state (e.g., node energy levels, distance to destination) and selecting routing actions. The agent receives rewards based on the effectiveness of its decisions, such as successful packet delivery or minimized energy consumption. Over time, the RL agent learns to map network states to optimal routing strategies, leading to improved network performance.

The main advantages of RL for routing [5] that can be achieved in IoT networks are:

- **Dynamic adaptation:** Unlike traditional protocols, RL agents can continuously learn and adapt their routing decisions based on real-time network conditions, making them ideal for dynamic IoT environments.
- **No prior knowledge required:** RL eliminates the need for pre-programmed routing rules or extensive historical data, which are often unavailable or impractical in large-scale and evolving IoT networks.
- **Energy efficiency:** By optimizing routing paths and minimizing unnecessary data transmissions, RL can significantly improve power efficiency, extending the lifetime of battery-powered devices in IoT networks.
- **Scalability to complex networks:** RL algorithms can effectively handle the growing number and diversity of devices within an IoT network, preparing for scalable and strong routing solutions.

1.2 Contribution of This Work

This work presents the routing protocol modified Energy-Efficient Routing based on Reinforcement Learning (mEER-RL), which is built upon the foundation laid by the Energy-Efficient Routing based on Reinforcement Learning (EER-RL) protocol proposed by Mutomba [8]. EER-RL is a cluster-based, RL-driven protocol designed to achieve energy efficiency in IoT communication. It utilizes RL to balance energy consumption across devices, which mainly helps in extending network lifetime. Our work compares mEER-RL with both EER-RL and LEACH (Low-Energy Adaptive Clustering Hierarchy) [9], the latter is a widely adopted protocol that forms the basis for many clustering approaches in sensor networks.

Clustering offers a well-established strategy for managing energy consumption in sensor networks. It reduces energy consumption and packet overhead by enabling data collection and aggregation through the Cluster Head (CH), significantly reducing the volume of data transmitted to the Base Station (BS). Conventional routing protocols faced limitations in IoT and sensor networks, encouraging the development of LEACH. This protocol utilizes a randomized rotation mechanism for selecting CHs, facilitating a more even distribution of energy load across the network. Deeper exploration of protocols details is in the next chapter.

The proposed protocol mEER-RL is a modified version of EER-RL specifically designed to address node mobility and enhance packet delivery ratio while improving the RL performance with more energy-efficient routing. By incorporating these crucial factors into the protocol's decision-making process, we aim to verify mEER-RL's ability to adapt to a critical random factor within the network which is highly impacting routing efficiency. This thesis aims to comprehensively evaluate and verify that

mEER-RL effectively routes traffic for both fixed and mobile nodes while simultaneously extending network lifetime and ensuring reliable data transmission.

The key contributions and improvements that mEER-RL presents:

- **Enhanced reliability:** addressing node mobility, a crucial factor often neglected in existing protocols. This consideration leads to improved packet delivery ratios and overall network reliability.
- **Adaptability:** By incorporating node mobility into the RL agent's reward function, mEER-RL can dynamically adapt its routing decisions based on network conditions, to make it suitable for both fixed and mobile IoT deployments.
- **Improved energy efficiency:** Building upon EER-RL's core strengths allows to maintain a focus on energy efficiency, maximizing network lifetime by optimizing routing paths and minimizing data transmission through packet duplication and losses prevention with improved multi-hop technique.
- **Advanced performance:** By carefully comparing mEER-RL with established protocols like LEACH and original EER-RL, it helps highlighting its advantages in terms of network lifetime, energy consumption, and Packet Delivery Ratio (PDR).

1.3 Thesis Structure

The thesis is structured across five chapters. Chapter 2 establishes the foundation of this work. It explores the architecture and components of an IoT system, focusing on routing protocols and the challenges they face in dynamic IoT environments (mobility, energy efficiency, etc.). Additionally, this chapter examines network clustering methods and explores RL and its applications in routing protocols.

Chapter 3 investigates the existing related cluster-based protocols, namely LEACH and EER-RL, through a comparative performance evaluation using simulations. The chapter starts by discussing the performance evaluation metrics employed throughout the thesis and then details the operation of both protocols.

Chapter 4 exposes the core of this thesis, the mEER-RL protocol. It precisely details the modifications and additions made to enhance the performance of EER-RL and make it applicable for both fixed and mobile IoT networks. These modifications involve aspects like CH election methods, automatic CH distance thresholds calculation, node parameter updates, and multi-hop communication strategies for packet duplication prevention. It also explores how mEER-RL incorporates dynamic learning and a modified Q-function for balancing energy and reliability by the consideration of packet delivery. This chapter also evaluates the performance of mEER-RL through simulations, comparing it against existing protocols (LEACH and EER-RL). It analyzes the effectiveness of the various modifications introduced and presents a comprehensive performance comparison.

Finally, Chapter 5 serves as the conclusion. It summarizes the key findings of the thesis, summarizing the contributions of the mEER-RL protocol and its potential benefits for routing in IoT networks. Additionally, this chapter identifies potential areas for future research and improvement points of the mEER-RL protocol.

1.4 Chapter Summary

This chapter has established the base for the thesis by outlining the motivation and objectives. It presents the reasons behind necessitating the shift towards intelligent routing solutions using RL which is offering dynamic adaptation and eliminating the need for pre-programmed rules, to

improve energy efficiency and scale effectively. The chapter has detailed the contributions of this work. The proposed modified Energy-Efficient Routing based on Reinforcement Learning (mEER-RL) protocol, a cluster-based, RL-driven protocol focused on energy efficiency and transmission reliability. It specifically addresses node mobility, a crucial factor often neglected in existing protocols. The chapter is concluded by outlining the thesis structure.

Chapter 2: Background and Literature Review

2.1 Architecture and Components of IoT System

2.1.1 IoT Network Layer and Routing Protocols classification

2.1.2 IoT Routing Challenges

2.1.3 Mobility Challenges

2.1.4 Data Routing Metrics

2.2 Clustering Methods in IoT Networks

2.2.1 Network Clustering Advantages

2.2.1 Network Clustering Approaches

2.3 Reinforcement Learning Overview

2.3.1 Reinforcement Learning Applications in IoT: A Review

2.3.2 Reinforcement Learning Technique

2.4 Cluster Based Routing Protocols for IoT

2.4.1 Classical Approaches

2.4.2 Optimization Approaches

2.4.3 Machine Learning Approaches

2.5 Chapter Summary

Internet of Thing (IoT) [1] is a network of interconnected physical objects that collects data from the environment and exchange them with other devices or the cloud and communicated through the global internet in a secured real-time manner. It is a key technology for ubiquitous computing, to have all physical objects be embedded with computing and connected anywhere. The data is collected from the environment using embedded sensors in smart devices, WSNs, or smart network enabled objects like Radio Frequency Identification (RFID) tags. This data is then stored and processed to extract insights and take actions accordingly.

IoT devices or “things” are connected over wireless methods and uniquely identified with an Internet Protocol (IP) address. They are used to remotely monitor the environment, track movements, make predictions based on data history, and interact with the smart objects to take control decisions towards actuators [2].

2.1 Architecture and Components of IoT System

Due to the different IoT applications and design requirements, and the wide range of “things”, there is no single architecture for IoT as a universal agreed reference. The most common architectures used in research are described as following:

- IoT basic architecture: Figure 1 illustrates the basic IoT structure that consist of 3 layers [10], perception/sensing layer that collects the data from the environment and respond to the control requests, Network and management layer for communication management and data transmission between intelligent devices (things), network devices and servers. While the application layer is responsible of providing the users with the necessary software to monitor, store and control services.

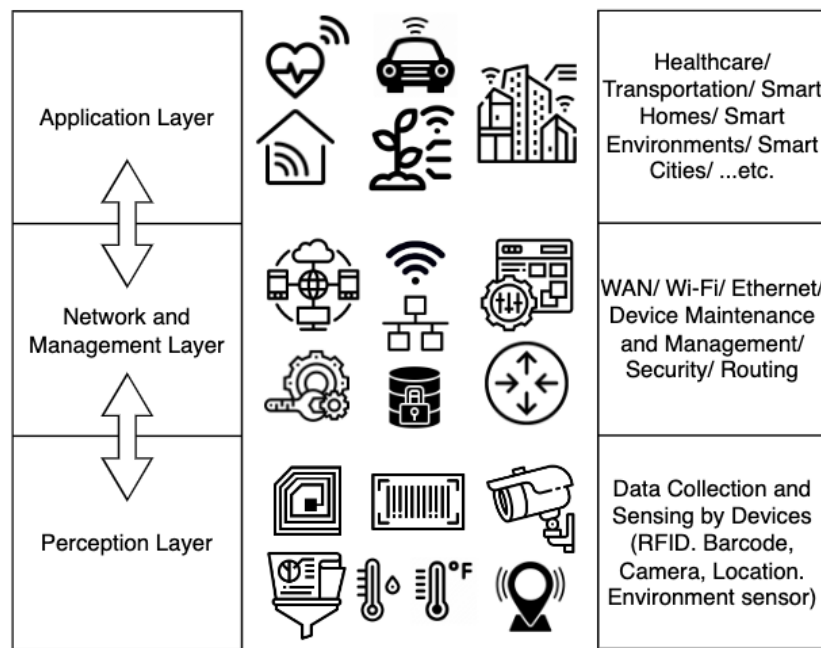


Figure 1 Basic IoT 3-layer structure

- Service-oriented Architecture (SoA): Other IoT structures includes more layers [11] [12], SoA is extending the basic 3-layer architecture with one more layer between application and network layer called service layer. This layer is responsible of supporting application layer by discovering the service requests and interact with the connected objects without the need of location knowledge. It also provides service management for a reliable and available service. Service interface is also part of this layer, which is used to define and support interactions between the services for seamless combination.
- 5-layer Middleware architecture: In this architecture, there are 5 layers which are more sufficient for IoT in research and supports business models. The layers are perception, transport, processing, application and business layer. Both application and perception layers perform the same functionality as the basic 3-layer architecture. Transport layer is responsible of communication and transferring the sensed or identified data to the processing layer through wireless technologies such as Bluetooth, NFC, RFID and WIFI. While the processing layer, which is also referred

as the middleware layer, has critical functionalities to make IoT works efficiently. This layer is performing data aggregation from different devices and filtration of unnecessary data of the received messages from the devices. It also performs information discovery, as well as providing access control among the devices for a secured and controlled access. It uses different technologies including databases, cloud computing and big data processing models. The business layer in this architecture is used for managing the entire IoT system, including application, business models and user privacy. Table 1 summarizes the layers of the most common IoT architectures.

Table 1 The layers of the most common IoT architectures

3-layer	• Application • Network • Perception
5-layer	• Business • Application • Middleware • Transport • Perception
SoA	• Application • Service • Network • Perception

- Other architectures: There are other various architectures with different layers like Cloud-based architecture which uses the cloud to store, process, present and manage IoT data at any time. Edge computing-based architecture is proposed to enable computing services for performance enhancement by minimizing latency, saving bandwidth and providing

security. Another architecture is IoT-Fog, for intelligent network monitoring and control of grids.

The network layer in all IoT architectures is responsible for connecting all of the different devices and components of the IoT system. It is achieved by handling all of the communication between the devices, from the sensors and actuators to the coordinators, gateways, and servers. While there are many different communication protocols that can be used in IoT systems, it is important to choose protocols that are easy to be implemented and used. This is because the communication between the devices needs to be smooth and reliable in order for the IoT application to succeed and work properly.

For the sensing/ perception layer components, the IoT is made up of a number of key components and supporting technologies, all of which are essential for its operation. The data is read or sensed from surrounding using physical identifiers like RFID [13], Quick Response (QR) code [14], barcodes, Near-Field Communication (NFC) and other intelligent devices embedded with sensors like smartphones, smart wearable devices, smart vehicles or a group of sensors working together forming a WSN [2]. WSNs consists of a large number of different types of small sensor nodes that are connected together and deployed to sample, process and deliver the information to the external system which is the Base Station (BS) – Sink through wireless connection [15]. BS works as the wired backbone and network gateway; it has powerful storage and processing abilities without any power constraints [16], see Figure 2. Sensor nodes are known to be a low cost, low-data rate, low energy consumption and short-range link. Each of them is equipped with a transducer, microcontroller, radio transceiver and a power supply [2].

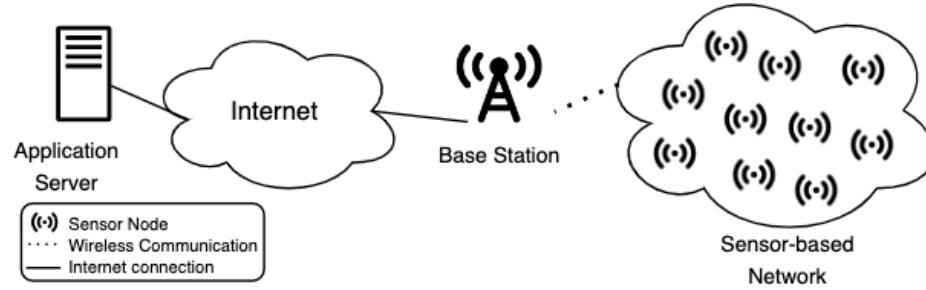


Figure 2 Basic sensor network connected to internet forming IoT

WSN is used as a solution for tracking and monitoring the physical objects or environment conditions, this ability allows for different application of remote management and tracking like monitoring temperature-sensitive products or tracking sensitive industrial tools for enabling preventive maintenance [17]. Integrating identifiers and WSNs with the internet creates an IoT infrastructure, enabling smart devices, not just computers, to connect to the internet and become terminal devices. This integration will need a communication technology to transfer data to the system or required server through a network coordinator. A device or “thing” will communicate with the coordinator through different technologies depending on the type of that identifier, those technologies include Bluetooth, WIFI, Zigbee, [18], Z-wave, LTE or 6LoWPAN [19]. The network coordinator works with or as a network gateway to transmit the data through the internet towards the server or cloud, which is accessible by the user through the application for data analysis and remote control.

2.1.1 IoT Network Layer and Routing Protocols

Classification

IoT is becoming wide in use for small and large fields. The large fields contain a large number of sensing nodes (devices that collect the data), allocated randomly on the area, mostly forming WSNs. Smart Homes,

smart cities, smart grids, agriculture and healthcare IoT normally will have various nodes to sense different type of information like temperature, humidity, location, soil moisture and may take automatic response actions like switching on/off a device, rising temperature or enable a warning message. All IoT architectures share the need of the transmission and network layer. The complexity of network management increases due to the dynamic nature, mobility, power constraints and the increasing number of such network nodes. The problem of routing is defined as path selection from the source to send packets to the destination while meeting the Quality of Service (QoS) requirements and network resources optimization. Current traditional routing techniques are based on assumptions that are no longer sufficient to suit IoT network conditions [5]. Therefore, nodes must have the ability of taking routing decisions for improved performance.

One of the biggest challenges in sensor based IoT is that power units of the sensor devices are small and usually hard to be replaced, although the communication is the most energy consuming in the node. Therefore, packet routing is one of the important parts of improvement to reduce power consumption in order to keep the node operational as long as possible. Routing optimization also aims to reduce complexity, interference, delays, congestion and packet drops, while having the ability to self-configure and organization. Packet routing should have the ability to adapt to the topology and other factors changes (signal strength, residual node energy, link quality, etc.).

Routing improvement for IoT is mostly associate with WSNs as it plays an integral part of this technology that provides faster connectivity and more bandwidth than the other short-range technologies such as Bluetooth and Zigbee. WSNs also contains a larger number of sensor nodes

in a wireless field allocated randomly with different specifications, which raises the challenge in packets routing.

Routing protocols for IoT are most commonly classified based on the following [20][3][21][22][23], summarized in Figure 3:

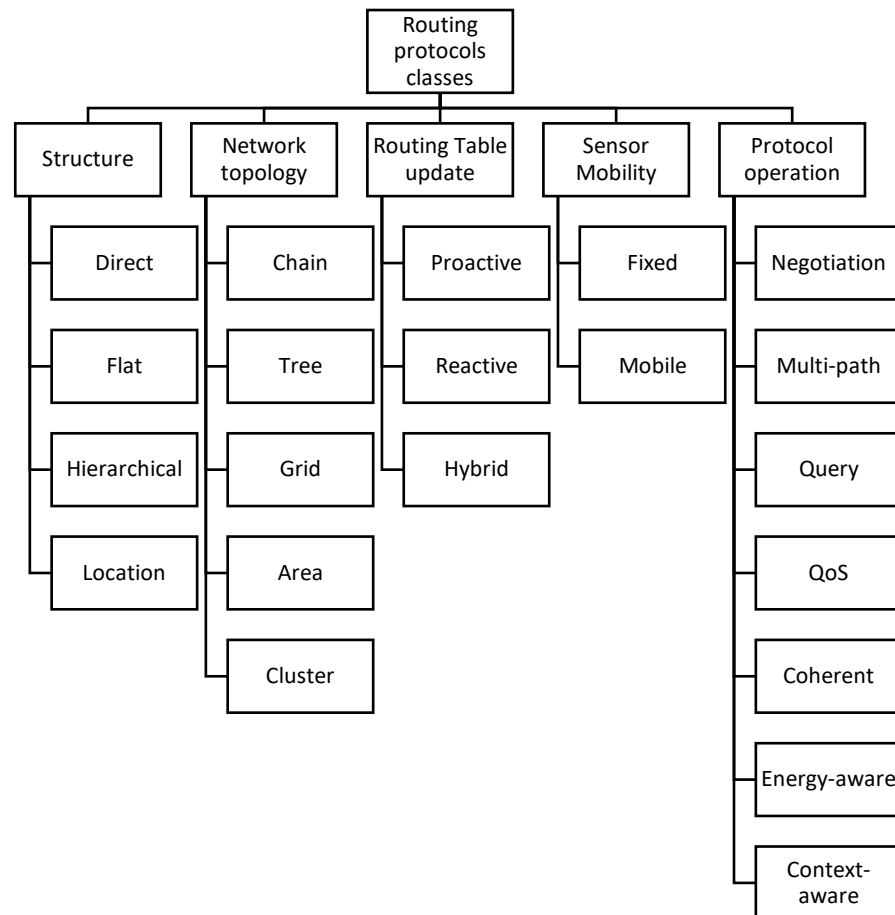


Figure 3 IoT Routing Protocols Classification

a. Structure

Routing protocols in can be categorized based on network structure into direct, flat, hierarchical and location-based routing. Direct routing is the simplest type of communicating with the destination, but it is not feasible as it leads to less scalability and energy inefficiency, as well as having more collisions especially in large networks. In flat routing, all nodes perform similar functions of sensing and routing. Communication is based on multi-hop, which is more scalable than direct, but still leading to

energy efficiency issues as the nodes closer to the sink will have all packets communicated through them. While in the hierarchical routing, the network is subdivided into groups or clusters to achieve better scalability and energy efficiency by having a head node for each group that is responsible of data routing between them. Lastly, location-based routing utilizes node location information to determine the best route, which requires to be accurate.

b. Network topology

The routing protocols are also categorized according to different topologies, chain, tree, grid, area and cluster based. In the chain-based routing, nodes are forming chains and each of them forwards the data packet to the next node in the chain, where each chain has a head that aggregates the data from all members into one message before sending it to the BS, this aggregation approach is used to eliminate packet redundancy. Tree-based routing is where the nodes are organized into a tree structure and forwarding the data to the parent node which aggregates the data and forward it to its own parent, each node will forward to its parent until reaching the root node. The grid-based routing divides the network into grids, where each node forwards the data to the next node in the grid without requiring a routing table, but only knowledge about the location of the next node. Area-based routing is having the nodes designated to specific areas. While in cluster-based routing, the nodes are forming clusters with a Cluster Head (CH) elected for each, the CH is responsible for routing data packets between the Cluster Members (CM) and the BS.

c. Routing table

In this type of classification, routing protocols are classified as proactive, reactive or hybrid based on updating the routing table. A

protocol is proactive when each node periodically maintains its own routing table and update its neighbors about the topology. In reactive, the node sends a query of the route only once a packet needs to be transmitted and no route available in the table. A hybrid protocol that merges the specifications of both proactively maintaining a base routing table, and reactively query for unavailable routes once a transmission is required.

d. Sensors mobility

Related to the movement of the sensor nodes, all nodes are fixed and stationary in static routing protocols, and the network topology does not change. While in mobile routing protocols, sensor nodes are moving and causing the network topology to be dynamic which requires more complexity for routing management.

e. Protocol operation

Based on the protocol operation, the routing protocols are classified into five types. Negotiation-based routing between sensor nodes to reduce redundant information before transmitting data. Query-based routing is a protocol type that relies on destination node queries to find the source node. Multipath-based routing improves the network performance by constructing multiple routes from source node to destination node. Coherence routing protocols uses coherence communication technique to minimize energy consumption during data processing. QoS-based routing concentrates on guaranteeing the QoS parameters of the data, including reliability, delay and bandwidth. Energy-aware routing protocols are used as the sensor nodes in WSNs have limited battery power and recharging or replacing batteries is not always possible. Therefore, energy-aware routing protocols are used to select the nodes with the higher energy resource for an extended network lifetime. Lastly, in context-aware protocols, the environment is continuously monitored by the system and routing

decisions are made without user input, this is done as the system collects information about the environment and uses the contexts such as memory, processing power, location, and mobility speed of each node.

2.1.2 IoT Routing Challenges

IoT routing protocols face a number of challenges and issues during the design and implementation, which are as follows [22][24][25][26][27], and summarized in Figure 4:

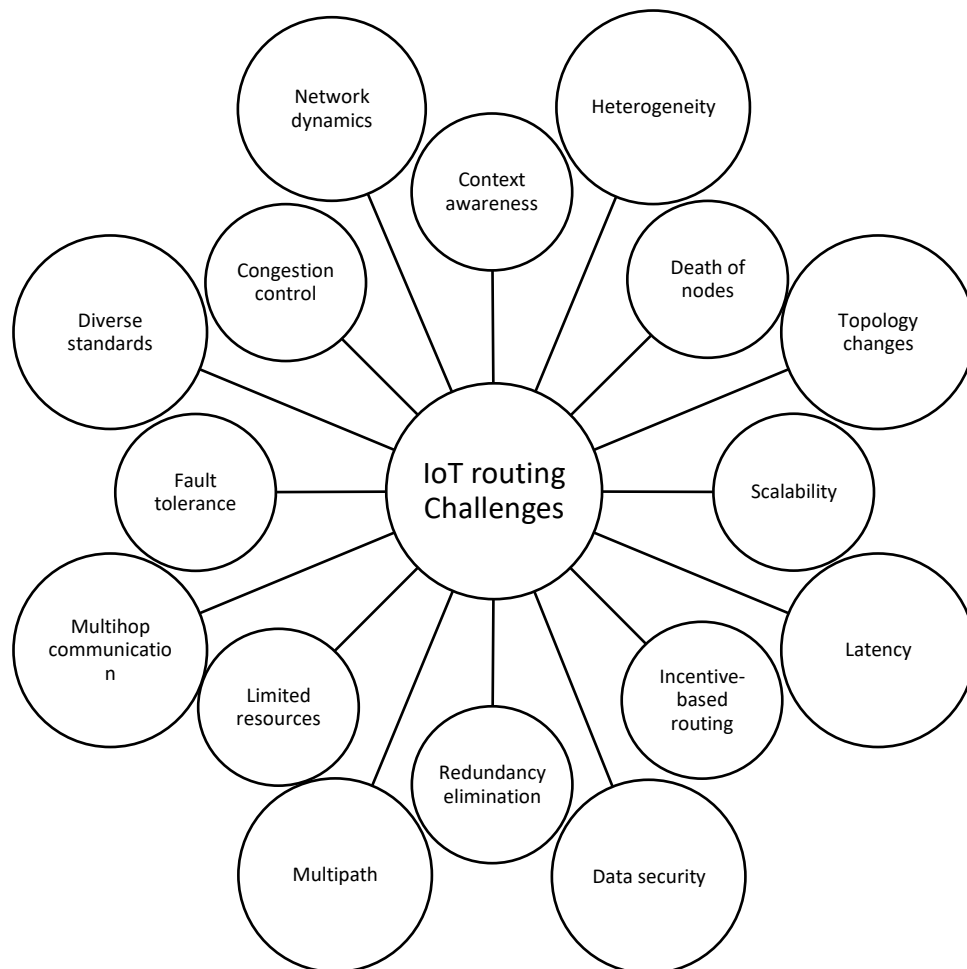


Figure 4 IoT routing challenges

- a. Context-awareness: IoT devices should be able to collect and understand the context of the environment to get the required knowledge for making fast and efficient routing decisions. The context

awareness factors of a node can be residual energy, memory, processing power or link quality.

- b. Heterogeneity: Different types of devices and technologies with different networking standards adds complexity in the routing process. A common routing protocol that includes all types of heterogeneity is needed.
- c. Energy consumption and Death of nodes: Energy constrained nodes in the network dies when it is overused, and usually those nodes batteries are hard to be replaced. This creates holes in the network causing routing difficulties. Nodes energy should be conserved, and exhausting specific parent nodes should be avoided for a longer lifetime, nodes should also be able to reduce control packets and handle malfunctioning due to power failures.
- d. Topology changes: Frequent topology changes can be experienced by the network due to nodes mobility, complete energy reduction of nodes and other environmental factors. This is causing difficulties in maintaining the routing table, which makes reactive and hybrid routing protocols are needed to handle the changes.
- e. Scalability: IoT devices may be stationary or mobile. The nodes are deployed randomly, and mobile nodes could leave or join a network over time, this causes to change the size of the network to be smaller or larger, and the routing protocol should be able to scale and handle all network sizes.
- f. Latency: The generated data in IoT will get expired within some span of time, while there are applications that requires real-time data transmission. Handling latency in routing protocols is very important to maintain quality of services.

- g. Incentive-based routing: Many devices connected to the IoT makes it necessary to cooperate and forward data packets between them for a reliable and successful routing.
- h. Data security: IoT data is often sensitive, and it is being wirelessly transmitted within different networks that possess different owners. Hence, authentication and data security are necessary, to prevent data from unauthorized access.
- i. Redundancy elimination: Large amount of data is being sent for processing through the network, and redundant data can be generated. A routing protocol should be able to eliminate redundancy and reduce bandwidth usage and energy consumption.
- j. Multipath routing: performing load balancing and extending network lifetime are important. Load balancing and multipath routing helps to improve the network reliability, increasing fault tolerance and QoS.
- k. Limited resources: Limited resources of energy supplies, processing power, memory capacities, wireless communication range and bandwidth are significant challenges for running a successful IoT technology. Multihop communication is required for short wireless communication range, and size of forwarded packets have to be limited due to the small storage memory and limited bandwidth. Routing protocols needs to consider the limitation of resources for efficient and lightweight transmission.
- l. Multihop communication: Most IoT devices are low-powered and have short-range transmitters. A relay mechanism must be used to transmit data from source to destination.
- m. Fault tolerance: There is always a threat of the network performance being affected or fail due to different factors such as environmental, deployment mechanism, physical damage, or energy constraints. Such unexpected events require the routing protocols to have some

mechanisms to handle failures and recommend alternative routing paths.

- n. **Diverse standards:** IoT brings various technologies together, such as traditional networks, WSNs, Zigbee and WiFi. These technologies have diverse working principles and use different protocol stack, that should be considered in routing processes.
- o. **Congestion control:** All types of networks may normally get congestions due to the increase of traffic to be transmitted, which causes packet losses, delays and network lifetime reduction. Efficient routing protocols should be able to overcome congestions.
- p. **Network dynamics:** Dynamic WSNs require on-demand routing path generation due to the quickly changing network topology. Network instability may cause the pre-calculated paths to be not used when they are needed, as it could be already changed.

2.1.3 Mobility Challenges

IoT technology encompasses a diverse range of devices, from simple temperature sensors to complex autonomous cars sensors. These devices can be either fixed, such as fire detection and power sensors for control, or mobile, such as smartphones for location detection and vehicles for traffic management.

All of the previously discussed challenges are faced when having both fixed and mobile nodes in the network. Mobility of nodes change their location over time, they add more challenges and complexity to the communication due to the high and frequent topology change that have a significant impact on the network performance [28][29][30]. This challenge requires an optimized design and network routing protocols that is able to handle the additional challenges posed by mobile nodes:

- Energy consumption: The frequent information exchange and routing updates consumes more energy, while the IoT devices have limited battery life.

- Increased latency: The moving nodes may experience increased latency, as they need to constantly update their routing information and reconnect to the network.

- Connectivity and security breakage: Moving nodes in and out of the network coverage area can cause link failures and make the network vulnerable to security attacks.

- Node location: Knowing the mobile sensor nodes position is important for data collection, so routing protocols are required to be designed to help with this task.

- Self-organization nature: The routing protocol should be able to adapt to the self-organizing nature of the nodes.

2.1.4 Data Routing Metrics

For efficient routing in any type of a network, the important measurement metrics should be considered while designing or implementing the routing protocol, because of different factors that affect the performance and quality of communication. According to the authors of [31], the most important metrics of WSNs have been collected and categorized based on the metrics used in the surveyed related publications. The metrics were classified based on the main network requirements as node centric, hop centric, path centric, end-to-end centric or network centric. Each of them is categorized based on the network model layers which are Hardware, Physical layer, Data Link layer, Network layer and Transport layer metrics, see Table 2.

By referring to the classification in Table 2, our concern in this study is to extend the network lifetime which is a network centric. Minimizing energy consumption is an important metric for an extended network lifetime, which is a Node centric within the hardware. In order to minimize the energy through efficient routing within the network layer, we require less transmission requests with lower required energy as possible. Thus, distance and hop-count are important metrics as well as the data collection management which directly affect the number of transmissions within the network. All of these factors of hardware and network layer are working together on the same objective once employed efficiently.

Table 2 Summary of metrics for WSN according to Yuan, Kanhere & Hollick [32]

Layer to centric	Node	Hop	Path	End-to-end	Network
Hardware	Energy consumption, Memory consumption, computation time, time offset, clock skew	-	-	-	Network lifetime
Physical layer	Energy detection	RSSI, LQI, SNR, Packet corruption rate	-	-	-
Data link layer	Radio duty cycle, packet auditing	Single hop throughput, single hop PRR, B factor, single hop delay/jitter	-	-	K factor
Network layer	Neighbor count	Compound routing metrics	Hop count	E2E delay/jitter, E2E throughput, E2E reliability	Data collection (throughput, reliability, delay, energy efficiency)
Transport layer	In/out throughput, information fan-out	-	-	Network efficiency, network fairness	-

Other metrics may be added based on the designed solution and application requirements. In this thesis, our concern is to ensure proper packets routing for fixed and mobile nodes within the network. Therefore, mobility and location metrics are important as well.

2.2 Clustering Methods in IoT Networks

IoT devices are known to be heterogeneous and dynamic, they are also usually energy constrained. Hence, energy efficiency is one of the most important objectives of IoT network optimization. It was highlighted in the previous section that data transmission and communication is the most energy consuming part of the node. In order to reduce energy consumption while maintaining the network performance, avoid delays, reduce congestion, and adapt to topology changes, clustering was identified to be the best method to accomplish these goals.

In order to understand the advantages of clustering method for routing, comparing to non-clustering routing protocols would show the necessity. In non-clustering routing protocols, all nodes are communicating with the BS directly and having the same tasks, which is causing congestion and data redundancy [3]. Also, the BS could be very far, and the energy consumed while sending the data directly to it is causing high energy consumption and reducing the network lifetime due to the large energy required to transmit data. In order to increase the network lifetime, diving the nodes into non-overlapping groups i.e. clusters, would solve the issue, see Figure 5.

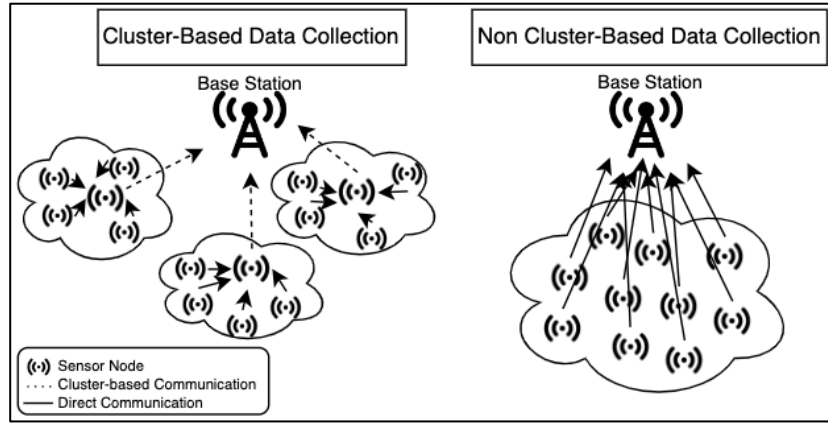


Figure 5 Clustering vs Non-clustering nodes to BS communication in sensor-based networks

Clustering technique is used in different network types, including MANETs to improve routing with energy limitations and movement of nodes [32]. Clustering is also used for VANETs to utilize context and geographic location to minimize routing overhead and delay [33]. Many studies on WSNs using clustering for network lifetime maximization and energy management [34], and IoT for saving energy as well [35].

Clustering in network routing is a topology management technique that identifies groups of similar nodes, forming smaller clusters based on characteristics or relationships. Network functions like data collection and forwarding, are distributed among the group members. There are 3 types of nodes in a cluster, the CH that acts as a local controller of the received data from normal nodes that are called cluster members (CM), CH then forwards the aggregated data to the next cluster hop through the gateway node of that cluster. The gateway node acts as a bridge between two clusters, although this node is not necessary. The clusters forward their data to reach the BS or sink, which serves as a communication channel between the sensor network and Internet gateway.

Network clustering works in two main phases [36], cluster establishment and data forwarding. Cluster establishment includes CH

selection and cluster formation, either grouping the nodes then selecting the CH, or CH is selected first then inviting other nodes to join the cluster. There are different methods followed for CH selection, see Figure 6. Either selecting fixed nodes, random or conscious as CH. Fixed CH selection is based on the resource-rich nodes for the whole network lifetime, which will drain the node energy quickly and cause death. Random CH selection that could be beneficial but accidental unfair randomness can lead to congestion and chronic energy consumption. While the most common method is conscious selection, which is based on the circumstances of the nodes and network, which makes active appropriate node selection based on nodes parameters. The CH role will be rotated (except in fixed CH) and re-clustering will be performed. In the transmission phase, the data is collected and fused by the CH in one message to be communicated to the BS, data fusion reduces the amount of data transmission which minimize overhead and power consumption. A CM can transfer data directly to the CH or through another closer CM, which is called intra-cluster routing, while the communication between CHs and BS is called inter-cluster routing.

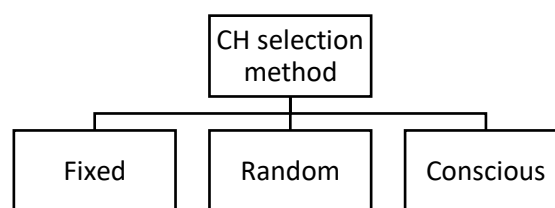


Figure 6 CH selection methods in clustering

2.2.1 Network Clustering Advantages

This technique is the most commonly used routing topology in sensor network based IoT due to the following advantages [37][20][38][39], see Figure 7:

- Eliminating packets redundancy and congestion caused by the large number of devices generating traffic, by data aggregation through cluster heads.
- Reducing routing overhead and saving bandwidth by keeping routing decisions between clusters and BS handled by cluster heads.
- High scalability as the network is divided into smaller clusters for easier traffic management and routing.
- Higher stability periods.
- Better energy efficiency and extended network lifetime, by minimizing energy consumption due to the hierarchical design and consuming lower total transmission power and routing management strategies.
- Providing load balancing, reliability and fault tolerance due to the existence of multiple paths.
- Reducing latency due to the route arrangement of intra and inter-cluster communication.

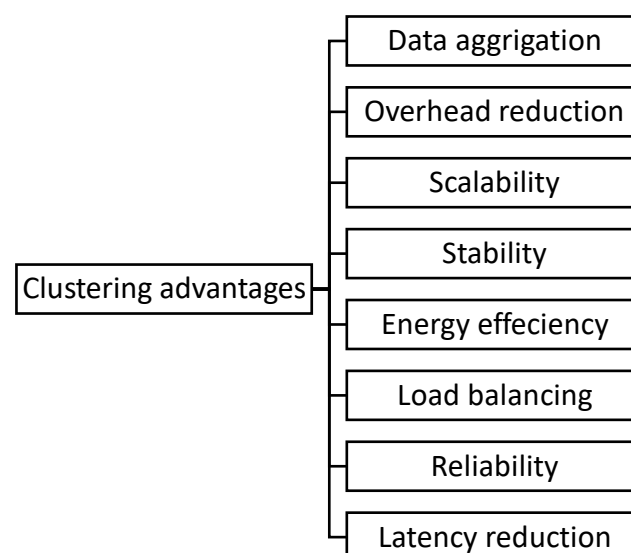


Figure 7 Clustering advantages for IoT routing optimization

2.2.2 Network Clustering Approaches

Many researches are focusing on the improvement of WSN routing, it is a network of smart things, and connecting it with the internet forms an

IoT infrastructure [2]. WSN is actually considered part of IoT perception layer once connected to an internet gateway. There are some researchers who proven that existing clustering algorithms of WSNs are suitable for different IoT application scenarios [39][36]. WSN has been found to be the main method of IoT enablement [34]. Therefore, the remaining of this research includes related work to WSNs and sensor based IoT routing protocols.

There are 3 main strategies used for clustering process [20], which are classical methods, optimization approaches and machine learning techniques.

- Classical clustering methods: In this method, statistical techniques are used to determine the number of clusters and exchanging the role of CHs. The CHs calculates the residual energy of nodes based on a threshold, and this triggers the network to re-cluster. They are simple to implement and mostly suitable for event detecting and periodical data gathering applications. This method ensures that all nodes have an equal chance of becoming CHs. At the same time, this may cause the load of data packets to not be evenly distributed among cluster heads. Another feature of the method is they do not rely on the network size or topology. However, the random selection of CHs in this method causes irregular traffic flow between them. Using two-level clustering reduces energy consumption, but the single hop inter-cluster communication leads to increased energy consumption. They have also some limitations, such as the connectivity between all network nodes may not be guaranteed, and a uniform coverage may not be provided throughout the network. This leads to the formation of energy holes and hot spots because low energy nodes acting as CH.
- Optimization approaches: This method is used to find the solution for NP-hard problems, the problems that traditional methods are unable to

solve them within a specified time. There are two main approaches for optimization, which are heuristic and meta-heuristic. Heuristic approaches are greedy and work on one problem at a time, meaning that finding a solution is fast based on trial and error, but not necessary the best solution. While meta-heuristic works on multiple problems at a time to find a global minimum for an optimal solution. Different meta-heuristic algorithms are used to improve clustering, and the most popular are Evolutionary Algorithms (EAs), approximation algorithms and swarm intelligence.

One of the EA approaches is Genetic Algorithm (GA) that is used to extend network lifetime and improved QoS, it solves a number of problems such as non-continuous, multidimensional, non-differential, non-parametrical problems. Despite their advantages, these algorithms are computationally expensive and may not be able to find the optimal solution in all cases, and unable to be suitable for large areas or real-time applications due to their response time limitations.

For the approximation algorithms, Load-Balanced Clustering Algorithm (LBCA) and Greedy Load-Balanced Clustering Algorithm (GLBCA) are used to evenly distribute the load among CHs. LBCA assumes that all nodes have equal traffic load, it uses a Breadth-First Search (BFS) scheme to find the closest cluster head for each sensor node. This algorithm is not suitable for large WSNs, as it requires a lot of memory. GLBCA takes into account the unequal traffic load of sensor nodes. It uses a bipartite graph to form clusters in such a way that the cardinality of each cluster is maximized. This algorithm is more scalable than LBCA and can be used for large WSNs but may not find the optimal solution.

Lastly, Swarm Intelligence (SI) is a field of study that designs algorithms inspired by the collective behavior of animals and insects. SI algorithms have been shown to be effective in solving a variety of complex

problems. Some of the popular algorithms are Ant Colony Optimization (ACO), Firefly Algorithm (FA), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BFO). FA, ABC and PSO are used for CH selection, ACO is used to determine optimal inter-cluster route. BFO is used to enhance the lifespan of the network. Optimization approaches have several advantages over classical clustering algorithms, such as flexibility, scalability and solving complex problems. But still their limitations should be considered, that they are computationally expensive that needs high memory and processing requirements, they also may involve intense overhead communication to handle the routes, which makes them not suitable to find the best solution for all applications.

- Machine Learning (ML) techniques: ML is a technique that learns from experience or study, it is able to deliver generalized solutions and used in many fields. It increases efficiency and cost-effectiveness of computational processes, it also provides automatic analysis for complex data. The categories of ML are Supervised learning, unsupervised learning and reinforcement learning.

In Supervised Learning (SL), the algorithm is fed with a set of labeled data of inputs and corresponding outputs, the system finds the relationship between them during the learning process in order to be able to create a model that indicates the relations and forecast the output of a new input. There are two categories of SL, which are regression and classification. Regression works on the estimation of precision continuous performance for minimum output errors. Classification is used for predicting discrete output values and have different algorithms: Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN), Deep Learning (DL), Support Vector Machine (SVM) and K-Nearest Neighbor.

For Unsupervised Learning (UL), the algorithm is trained on a set of unlabeled data and no corresponding output values to the inputs. It learns the relationships between the input features by finding patterns in the data, which is suitable for applications like anomaly detection and data aggregation. UL algorithms are divided into two main categories: clustering that groups similar data points together, and dimensionality reduction that reduces the number of input features in a database while preserving as much information as possible.

RL where the algorithm learns from its interactions with the environment. The algorithm is rewarded for taking actions that lead to desired outcomes and penalized for taking actions that lead to undesired outcomes. Over time, the algorithm learns to take actions that maximize its rewards. This method adapts to changing conditions which makes it well-suited for complex and dynamic problems. Q-Learning is known to be the mostly used approach as RL algorithm, more information about RL is discussed in a later section.

ML techniques are used to improve the performance and efficiency of wireless networks, including identifying and resolving problems of broken links to ensure reliable and efficient data transmission. They are also used for optimizing data aggregation which reduces the amount of data that needs to be transmitted for more energy efficiency and network throughput improvement.

According to the above clustering approaches, they are suitable to solve different problems of clustering. However, as the routing in a wireless network is full of dynamics and have energy constraints, RL is found to be suitable for environments involving complex and uncertain behavior of nodes.

2.3 Reinforcement Learning Overview

Reinforcement Learning (RL) [40] is a class of Machine Learning (ML) in the Artificial Intelligence (AI) technologies, performing different methods of advanced functions and algorithms that helps in creating predictions which assists in making decisions for complex problems with dynamic environment behavior including routing in networks [5]. RL is basically a computational approach to learn from its interactions with an environment. The information is mathematically evaluated to be directed for the desired goal through the best experienced action. The learning is based on exploration and exploitation of the actions to be taken by the agent (machine/ computer/ program ...etc.) that leads to the most valuable reward based on trial and error. The action may affect the immediate reward and the accumulated reward.

2.3.1 Reinforcement Learning Applications in IoT: A

Review

RL has a number of potential applications in IoT. IoT devices are often constrained in terms of resources, such as energy and bandwidth. RL can be used to develop efficient algorithms for these devices to perform tasks such as routing data, managing energy, and take control actions.

In smart cities, the amount of data collected is huge, which is hard to be analyzed for decision making and knowledge extraction. Kolomvatsos and Anagnostopoulos [41] have utilized RL technique with data clustering method to use a Query Controller (QC) that learns data separation and assigning queries to the Query Processors (QP) that is responsible of executing queries, which helps in reducing total response time. The method learns the best matches of inputs and outputs for each query based on historical data and a set of query families, it also takes into account the load

of each QP when making decisions. Another approach was proposed in [42] that uses RL to learn the semantics of IoT protocol messages by automatically interacting with the unknown IoT devices and takes control of the device to send the message to the desired goal with minimum interactions.

In [43], the authors have focused on the increased energy consumption with the population and home appliances growth for smart homes. RL has been used to learn the optimal home appliances scheduling, taking into account the electricity cost, user comfort constraints, Distributed Generation (DG) and Electrical Energy Storage (EES) capacity. This strategy is designed to effectively reduce the electricity cost and peak load while satisfying the user's comfort constraints. The results also showed that the proposed strategy is superior to traditional energy management strategies in terms of cost savings and peak load reduction.

In [44], RL was used to develop an offloading technique for distributing tasks across edge, fog, and cloud computing resources in real-time. The goal of the proposed technique is to minimize the task execution time and energy consumption while meeting the application-specific requirements, such as minimized latency and improved reliability. This technique is scalable to large IoT networks and effectively improves the task execution time and energy consumption compared to traditional task offloading approaches.

Due to the high computational complicity and large consumption caused by the direct measurement of large-scale training data set using statistical-based and ML-based methods, an RL-based end-to-end network traffic prediction mechanism in IoT was proposed by the authors in [45]. It is a high accuracy prediction model with lower computational complexity

and a smaller scale training data set, which is enabling efficient resource allocation and congestion control in real-time basis.

The authors in [46] have provided a comprehensive survey of the use of RL for IoT security to identify the different proposed solutions against IoT systems cyberattacks. Intrusion detection systems that learns detecting new attacks, a system that learns to detect anomalous behavior in IoT systems which could be indicative of an attack, a secure routing protocols that can adapt to changes in the network topology and traffic conditions to avoid routing traffic through malicious nodes, a secure access control mechanism that can learn to identify and authenticate legitimate users and devices, those are examples of the discussed security solutions for a reliable technology.

RL was used by the authors in [47] to present a routing protocol for cognitive radio network based IoT environment. This protocol works on minimizing packet collisions with primary users and maximizing the network throughput by adding the channel selection decision capability to the network layer. It successfully achieved better network performance compared to the existing approaches in terms of average data rate, throughput, packet collision, and end-to-end delay. While in [48], the authors have focused on the load-balancing problem in the routing protocols for low-power and lossy networks. RL was used to learn the optimal parent selection policy based on the dynamic network traffic conditions and network topology changes. The routing information are maintained by each node from its neighbors to be used as routing metrics in the Q-function, which is based on the congestion level, link quality and hop distance.

Many researchers have contributed to improve network routing protocols, and clustering was the most method used due to the desired

advantages that it provides. IoT and WSNs shares the same challenges and requirement, and the ability of moving WSN routing protocols to IoT was already proven. Hence, they share the routing protocols that can be applied for efficient network packet management. The next section discusses different proposed routing protocols related to WSN and sensor based IoT.

2.3.2 Reinforcement Learning Technique

The main advantage of RL is that it does not require collected datasets to be available for the learning process, which requires a decision to be taken for each input. In order to model RL, it is important to understand Markov Decision Processes (MDP) which is used to model decision-making problems in uncertain environments and dynamical systems.

Different algorithms are implemented using MDP, such as dynamic programming, value iteration and RL. Generally, MDP is a discrete time stochastic control process [4]. It provides a mathematical framework for modeling optimization problems that are under a decision maker control and described with random outcomes. MDP can be finite or infinite time horizon, it is typically defined as a 4-tuple (S, A, P, R) , where S is a set of states, A is a set of actions, P is a state transition probability matrix (i.e. the probability of transitioning from a state s to state s' by executing the action a), and R is the reward function to obtain the reward r by executing the action a . The mapping of a state to an action is called a policy, which is denoted by π . Finding an optimal policy π^* is the goal of an MDP to maximize the reward function or the expected total reward in case of infinite time horizon. In many cases, the state of the system is not fully observable, and a Partially Observable MDP (POMDP) can be used to model decision-making to find the optimal policy to maximize the expected long-term reward. POMDP is designed by defining 6-tuple (S, A, P, R, Ω, O) where Ω is the set of observations, and O is the observation

probabilities. There is another complex type of MDP called Markov Game, where multiple players (agents) are playing a dynamic game with probabilistic transitions, a set of agents are defined with a global state and action spaces. In addition to the transition probability and reward functions, the rewards are obtained after all agents executed their actions, and each of the agents works on finding their own optimal policy to maximize their own expected long-term reward which is achieved at the end of the learning process.

RL problems are modeled using MDPs for the learning process, based on the reward function algorithms, it is similarly formed as sensation, action and goal. To reach a goal, all RL agents observe the state of the environment in order to get the required information leading to the interaction and deciding the next action that influences the environment back. This leads to receiving the immediate reward and a change in the environment state. The main elements of RL system are Policy, Reward, Value function and optional environment model which are described as:

- The policy is the core of RL, it is the behavior of the agent learning at a given time. It can be a function, a lookup table or involving extensive computation. It is defined as the probability of mapping between the environment state and the action to be taken at that state.

- The reward is a function represents the goal of the problem, calculating the immediate reward. The environment sends the reward to the agent after each time step, either as a good or a bad event. The agent aims to maximize that reward over the long run by altering the policy.

- The value function specifies how much good to be in a state for the long run, calculating the long-term reward (end of episode). It finds the total amount of expected reward starting from that state to be accumulated over the future, as a prediction.

- The model (in some systems) is what simulates the behavior of the environment, for using a model-based method.

To understand the process more basically, after taking an action over the environment, the received reward can be positive for a desired state or negative otherwise, then the agent is aiming to find the optimal policy for reward maximization, by repeatedly using the gained immediate reward and the new environment state to learn the effect and adjust the policy based on it, until approaching the optimal policy of the agent, Figure 8 simplifies this process of the RL state-action-reward model. In the reward function, it is important to note that a short-term (immediate) reward is what gained immediately after each action execution which indicates a good or bad action at that state. While a long-term reward is what accumulated and gained at the end of the learning process indicating how good or bad learned policy. Focusing on the appropriate reward function during the process depends on the design requirements.

There are two approaches to solve RL problems. The first approach is model based, that is improving the policy after learning the environment and computing the transition probability matrix. It makes the learning process fast, because of its ability to reuse information stored in their internal models. But the cost is a greater size storage requirement, and the dependence on an accurate initial environment model. The second RL approach is model free which is mostly used for routing problems, it does not require a prior knowledge of the environment model to improve the policy, nor a transition probability matrix.

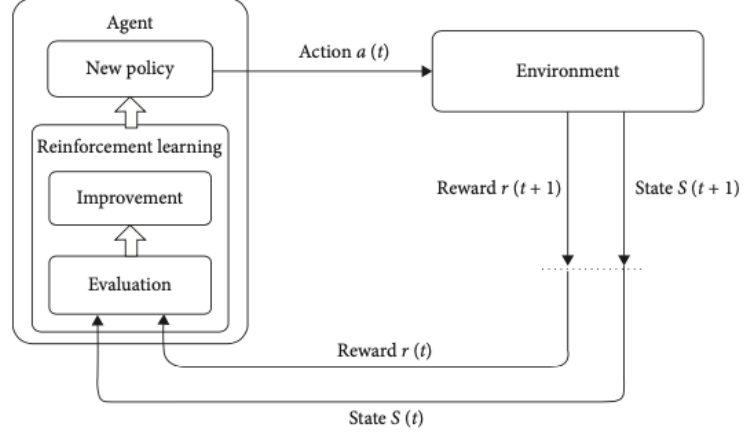


Figure 8 Reinforcement Learning state-action-reward model [8]

The state, action reward modeling for RL is done by applying MDP. In time t , the probability π of taking action to transition from state ‘ s ’ to state s' is denoted by equation (2.1). Each taken action reflects a change in the state and generates a new reward for the agent, this is done through one “step” of an episode. In the end of each step, the agent collects the new observations to be used in the next step and learn. A reward r_t is received by taking the action a_t in state ‘ s ’, and the cumulative reward R is logged after reaching the final result, reaching a previously set bad condition or reaching the final step in an episode. After each episode, the agent learned a part of the policy and it can choose the same or different action at time t based on the previous episode (learning session). The agent is aiming to maximize the reward in each episode, in order to get the best policy that gets the highest global (long-term) reward G_t of taking actions in an environment.

$$\begin{aligned}
 \mathcal{P}(s, a, s') &= \Pr(s'|s, a) \\
 &= \Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \mid \sum_{s' \in \mathcal{S}} P(s'|s, a) = 1
 \end{aligned} \tag{2.1}$$

To address optimality and definition of G_t , there are three basic models can be used [5]. Finite Horizon, Infinite Horizon and Average Reward. In the Finite Horizon, the agent lifetime is known, and it should

optimize the reward during this period where the number of steps is known (number of hops in routing), calculated by equation (2.2). The Infinite horizon model is most dominant in RL systems and the lifetime is unknown. In this model, the agent is concerned about maximizing the long-term reward. Referring to equation (2.3), γ is called the discount factor and its value is between 0 and 1, once the value approaches 0 then the agent is concerned about maximizing the immediate reward, while approaching 1 means the future reward is most concerned. In the Average Reward model of G_t , the agent works on optimizing the average reward for the long term, expressed in equation (2.4).

$$G_t = \sum_{k=1}^h R_{t+k} \quad (2.2)$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k * R_{t+k+1} \quad (2.3)$$

$$G_t = \lim_{h \rightarrow \infty} \frac{1}{h} \sum_{k=0}^h R_{t+k+1} \quad (2.4)$$

Q-Learning is a well-known model-free method to apply RL, it is used to find the optimal policy by finding two value functions, state value and action value functions. To estimate how good for an agent to be in a state and estimate how good an action to be taken by the agent in a state, respectively. This is for the aim of finding the optimal policy that works on maximizing the long-term reward and minimize the overall cost. The Q-value is found for all state-action pairs to find the value of specifically taking an action in a specific state following a given policy as $Q_{\pi}(s, a) = E[(G_t | s_t = s, a_t = a)]$, where G_t is the global discounted reward at time t . The optimal q-value is then found as the maximum value of that policy of state-action pairs to be included in the Q table. On the other side, the V function finds the value of a state under that policy as $V_{\pi}(s) = E[(G_t | s_t = s)]$. Both value functions are correlated according to equation

(2.5), as the state value in an optimal policy must be equals to the expected return of the best action value, this makes Q-Learning easy to be implemented due to the independency of the action function approximation from the followed policy.

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a), \quad \forall s \in \mathcal{S} \quad (2.5)$$

To find the optimal Q-values of all state-action pairs, it is done while going through the sequence of the learning process (in a learning step of an episode), as the agent takes an action in the current state and receives a reward then moving to the next state which depends on the taken action, this is done using equation (2.6), where α is the learning rate and γ is the discount factor.

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \left[r_t(s, a) + \gamma \max_{a' \in \mathcal{A}} Q_t(s', a') - Q_t(s, a) \right] \quad (2.6)$$

The equation shows that the new Q-value depends on the current one, and the temporal difference between the expected Q-value in future and the current value. The learning rate (α) is used to control the effect and influence of new knowledge to the current Q-value, it can be a constant or set to be dynamically adjusted during the learning process i.e. learning rate value is decreasing while training in order to get less influence of new information (it is initially set high and mostly non-negative random value between 0-1) and encourages early exploration for more stabilized learning as the agent converges towards a good policy. Therefore, the learning rate is used to prevent the non-convergence of Q function, as getting a new different reward for every state-actions pairs attempt will cause its performance to be unsteady and bouncy, although the convergence speed is an open issue. The discount factor (γ) is used for reward concentration, either to let the agent focus on the immediate reward by setting γ closer to 0 or focusing on the total future reward by setting γ closer to 1. The

algorithm of Q-Learning will terminate once a previously set episodes number or all Q-values converge (for all state-action pairs), then it returns the optimal policy which indicates a specific action at each state for the maximum values learned [4].

The convergence improvement by the agent occurs through processes called exploration and exploitation. Exploitation works on the improvement of solution based on local search for solutions already found (limited search space), but it may cause (alone) blockage around a local optimum. Hence, the exploration works on searching in a larger space (global) to find other solutions, this leads to the global optimum by visiting every state and consider new solution leading to optimal policy.

A tradeoff between both exploration and exploitation should be considered, as it has a huge impact on the convergence speed to the optimality in the learning process. Different strategies are available as a solution for this tradeoff problem, heuristic is commonly used for RL problems related to large solution spaces systems and it reflects a reasonable cost, it includes greedy, ϵ -greedy, interval-based and probability distribution-based strategies. In greedy strategies, the selection of the action with the highest Q-value is greedily done. This strategy alone may never explore actions with initially low Q-values, which in turn may never converge to global optimum. While in ϵ -greedy strategies, the action with the highest Q-value is selected by default. For the goal of exploring alternative actions, the ϵ probability allows for a random action selection from the eligible actions in the current state regardless of its reward. A different type of selection is followed for the interval-based strategy, each action statistics are stored including the number of trials and successes, allowing for the action selection based on the confidence interval on the action success probability. In the last strategy, the selection decision is

based on a chosen probability distribution, Boltzmann probability distribution is an effective strategy that is sometimes used in routing protocols problem scenarios, described as equation (2.7).

$$P(a_t|s_t) = \frac{e^{\beta * Q(s_t, a_t)}}{\sum_{a \in \mathcal{A}} e^{\beta * Q(s_t, a)}} \quad (2.7)$$

There are also extensions of Q-Learning algorithm that provides advanced models, which are SARSA, Q-Learning for Marcov games [4]. SARSA (State, Action, Reward, State, Action) is an online algorithm to approach the optimal policy by updating it directly while interacting with the environment after each taken action. It differs from the normal Q-Learning that works offline and obtains the optimal policy only after the convergence of all Q-values. The other algorithm Q-Learning for Marcov games works by setting Q functions related to multiple agents in the same environment, each agent updates its values according to the observation of its own and other agent values.

2.4 Cluster-Based Routing Protocols for IoT

IoT technology continuous to expand, it connects a vast and growing number of devices to the internet. The exponential growth increases the need of efficient and scalable routing mechanisms to effectively manage data communication within the IoT network. Cluster-based routing protocols have gained significant attention in addressing this challenge, offering a promising approach to optimize the network performance, enhance adaptability to network dynamics, and improve energy effeminacy in IoT networks.

This section provides an overview of different cluster-based routing protocols for IoT and WSN. We begin by examining classical and optimization cluster-based approaches. We also explore the emerging

trends in cluster-based routing, where the integration of ML and RL techniques is used to optimize routing decisions and enhance network performance.

2.4.1 Classical Approaches

Most of the clustering routing protocols approaches are based on the popular cluster-based routing protocol Low-Energy Adaptive Clustering Hierarchy (LEACH) [9]. It was initially proposed due to the conventional protocols are not optimal for sensor networks. Hence, we start our examination with this protocol to understand the base of classical clustering applications. A summary of the surveyed classical protocols is presented in Table 3.

LEACH (Low-Energy Adaptive Clustering Hierarchy): This protocol is depending on a mathematical equation to randomize the rotation of CH role periodically among the higher energy nodes, which will allow all nodes to participate in this role rotation. The data is aggregated by the CH from all CMs according to the TDMA schedule created by each CH to the members, then the data fused and transmitted to the BS in a one-hop communication mechanism.

LEACH extensions: There are some extensions of LEACH designed for special improvements, following is a brief summary for some of them.

Centralized LEACH (LEACH-C) [49]: CHs are selected based on their location (using GPS) and residual energy, this information is collected in the beginning of each round and sent to the BS which selects the CH and forms the clusters by running an optimization method to determine the clusters. The node is considered in CH selection if its energy is less than or equals to the average energy, else then not. Then the transmission phase is similar to LEACH.

LEACH-Mobile [50]: it is used to verify that the mobile nodes are able to communicate with the CH while transmitting a request packet, then the response packet from the CH should be received within the allocated time slot of the cluster TDMA schedule. The node will send a new join-request within the next slot if the response message was not received from the CH in the allocated slot, then decides which cluster to join.

Energy-LEACH (E-LEACH) and Multi hop LEACH (M-LEACH) [51]: CH election is initiated randomly, in the next rounds E-LEACH is designed for improving the performance by selecting CHs based on residual energy. While M-LEACH is designed for improved transmission between CH and BS in a Multihop method, the CH collects the data from members and transmits an aggregated packet to nearest CH to BS (relay CH).

Vice LEACH (V-LEACH) [52]: introducing a new node called Vice CH in each cluster, where it presents the next node to be the new CH when the current CH dies. This protocol aims to avoid CH election in each round and minimize energy consumption as well as providing reliable communication by having the Vice CH always ready to take CH role. The CH election method is based on minimum distance, maximum residual energy and minimum energy.

MMOHRA [53]: This protocol assumes having a stationary BS and CHs, while few CM are mobile and select their own CH to communicate with. The CH role election is on demand whenever the network topology changes, it is rotated between different nodes based on weighted sum of five metrics, which are average energy consumption (energy), control overhead (overhead), reaction time (rtime), link quality indicator (lqi) and hop count (hops). Data is forwarded from the CM to CH, which will pass the data to the BS. The same calculated value of the weighted sum will

indicate the change in the topology in case of a mobile node became far, which will cause high energy transmission and longer time that will trigger CH re-election.

ECRP (Energy-aware Cluster-based Routing Protocol) [54]: This protocol aims to avoid repetitive clustering by performing the setup phase only once in the first round, similar to LEACH. The intra-cluster communication in the data-forwarding phase is based on TDMA scheduling to avoid collisions, while the inter-cluster communication is based on a cost function to forward data through relay nodes by forming a multi-hop path towards the BS. The cost function considers relay nodes' residual energy, the distance from CH to relay and the distance relay to BS.

Table 3 Classical cluster-based routing protocols in IoT networks

Protocol	Year	Application specific	Approach	CH election method	Routing Method	Routing metrics	Mobility	Mobile Organization	Simulator
LEACH [9]	2000	No	Classical	Randomized probability rotation among higher energy nodes	Static (single hop)	Static (Node to CH, CH to BS)	No	-	MATLAB
LEACH-C [49]	2000	No	Classical	BS Control (based on energy & location)	Static (single hop)	Static (Node to CH, CH to BS)	No	-	Ns
LEACH-Mobile [50]	2006	No	Classical	Randomized probability rotation among higher energy nodes	Static (single hop)	Static (Node to CH, CH to BS)	Yes	TDMA schedule based: new join-request once no response within the allocated slot	Ns-2
E-LEACH M-LEACH [51]	2007	No	Classical	Initially random, then based on residual energy	Multihop	Distance from next hop to BS	No	-	-
V-LEACH [52]	2013	No	Classical	Min distance, max residual	Static (single hop)	Static (Node to	No	-	MATLAB

				energy, min energy		CH, CH to BS)			
MMOHRA [53]	2014	No	Classical	Weighted sum of energy, overhead, rtime, lqi and hops	Static	Static (Node to CH, CH to BS)	Yes	Weighted sum evaluation	Ns-2
ECRP [54]	2020	No	Classical	Randomized probability rotation among higher energy nodes	Cost function	Residual energy, distance to relay, distance between relay & BS	No	-	Castalia

2.4.2 Optimization Approaches

Cuckoo Optimization Algorithm based Routing Protocol (COARP) [55]: An energy-aware clustering-based routing using cuckoo optimization algorithm. Cuckoo algorithm is performed in the BS and used for the CH selection, the nodes with higher energy and less distance to the BS have more chances to be selected, while keeping the distances between CHs are maximized. Then the CM send their data to the CH based on the TDMA schedule, and the CH collects the data from the transmitting nodes to send to the BS in a single hop.

HiTSeC [56]: One of the Swarm Intelligence algorithms is called Bat Optimization Algorithm (BOA). It is used along with a trust level value which determines a trusted or malicious node, this value is increased by the BS once the data evaluated and are correct. BOA is used by the BS to select the CHs among vehicle sensor nodes, the selection is based on the residual energy, trust value and number of neighbors to cover as much nodes as possible. The BS will start new CH elections to all nodes after a period of time or one or more CH energy level becomes below the threshold. Data communication is based on single or Multihop between CHs to reach the BS.

Butterfly Optimization Ant Colony Optimization protocol (BOACO) [57]: This protocol employs a hybrid approach that combines the Butterfly Optimization algorithm (BOA) and the Ant Colony Optimization (ACO). BOA is inspired by the behavior of the butterflies' search; it is used to select optimal sensor nodes as CHs during the clustering process. The searching of CHs in the algorithm is based on the residual energy of each node, distance to neighbor, distance to BS and node degree. The position of each butterfly within the search space corresponds to the coordinates of a sensor node. ACO is inspired by the behavior of ants to discover paths and food. It is used to overcome the limitation of uncertain convergence time with residual energy, distance to BS and node degree. The algorithm is used to determine the shortest routes for data transmission between cluster heads and the base station by placing an ant in each CH to generate a Forward Ant packet, the packet is randomly transmitted to the next CH until reaching the BS. Each packet generates the local database that includes each visited node's information including distance to BS, node degree and residual energy. The residual energy calculation is based on the number of packets transmitted through the network. Based on the collected information, the "pheromone" value intensity updates the CH routing table.

(GAPSO-H) [58]: It is a Genetic Algorithm and Particle Swarm Optimization based Hybrid Algorithm. This protocol works in the environments where the sensor nodes are fixed, and the BS is moving. GA is used for the CH selection, with the consideration of the node residual energy, distance between node and BS, node degree (number of neighbors surrounded by node), node average energy and energy consumption rate. Then PSO algorithm is used to optimize the BS mobility and obtain the next move by targeting the lowest energy CH region, aiming to minimize their energy consumption before they die.

MADCR [59]: In the application of transportation, improving the communication network lifetime will affect to reduce the vehicles latency by having a reliable network to share the road information with the Roadside Unit (RSU). The clusters are formed based on the Euclidean distance calculation, once a vehicle node moves far from the cluster, then re-clustering process will be initiated. Then by using the clustering mechanism, CH is selected using Mayfly Optimization Algorithm (MOA). It is based on the calculation of the distance between the moving vehicles, to find the best position similar to the behavior of mayflies' dance. Finally, the CH aggregates the data from the CMs to forward it to the RSU.

SOMACA [60]: This protocol depends on the vehicle's nodes similarities, which are the speed, position, node degree and others. It utilizes the Euclidean distance as a main parameter during the cluster formation step. Sparrow Search Algorithm (SSA) is used for CH election, which is based on randomness in the search process. The algorithm maintains a population of sparrows, representing potential CH candidates, and iteratively updates their positions based on their fitness and interactions with other sparrows. The fitness of each sparrow is evaluated using a function that considers factors such as residual energy, node degree, distance to neighbors, and distance to the base station. The SSA terminates when a specified criterion is met, and the best sparrow at the end represents the optimal CH selection. The transmission phase is inspired from a non-clustering algorithm TSRP [61], which is proposed to defend against verities of attacks by using a trust value of the neighbors and exclude malicious nodes from communication with the BS. The route selection is based on trust value, residual energy and hop count. Then a route upkeep phase is used to ensure the established routes remain stable and reliable

throughout the operation by monitoring the links for failures or performance degradation, which will start route adaptation process.

A summary of the surveyed optimal approaches for routing protocols is presented in Table 4.

Table 4 Optimization approaches for cluster-based routing protocols in IoT networks

Protocol	Year	Application specific	Approach	CH election method	Routing Method	Routing metrics	Mobility	Mobile Organization	Simulator
COARP [55]	2017	No	Optimization	BS control (cuckoo algorithm: more energy & less distance to BS)	Static (single hop)	Static (Node to CH, CH to BS)	No	-	MATLAB
HiTSeC [56]	2018	Yes (Transportation)	Optimization	BS control (BOA)	Single/ Multihop	Distance to BS	Yes	Re-election after time interval	MATLAB
BOACO [57]	2020	No	Optimization	BOA	ACO	Distance to BS, node degree, residual energy	No	-	MATLAB
GAPSO-H [58]	2021	No	Optimization	GA	PSO	Static (Node to CH, CH to BS)	No, Mobile Sink	-	MATLAB
MADCR [59]	2021	Yes (Transportation)	Optimization	MOA	Static (single hop)	Static (Node to CH, CH to BS)	Yes	Distance calculation using Euclidean distance	MATLAB
SOMACA [60]	2023	Yes (Transportation)	Optimization	SSA	Single/ Multihop	Trust value, residual energy and hop count	Yes	Upkeep phase: route re-formation	MATLAB

2.4.3 Machine Learning Approaches

FTIEE [62]: A different clustering method was designed for this protocol, where the whole network is divided into a constant number of fixed size squares, while keeping smaller size for the clusters closer to the sink, then using Q-learning for selecting the best CH based on the shortest distance to BS without the need of election. Also, the same method Q-learning is used for the transmission based on residual energy and distance, the output will select the next hop either as the BS directly or CH. This protocol is designed to minimize energy consumption by reducing overhead of CH reselection and being fault tolerance.

QL-Cluster [63]: QL-Cluster protocol was specifically proposed for mobile Wireless Body Sensor Networks (WBSN). It is based on Q-Learning to find the best route between the nodes and remote healthcare station in square clusters splitting mechanism within a specified area. This protocol aims to maximize reliability and reduce network overhead. The cluster formation and CH selection is based on a threshold value of the distance between nodes and CH, and between CHs. The CH is often selected as the node that starts communication within the cluster or the node that is centrally positioned within the cluster. When the residual energy of the current CH falls below a certain threshold, the CH role will be transferred to the node with the highest residual energy in the cluster. Q-learning in this protocol is used to find the shortest path from source to destination (monitoring station), it is based on the distance between the destination node and the next feasible node, which is the next hop that gives the best reward of the path that is closer to destination. A variable lock “L” is used as an indication of a node’s Q-value is constant forever (L=1 is fixed, L=0 is changeable), this value is used to minimize computation

requirements which will reduce the number of Q-value comparisons of the possible routes, and will be directly reflected to the current Q-value.

RLBCA & ODMST [64]: RLBCA focuses on solving the issue of hot spot for CH formation, which is the problem of the nodes closer to the sink having their energy decreases more compared to the other nodes. All the nodes are static while having a mobile sink collecting data from CHs, the CH is selected based on the energy level of the nearest neighbor using RL. In the same solution, ODMST algorithm also uses RL and was proposed to find the shortest distance between a CH and mobile sink and traverse the path for data collection. This algorithm saves the energy of mobile sink by visiting only the interested CHs to send data, which also saves the CH energy when the mobile sink comes closer.

RLBR [65]: RLBR was proposed for lifetime optimization of WSNs. It takes into account the residual energy, link distance and hop count to learn the best path from the node to the sink using MDP based RL. This protocol is suitable for small networks, as it is designed as flat routing. The algorithm can be used inside each cluster or among CHs for hierarchical routing communication.

Q-Learning [66]: Q-Learning based routing approach was proposed to learn the shortest path to the next destination (CH). It is combined with an RL based mobile sink model for dynamic routing with efficient data collection from CHs by moving towards the next CH through the shortest path and energy evaluation. This combination improves the network stability and routing performance for an extended network lifetime.

MDC-K [67]: K-means method is used by the BS for clustering and selecting the CHs based on the minimum distance to k centroids of the algorithm output. For the data transmission, a mobile data collector is used

to move around the CHs and collects all data before sending them to the BS.

EER-RL [8]: This protocol uses Q-learning to select the CHs based on the energy level, transmission range and hop count to the BS. A distance threshold is defined to be between CHs to avoid inefficient allocation, and the CH cannot be located at the edge of the network. The nodes closer to the BS are not joining any cluster as they can directly communicate to the BS to save energy. All nodes calculated the Q-value to select the optimal route, and updating it based on the consumed energy for each transmission.

RLBEEP [68]: The routing procedure is similar to the method used in RLBR. In this protocol, the reward function was improved for better route learning. Computational load reduction and energy loss prevention was proven by using the sleep scheduling technique and data fusion.

ReLeC [69]: For the same shared purpose, which is extending the lifetime of the network, RL was employed in the protocol ReLeC to select the optimal path, while relying on hop count and residual energy from the information inserted in the header of the sender packet. This protocol is an extension to EER-RL. The new contribution in ReLeC is the hop count calculation in the Q-function which is inspired by Shannon-entropy to quantify uncertainty.

PECR [70]: This protocol uses unsupervised ML clustering algorithm called K-means, it is used to create clusters and select CHs in the BS based on the distance from node to BS and node energy. The BS selects CHs and a Main CH (MCH) that collects the data from all CHs to forward them to the BS. Data is communicated in a Multihop basis, from CM to CH to MCH to BS, the path is selected based on the shortest distance. CH and MCH roles are rotated between the nodes after each round.

RPLS [71]: has focused on a specific application, that is WSN over Software Defined Networks (SDN) for forest fires detection. The protocol RPLS is designed to delay re-clustering to save energy and focused on keeping real-time responding by anticipation and post-failure reliability parameters. RL was used for routing, considering energy efficiency parameters and post-failure reliability parameters.

A summary of the surveyed ML based protocols is presented in Table 5.

Table 5 ML approaches for cluster-based routing protocols in IoT networks

Protocol	Year	Application specific	Approach	CH election method	Routing Method	Routing metrics	Mobility	Mobile Organization	Simulator
FTIEE [62]	2015	No	ML	RL (Highest Q-value)	RL	Distance, Residual energy	No	-	Ns-2
QL-Cluster [63]	2017	Yes (WBSN)	ML	Initial communication becomes CH, then higher energy node is CH periodically	RL	Distance, lock value variable	Yes	Distance threshold to the CH	-
RLBCA & ODMST [64]	2018	No	ML	RL Energy evaluation	RL	Mobile sink speed, residual energy	No, Mobile Sink	-	MATLAB
RLBR [65]	2019	No	ML	No CH election in the design	RL (Intra-cluster Routing)	Distance, Residual energy, hop count	No	-	Ns-2
Q-Learning [66]	2021	No	ML	Energy evaluation	RL	Distance (Shortest route as one-hop)	No, Mobile Sink	-	Ns-3
MDC-K [67]	2021	No	ML	BS control (K-means: min distance & max energy)	Static (single hop)	Static (Node to CH, CH to BS)	No, Mobile sink	-	MATLAB
EER-RL [8]	2021	No	ML	RL (Q-value, distance between CHs)	RL	Distance, Residual energy, transmission range	No	-	MATLAB
RLBEEP [68]	2022	No	ML	No CH election in the design	RL	Distance, Residual energy, hop count	No	-	Python

ReLeC [69]	2022	No	ML	RL (Q-value, distance between CHs)	RL	Distance, Residual energy, transmission range	No	-	MATLAB
PECR [70]	2022	No	ML	BS control (K- means: min distance & max energy)	Multihop	Min distance	No	-	MATLAB
RPLS [71]	2023	Yes (Forest fire detection over SDN)	ML	Re-clustering delay	RL	Energy, post- failure parameters (link quality), packet delay	No	-	-

All the protocols mentioned above emphasize the need to optimize energy efficiency in IoT network nodes, regardless of the application type. This is due to the energy constraints of all sensor devices. However, we have found that most cluster-based routing protocols for WSN was not considering mobility as a metric in the design, that is more complex to be managed but much important for the recent and future applications of IoT related to healthcare, vehicular networks and other moving nodes. The surveyed protocols by [36] has also shown that the majority is considering fixed nodes only.

2.5 Chapter Summary

This chapter studied the use of RL to enhance cluster-based routing protocols of IoT networks. We identified the limitations of traditional approaches that rely on predefined rules or optimization algorithms for cluster formation and routing decisions. We found that RL offers a dynamic approach that enables the network to adapt to changing conditions and learn optimal routing strategies. We also concluded that energy efficiency remains a critical objective, as highlighted by the surveyed ML approaches. However, mobility is often neglected despite its growing importance in various IoT applications.

This chapter also highlighted a critical research gap, which is most existing protocols fail to consider mobility as a routing metric. This limitation delays their applicability in dynamic environments with moving nodes. Future research should prioritize addressing this gap by integrating mobility metrics like node velocity or location detection into RL framework. This will allow for the development of more adaptive and strong routing protocols that can effectively handle dynamic IoT networks.

This chapter contributes to the ongoing research on optimizing routing protocols for the evolving IoT networks. By the influence of adaptive capabilities of RL and addressing the challenge of mobility, we can open the way for more efficient and reliable communication in different IoT applications. Building on these insights, Chapter 3 studies deeper by evaluating the performance of two specific protocols: LEACH the base of cluster-based routing protocols, and EER-RL that relies on RL techniques for both CH election and communication phases. Through a comparative analysis, we aim to assess the effectiveness of RL in enhancing routing protocols for energy efficiency in IoT networks.

Chapter 3: Comparative Performance Evaluation of Existing Protocols

3.1 Performance Evaluation Metrics

3.2 LEACH Model

3.2.1 Energy Consumption Model

3.2.2 Advertisement and Cluster Setup Phase

3.2.3 Schedule Creation and Transmission Phase

3.3 EER-RL Model

3.3.1 Network Setup and CH Election Phase

3.3.2 Cluster Formation Phase

3.3.3 Data Transmission Phase

3.4 Protocols Simulation and Evaluation

3.5 Chapter Summary

As stated in the previous chapter, there are a number of proposed protocols to support the routing operation in the IoT communication process. In this chapter, we are discussing and evaluating the most relative protocols to the subject and our work is considered an improvement to the previous proposed protocols which are LEACH and EER-RL. LEACH is the start of cluster-based routing protocols, then EER-RL was proposed to improve it by adding RL to the routing and communication phase for an extended network lifetime and more energy efficiency. However, to evaluate the performance of an IoT routing algorithm, there are common metrics which are used for evaluating quantitatively. It helps in showing the feasibility of the method.

3.1 Performance Evaluation Metrics

According to the reference [3], the below are the commonly used performance evaluation metrics for analyzing cluster-based routing protocols that can be used for IoT sensor-based communication networks:

- Packet Delivery Ratio (PDR): It is the ratio between the received packets by the BS and the originally generated number.
- Energy Consumption (EC): The total consumed energy by all nodes, or the average consumed energy.
- Packet delivery delay: Total End-to-End Delay (EED), or the average EED.
- Network lifetime: The time of First Node Dies (FND), the time of Half of the Nodes Alive (HNA), or the time of the Last Node Dies (LND).
- Routing overhead: the ratio of the total generated control packets to the total packets.

3.2 LEACH Model

It is a hierarchical routing protocol that aims to reduce the amount of data transmitted to the BS by evenly distributing the energy load among all sensor nodes in the network, by including randomized rotation of the high-energy CH such that it rotates among the various sensors. It is also tolerant to node failure and reduces bandwidth requirements, which is especially important in IoT networks with a large number of nodes and limited channel bandwidth. To avoid information overload, LEACH uses data aggregation to combine data from multiple sensors into a single packet before sending it to the BS. It utilizes randomized rotation of CHs for an even energy load distribution, and it uses data fusion mechanism to reduce the amount of transmitted data to the BS. The algorithm is divided into two main phases for cluster setup and data transmission.

3.2.1 Energy Consumption Model

The energy consumption is dominated by the radio dissipation of the transmitter and receiver over a specific distance. The energy consumed by the transmitter E_{Tx} is calculated by equation (3.1), which includes the energy required to run the electrical circuit E_{elec} for each transmitted bit and the energy required to amplify the signal E_{amp} . The energy consumed by the receiver E_{Rx} is calculated by the equation (3.2), which includes the energy required to run the electrical circuit for each transmitted bit.

$$\begin{cases} E_{Tx}(k, d) = E_{elec} \times k + E_{amp} \times k \times d^2 \end{cases} \quad (3.1)$$

$$\begin{cases} E_{Rx}(k) = E_{elec} \times k \end{cases} \quad (3.2)$$

The transmission operation consumes more energy because of the need to amplify the signal which is proportional to the square of the distance to the next hop. The amplifier energy should achieve an acceptable E_b/N_a which is calculated as the energy required to transmit one bit of

information, divided by the power spectral density of the noise added to the amplified signal.

3.2.2 Advertisement and Cluster Set-Up Phase

The decisions are taken by the nodes without the BS being in control, allowing all nodes to become a CH at least once of N/K a round (N: number of nodes, K: desired number of clusters) based on the suggested CH percentage in the network.

To select a CH for each cluster, each node determines a random number between 0 and 1. Then it is compared with a threshold value $T(i)$ of equation (3.3), where r is the current round, P is the desired percentage of CHs in the network, and G represents the set of nodes that are not selected as CH during the last $1/P$ rounds. Hence, the probability of a node in the set G to become a CH will increase each round.

$$T(n) = \begin{cases} \frac{P}{1 - P * \left(r \bmod \frac{1}{P}\right)}, & \text{if } n \in G \\ 0, & \text{otherwise} \end{cases} \quad (3.3)$$

Once the random number is less than $T(i)$, the node becomes a CH, or else it is ordinary node. Then the CH advertises a broadcasted message to the other nodes using CSMA MAC protocol.

The normal node joins the cluster based on the received signal amplitude from the nearby CHs which indicates the minimum required energy to send its data packet, the node chooses the CH randomly in case of signal strength equality. The nodes then inform the CH that it will be a member of this cluster using CSMA MAC protocol, the CH keeps record of CMs.

3.2.3 Schedule Creation and Transmission Phase

The CH creates a TDMA schedule for the members and allocates each CM to their transmission time slot, this allows the CMs to turn off their radio out of their allocated slot to minimize energy dissipation. Each CM with data to be transmitted will deliver it to the CH, which collects the data from all CMs and transmit it in an aggregated packet to the BS in a one-hop mechanism. In the next round, the same process repeats starting from cluster set-up phase until all nodes dies or reaching a predefined time period.

3.3 EER-RL Model

EER-RL is a cluster-based energy-efficient routing protocol built using reinforcement learning, it is designed for IoT and its related applications especially with WSNs. It is designed to learn making better routing decisions to balance the energy between network nodes which will extend network lifetime and improve scalability. All nodes share their local information with their neighbor nodes using feedback mechanism, this information is sent within the packet header and includes device id, residual energy, position coordinates, and hop count to the sink which can reduce end-to-end delay. The neighbor nodes overhear it and extract the encapsulated information to update its routing table and calculate the reward.

A network using EER-RL protocol goes through three phases: Network setup and CH election, Cluster formation, and Data transmission.

3.3.1 Network Set-Up and CH Election Phase

In the beginning of this phase, the BS broadcasts a heartbeat message that includes its position coordinates. The surrounding nodes receives the message and save the BS position, then computes the initial Q-value using

the initial energy level and the hop count, equations (3.4) and (3.5) are used for this computation.

$$Q = \begin{cases} \frac{1}{N_h}, & \text{if } E_{min} = E_{max} \\ p \times \left(\frac{E_r - E_{min}}{E_{max} - E_{min}} \right) + (1 - p) \times \frac{1}{N_h}, & \text{if } E_{min} \neq E_{max} \end{cases} \quad (3.4)$$

$$N_h \cong \frac{D_{link}}{TX_{range}} \quad (3.5)$$

N_h denotes the hop count from the node to the BS, that is calculated using the transmission range TX_{range} and the distance of the coordinates D_{link} . For the initial D_{link} value, it is approximated by finding Euclidean distance between each node and the sink using equation (3.6).

$$D_{i,sink} = \sqrt{(x_i - x_{sink})^2 + (y_i - y_{sink})^2} \quad (3.6)$$

p of equation (3.4) is the probabilistic value defining the impact of residual energy to N_h . In this equation, the higher p value leads to select the neighbor with higher energy as the next hop, rather than the closest one. While a lower p value gives more probability for the closest node as the next hop, rather than the one with higher energy.

In the simulation, it should be assumed that all devices have different energy levels and transmission ranges. Also, a distance threshold should be set between CHs and BS which helps to minimize network overhead and make it easier for far nodes to find a CH and make the link diverge to the Sink. The CH cannot be at the edge of the network as it will cause energy wastage for extending the energy distance when communicating with the BS and other nodes. Then the CH is elected based on the highest Q-value, noting that the distance between CHs should not be less than the

specified threshold, which will prevent the clusters to be formed in one side in case of high energy nodes exist close together.

3.3.2 Cluster Formation Phase

Once the election is done, all CHs sends an invitation message to all other devices within its transmission range. The message includes CH id, initial Q-value and location coordinates. Then the other nodes decide which cluster to join based on the distance to CH, then it sends a membership message including local information to the selected CH. Next, a membership confirmation is sent from the CH and forms the cluster. For the nodes that have the BS in their transmission range, they do not need to join any cluster, but saving energy by communicating directly with the BS. Inter-cluster communication is available for more energy efficient model, as the far CHs from the BS can send packets to other CHs or to a powerful node closer to BS in a multi-hop scheme. Intra-cluster communication is also possible, when a node within a cluster is far from its CH, it sends the packet in a multi-hop scheme through other closer members.

3.3.3 Data Transmission Phase

After the completion of cluster formation, the equations (3.7), (3.8) and (3.9) are used to update D_{link} value, as the Euclidean distance is found between the CM and CH, and between CH and BS. Intra-cluster and inter-cluster next-hops are considered.

$$D_{link} = D_{i,j} + D_{j,sink} \quad (3.7)$$

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.8)$$

$$D_{j,sink} = \sqrt{(x_j - x_{sink})^2 + (y_j - y_{sink})^2} \quad (3.9)$$

Moreover, this is the phase where RL is applied. The main components of the RL model in EER-RL for IoT networks are defined as:

- **State:** the state is the environment information taken from the network, which is useful for the decision making at a given time. Where the state space is the available routes to the sink through neighbor devices. Taking in consideration the residual energy, hop count and signal strength to the neighbors.
- **Agent:** all devices are considered as agents, and multiagent RL is required for the network. The agent takes decisions for the next-hop selection based on the shared information through the state, in order to route the packets to the BS.
- **Action:** reaction of the agent at a given time for a given state. According to the available state information, the action space is a set of available neighboring nodes routing to the BS.
- **Policy:** the agent behavior through a given state at a given time, mapping the state-action pairs. It can be stochastic or deterministic and improves over time.
- **Reward:** the cost of the taken action by the agent at a given state.

Each node learns to take better routing decisions, by using the immediate reward to update the Q-value, in addition to finding the optimal policy that maximizes the long-term reward. Three functions are used for the optimization. The energy consumption model is used to update the residual energy by subtracting the dissipated energy after every packet transmission. Energy consumption is calculated by the same equations (3.1) and (3.2) used for the energy consumption model of LEACH.

Then, the updated residual energy is used along with the hop count to evaluate the reward. The reward in the end is handed to update the Q-value by using it as an argument in the Q-value function. Equation (3.10) shows

the reward function, which is used in equation (3.11) for the Q-value update and route the packet to the highest Q-value neighbor.

$$r_{t+1} = \begin{cases} \frac{1}{N_h}, & \text{if } E_{min} = E_{max} \\ p \times \left(\frac{E_r - E_{min}}{E_{max} - E_{min}} \right) + (1 - p) \times \frac{1}{N_h}, & \text{if } E_{min} \neq E_{max} \\ -100, & \text{if } E_r < 0 \end{cases} \quad (3.10)$$

$$Q_{t+1}(s, a) = (1 - \alpha)Q_t(s, a) + \alpha \left(r_{t+1}(s, a) + \gamma \max_a Q(S', a) \right) \quad (3.11)$$

During the reward calculation, once a node has E_r less than 0 (dead), then a negative reward is assigned and encapsulated in the packet header to avoid network overhead, then all neighbors will update their routing tables.

In equation (3.11), the learning rate α controls the effect of the new knowledge to Q-value, setting it closer to 1 makes the learning faster. Then policy is updated according to the equation (2.3) of RL technique, the policy denotes that the agent chooses the neighbor with the highest Q-value for reward maximization. For the discount factor to focus on the future accumulated reward, it is better to be set closer to 1.

Multi-hop communication is performed for far nodes. Inter-cluster communication allows far CHs from the BS to transmit packet through other closer CHs. Similarly, intra-cluster communication allows far CMs from the CH in a cluster to transmit packets through other closer CMs.

During this phase, once a CH residual energy has reached the energy threshold, re-clustering request will be detected, and the protocol goes back to the CH election phase by evaluating Q-values of the nodes.

3.4 Protocols Simulation and Evaluation

Both LEACH and EER-RL codes are made through MATLAB. To support this thesis, both have been taken to be ran and simulated for results evaluation and study. To begin, both networks' parameters are defined as shown in Table 6.

Table 6 Parameters of LEACH & EER-RL simulation setup

Parameter	Value
Sensing field	100 × 100 m
No. of sensor devices	100
No. of CHs	10% of total nodes
Sink location	50,50
Transmission range	20 m
Data size	4000 bits
E_{elec}	50×10^{-9} joules/bit
E_{amp}	100×10^{-12} joules/bit/m ²
Initial energy	1-2 joules

The simulation starts with randomly scattering 100 nodes in a 2-dimensional 100*100 m field size, with a total number of CHs in the field is 10% of total number of nodes. The BS is positioned at the center of the field, where the coordinates are x=50 and y=50. As to study the energy consumption behavior, fixing some parameters would be preferred, it is assumed that all the nodes have the same transmission capability to transfer data over the range 20 m. Also, they all have data to be sent each round with data size is 4000 bits. However, the initial energy is defined as a random value between 1-2 joules. To calculate the transmission and reception consumed energies during the communication, E_{elec} and E_{amp}

are set to 50×10^{-9} joules/bit and 100×10^{-12} joules/bit/m² respectively, for all nodes.

After the network is defined, the protocol begins with calculating the initial values to start CH election. Then the other nodes start joining clusters for cluster formation. Figure 9 shows the sensing field with the formed clusters by both routing protocols LEACH and EER-RL. The difference between both protocols can be clearly observed on this step. LEACH could form 8 clusters only out of the 10, this is leading us to understand that only 8 nodes succeeded to have a random value less than the threshold. This value is changing in every iteration to be more or less than the desired number of clusters, which means the number of clusters is changing depending on this random value which is also observed during the simulation through multiple runs. EER-RL is more stable for CH election and forming the clusters, by noticing the nodes in the field that 10 clusters were formed based on the maximum 10 initial Q-values of the nodes. This is done while considering the threshold distance between the elected CHs to prevent close nodes from being CHs. Having no threshold between CHs is causing non-balanced clusters divisions and is not serving the energy efficiency purpose as what is found in LEACH. Another difference between the two protocols is the nodes having the BS within their transmission range. In LEACH, they join clusters. While in EER-RL, no need to join any cluster in order to save energy from being wasted through the unnecessary communication with a far CH, as less energy is required to directly transmit the data to the BS.

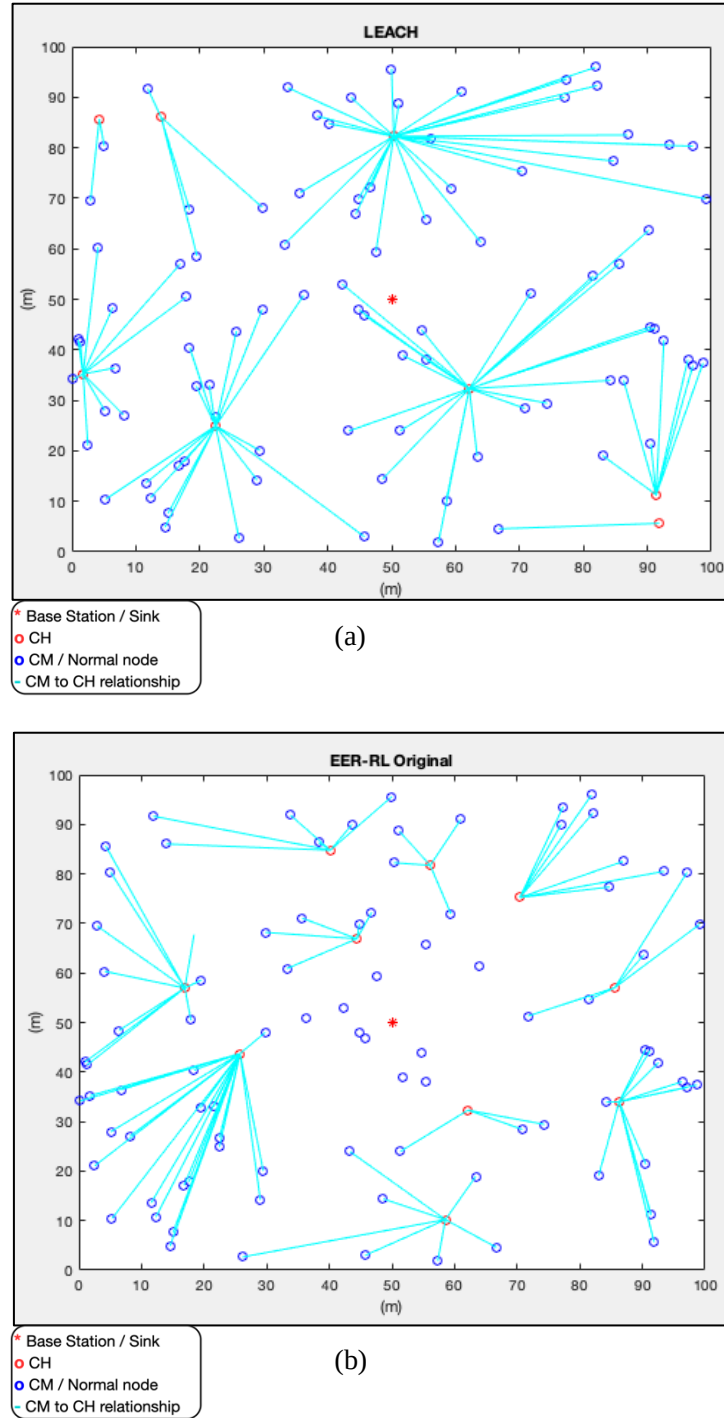


Figure 9 Network setup and cluster formation using both LEACH (a) and EER-RL (b) protocols

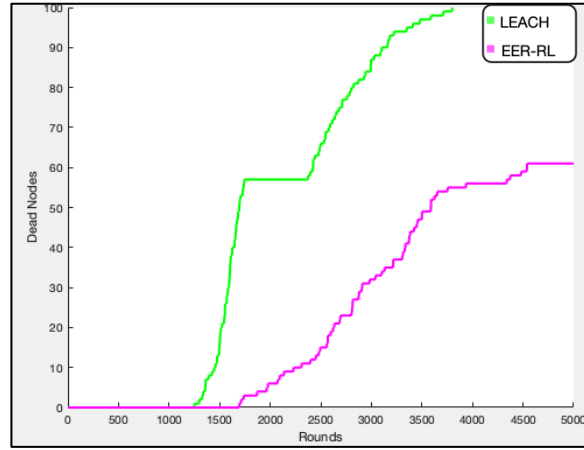
The simulation is set to end after 5000 rounds, or once all nodes are dead ($E_r < 0$). The results of a run are listed in Table 7, where the performance evaluation metrics used are the energy consumption (average energy) and network lifetime (first and last node dies) metrics. Also, as the

simulation is limited to 5000 rounds, another metric is added for the number of alive nodes after reaching the last round.

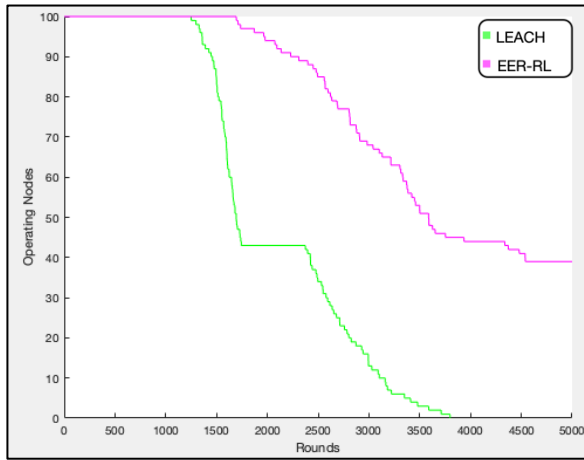
Table 7 Simulation results of LEACH & EER-RL

Evaluation metric	LEACH	EER-RL
First node dies at round	1250	1695
Last node dies at round	3799	5000+
Alive nodes after 5000 rounds	0	39
Average energy (J)	0.0286	0.0261

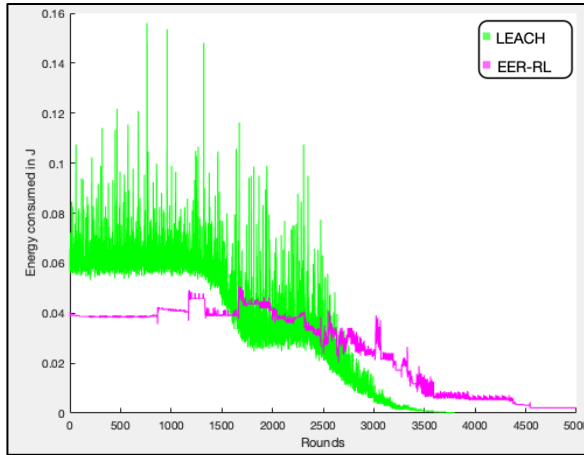
As can be observed from the results, LEACH network had the value of FND at round 1250, while EER-RL had reached a longer stable period by having FND at round 1695. LEACH have failed to reach more than 3799 rounds before the LND in the field, while EER-RL network had 39 nodes still alive after the last round of the simulation has been reached. Figure 10 (a) and (b) shows the total dead and alive nodes per round. From the Average Energy (AE) in the results, we can also conclude that despite the longer life of the network using EER-RL protocol, it has consumed less energy during the communication, resulting in much better energy efficiency than LEACH. Figure 10 (c) shows the energy consumption per round by both protocols, where the behavior in the consumption is much stable using EER-RL protocol, as the RL agent learns the optimal path selection and balancing the requirements of the routing metrics including the number of hops by periodically updating the route selection policy. LEACH is a simple algorithm that reduces control messages overhead and global data communication. But the random CH selection is not offering uniform distribution among them, and the data transmission is based on one-hop communication to the BS, these disadvantages still causing the protocol to be not approaching efficient energy consumption or other features like multipath availability.



(a) Total dead nodes per round



(b) Total operating nodes per round



(c) Total energy consumption per round

Figure 10 Simulation observations of LEACH and EER-RL in 5000 rounds

EER-RL's well-structured phases, readily available code, and compatibility with the popular simulation tool MATLAB, further support its suitability as a foundation for further development which will be discussed in the next section.

EER-RL features and advantages are found as:

- Extending the network lifetime and saving energy by following the clustering structure.
- Providing load balancing, reliability and fault tolerance due to the existence of multiple paths.
- Saving energy by excluding the nodes with the BS in their transmission range from joining clusters.
- Supporting Multihop routing for both inter and intra-cluster communications.
- Applying RL for the routing process which is most suitable for such uncertain and complex scenarios, where the RL agent is capable of offering self-organization benefits.
- Utilizing the same calculated Q-value for both phases CH election and data routing.
- The consideration of the distances between CHs for the best utilization of clustering method and better energy efficiency.

However, despite the advantages that the original EER-RL protocol is providing, it revealed limitations that are hindering its practical application. The protocol contained bugs causing infinite loops when attempting to adjust hyperparameters, and lacked crucial considerations impacting the RL agent's behavior and consequently, the resulting performance. Furthermore, the design of EER-RL assumed a static network with fixed sensor nodes communicating with a fixed BS, restricting its applicability to scenarios of environments involving mobile nodes.

Furthermore, most reviewed protocols utilizing ML techniques apply them solely to either CH election or packet route selection, neglecting the potential benefits in both aspects. Additionally, the majority of these

proposed protocols lack the support for mobile sensor nodes. Also, many of them adopt a mobile sink approach, which presents challenges such as single point of failure and delays due to dependencies on the mobile sink's location. This approach of having mobile sink is also not applicable for mobile nodes scenarios due to the movements causing location and communication blackouts.

Considering these limitations, the decision was made to modify and enhance the existing EER-RL protocol rather than proposing a new one.

3.5 Chapter Summary

This chapter explored the performance of existing cluster-based routing protocols, LEACH and EER-RL, in the context of sensor based IoT and WSNs. We evaluated their performance through simulations and identified strength and limitations. LEACH offers simplicity with low control overhead but suffers from uneven cluster distribution, limited routing options, and lack of fault tolerance. While EER-RL introduces RL for improved CH election and routing, leading to extended network lifetime, Multihop communication, and energy-aware decision making. However, it reveals limitations such as the lack of adaptability to network parameters and being limited to fixed network only. These finding highlight the need for further development in routing protocols for sensor based IoT networks. The next chapter will introduce our proposed protocol, which is built upon EER-RL while aiming for enhanced performance and applicability in the diverse network scenarios.

Chapter 4: mEER-RL: A Modified EER-RL Protocol for Enhanced Fixed and Mobile IoT Networks Performance

4.1 CH Election Method

4.2 Consideration of Distance Threshold Between CHs

4.3 Alive and Dead Nodes Parameters Update

4.4 Re-clustering Phase Rearrangement

4.5 Intra and Inter-cluster Multi-hop Communication

4.6 Dynamic Learning

4.7 Modified Q-function and Reward: Balancing energy and reliability

4.8 Node Mobility Consideration

4.9 Chapter Summary

This chapter discusses the journey of the modifications and enhancements applied to the EER-RL protocol for mobile IoT networks. While the original EER-RL protocol exhibits promising potentials and outperforms other protocols as discussed in Chapters 2 and 3, certain observations are highlighted for improvement during the performance evaluation.

This chapter introduces the mEER-RL protocol, Modified Energy-Efficient Routing based on Reinforcement Learning, specifically designed for energy-efficient data routing in mobile IoT networks.

Building upon the network specifications and RL learning process established in the original EER-RL (discussed in Chapter 3), mEER-RL introduces several enhancements and additions specifically targeted towards improving the protocol's performance within the simulated network environment. These modifications are discussed in detail in the following sections.

4.1 CH Election Method

The original EER-RL algorithm was designed to operate with a predetermined CH selection rate for of 10% of total nodes in the network field. While this approach may be sufficient in certain scenarios, mEER-RL introduces modifications for improved network adaptability. mEER-RL implements two key enhancements to its CH election process:

- **Automated CH number calculation:** The algorithm dynamically calculates the optimal number of CHs based on the initial network configuration. This eliminates the need for manual configuration changes and ensures appropriate cluster size for varying network densities, leading to better resource management.

- Near-Sink (NS) nodes exclusion: To achieve a balanced cluster size across the network, mEER-RL excludes NS nodes from the CH number calculation as they are not included in the clustering process. This ensures more uniform distribution of responsibilities and load balancing of the nodes involved in the clusters.

To investigate the impact of the CH percentage on network performance during setup and communication phases, a modified equation (4.1) is included in the algorithm. This equation accurately calculates the required number of CHs (CH_{tot}) based on three parameters which are:

- Desired CH percentage (CH_{perc}): The user-defined target percentage of nodes elected as CHs.
- Number of NS nodes (NS): The total number of NS nodes that present in the network around the sink, defined by a specific range.
- Total number of nodes (n): The total number of nodes deployed within the network field.

$$CH_{tot} = (n - NS) \left(\frac{CH_{perc}}{100} \right) \quad (4.1)$$

Simple Example 1 and Example 2 shows the result of the above equation for different network sizes, representing its ability dynamically determine the appropriate number of CHs in various scenarios.

Example 1

For $n = 100$, $CH\% = 15\%$, $NS = 12$

Modified equation:

$$\begin{aligned} CH_{tot} &= (n - NS) \left(\frac{CH_{perc}}{100} \right) \\ &= (100 - 12) \left(\frac{15}{100} \right) = 88(0.15) = 13.2 \approx 14 \end{aligned}$$

Example 2

For $n = 65$, $CH\% = 15\%$, $NS = 7$

Modified equation:

$$CH_{tot} = n \left(\frac{CH_{perc}}{100} \right)$$
$$= (65-7) \left(\frac{15}{100} \right) = 65(0.15) = 8.7 \approx 9$$

By using this equation, CH_{tot} is calculated accurately during two phases, as an initial value during the setup phase to establish the initial cluster structure, and before re-clustering during the communication phase to ensure ongoing adaptation to network changes and maintain efficient data routing.

4.2 Consideration of Distance Threshold Between CHs

To ensure efficient cluster organization and prevent excessive concentration of CHs in specific areas, the CH election process incorporates a check on the distance between a potential CH and all already elected CHs. This check should ensure the distance remains below a pre-defined threshold, thereby promoting balanced cluster distribution and mitigating potential network data routing inefficiencies. Unbalanced cluster distribution directly contributes to energy inefficiency within the network. This occurs when the distance is increased between the majority of CMs and their respective CH. As a result, CMs will consume more energy to transmit data to the CH, which directly contradicts the energy-efficient design philosophy of the proposed protocol and reduces network lifetime.

The method of EER-RL is designed to have pre-defined fixed values of the distance threshold which is suitable for the 10% CH number, 20m threshold (transmission range of nodes) during the setup phase, and 15m during re-clustering. These values must be changed when the network parameters are changed, including CH% and total number of nodes.

However, in some situations, once the majority of the CHs are elected, there are no more nodes are suitable to be elected as a CH to fulfil the condition of CH_{tot} number when the distance threshold is satisfied between all elected CHs, and other nodes are already within the same range of the elected CHs and none of them can create another cluster, which might cause infinite loop.

Two additions have been made to the protocol in order to adapt to the dynamicity of the network and the different possibilities of the parameters:

- No more clusters condition: Once all nodes are checked, and no more nodes are required to be elected as a CH, then stop searching and keep the current number of elected CHs as the clusters are efficiently distributed by following the threshold distance. Equation (4.2) represents the conditional CH election method, which depends on 2 essential parameters, Q-value and D_{j,CH_n} , where the latter represents the calculated distance between a node and CH. The method takes the node with the maximum Q-value in the network $CM(\max_{j \in n} Q_j)$, the node will be elected as a CH only if all $D_{j,CH(1:CH_{count}-1)} \geq T$. Where $D_{i,CH(1:CH_{count}-1)}$ is the distance between the current examined node and all available CHs, and T is the predefined distance threshold value. CH_{tot} will be updated to be substituted by the current number of elected nodes CH_{count} , only if no more nodes are found to be elected as CH after being examined and flagged using *node_counter*.

$$\begin{aligned}
& CH_{count} \leq CH_{tot} \\
& \rightarrow \begin{cases} CH(CH_{count}) = CM(\max_{j \in n} Q_j), & \text{if } D_{j, CH(1:CH_{count}-1)} \geq T \\ CH_{tot} = CH_{count}, & \text{if } node_{counter} = n \end{cases} \quad (4.2)
\end{aligned}$$

Algorithm 1 represents the method used with the help of the last equation for CH election and keep the lection out only when $D_{i,sink}$ is greater than the transmission range T_{range} . The next step of the method, $node_{counter}$ is used to count the number of examined nodes and prevent loops caused by the distance threshold.

Algorithm 1

```

While  $CH_{counter} \leq CH_{tot}$ 
  for  $i = (1:n)$ 
    if  $node(i).E > 0$ , then
      if  $D_{i,sink} < T_{range}$ , then
         $node_{counter}++$ 
      else
        if  $node(i).Q == \max(node[:].Q)$  and  $D_{i,CH[:]} \geq T$ , then
          Set  $CH(CH_{counter}) = node(i)$ 
           $CH_{counter}++$ 
           $node_{counter}++$ 
        else
           $node_{counter}++$ 
        end
      end
    end
  if  $CH_{counter} > CH_{tot}$ , then
    break
  end
  if  $node_{counter} = n$ , then
     $CH_{tot} = CH_{counter}$ 
    break
  end
end
end

```

The rule of this algorithm is to check if a number of CH_{tot} nodes has been elected as a CH, or all nodes has been already examined, then CH election phase is completed. The second part of this condition will

update the number of CH_{tot} if it is met, as all nodes have been examined and no more CHs to be elected. mEER-RL CH election method uses the condition to stop electing nodes in case of the current setup is satisfactory for the requirements.

- CH distance threshold formula: Instead of manually calculating and assigning the distance threshold between CHs, or keeping a fixed value which is mostly not suitable for all conditions and scenarios, a new method is proposed to make the protocol find the optimal threshold according to the network parameters, which will help the RL agent take better routing decisions and optimize the learning process. The distance threshold between the CHs should efficiently divide the network into more balanced clusters. We have made an analysis to find a general formula to be suitable for all possible conditions, because it depends on the area size, the number of existing nodes and the CH%. Equation (4.3) shows the formula we have ended with trial and error, depending on the inputs to reach the desired output. This value will be calculated before the set-up phase and before re-clustering, which does not need any RL algorithm to find it, as it will add unnecessary computational overhead.

$$D_{threshold} = \frac{2 \times \sqrt{x \cdot y}}{CH_{tot}} \times \frac{\sqrt{(x_{sink})^2 + (y_{sink})^2}}{100} \quad (4.3)$$

In the formula, x and y represents the 2D network field size in meters, while x_{sink} and y_{sink} are the coordinates of the centered BS. Table 8 shows the results of different input parameters of the network to find $D_{threshold}$ using equation (4.3), simulation comparison results are in later sections.

Table 8 Results of different network input parameters to calculate CH distance threshold using equation (4.3)

Example #	1	2	3	4	5	6
$x.y$	100^2	100^2	100^2	200^2	200^2	200^2
x_{sink}	50	50	50	100	100	100
y_{sink}	50	50	50	100	100	100
n	100	100	200	100	200	200
$CH\%$	0.1	0.15	0.1	0.15	0.1	0.15
CH_{tot}	10	15	20	15	20	30
$D_{threshold}$	14.1421	9.4281	7.0711	37.7124	28.2843	18.8562
$ceil(D_{threshold})$	15	10	8	38	29	19

4.3 Alive and Dead Nodes Parameters Update

Once a CH has reached the energy threshold of being in the same role, re-clustering is required. Before the re-clustering phase starts, the method re-calculates the alive nodes. In order to make more accurate filtering decisions, the Q-value of all dead nodes should not be considered during all the three phases when re-clustering is required, in order to prevent the examination of dead nodes values. This is because network clustering and optimal route selection relies on the updated Q-value to make the RL agent learns the optimal policy. All nodes are updated with any node state change within the network.

We have modified the method to be more careful with alive and dead nodes parameters by updating the Q-value of each dead node according to Equation (3.10) before re-clustering, dead nodes will always have -100 as a negative reward, which will make the RL agent dismiss selecting a route with negative reward. Also, simply examine the Q-value of alive nodes only during cluster formation and data transmission. This modification makes more accurate results and higher performance.

Algorithm 2 represents the modified method that examines the residual energy “*node.E*” of each node to separate alive and dead nodes, and updates the Q-value “*CMnode.Q*” of the alive node with the most recent value “*node.Q*”. While dead nodes values are updated to have alive condition parameter “*node.cond*” to be changed to 0 (not alive) and the Q-value to -100.

Algorithm 2

```

alive = 0
for i = (1:n)
...
    if node.E > 0, then
        CMnode.Q = node.Q
        alive ++
    else
        node.cond = 0
        CMnode.Q = -100
    end
end
end
.
.
.
... >> CH election phase << ...
... if CMnode.Q == max(node[:].Q) AND node.E > 0, then
... >> CH election phase << ...

```

4.4 Re-clustering Phase Rearrangement

Improper arrangements during communication phase can lead to several issues. One problem is the transmission of duplicate packets within a single round. This occurs when data is sent before the re-clustering condition is being checked and CHs are re-elected, resulting in packets being sent to the old CH instead of the new one, which wastes energy for all nodes involved. Additionally, as the nodes are re-assigned to new clusters, the incorrectly sent packets may fail to reach their intended destination and are ultimately dropped by the receiver node due to the

misleading transmissions, which falsely indicate the packet was successfully transmitted during its assigned time slot of the schedule. Another potential issue is the presence of confused nodes that mistakenly belong to multiple clusters, causing them to send duplicate packets through different CHs. This can be identified by monitoring the number of packets transmitted per round, which should not exceed the total number of alive nodes.

- In mEER-RL, the arrangement of the communication phase is modified to prevent misleading transmissions and energy wastage. By checking the re-clustering condition in the beginning of each round (episode), it ensures all communication starts only after nodes have been re-assigned to the new clusters and all cluster have been formed, then allowing safe packet transmission between CMs and their respective CHs. Similarly, the communication between CHs and nodes within the BS range is reorganized to prevent duplicate communication between old and new CHs. The method essentially checks if an existing CH has reached a critical energy level (pre-defined threshold). If so, it triggers the re-clustering phase before any communication within the network begins. As shown in Figure 11, mEER-RL verifies the CH energy according to the pre-defined threshold at the start of each round.

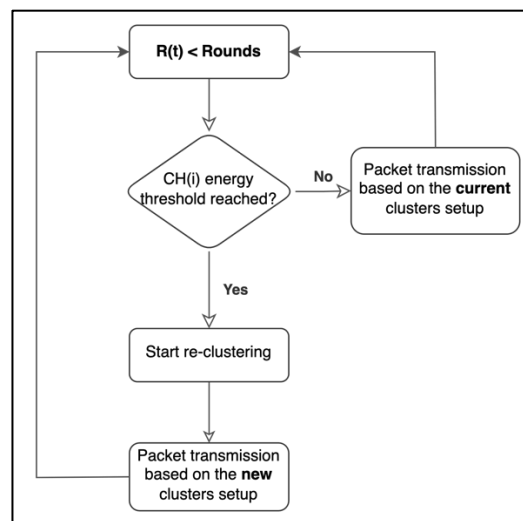


Figure 11 CH re-election process during communication phase

- mEER-RL tackles the issue of nodes mistakenly belonging to multiple clusters, which can lead to duplicate packets in the network reaching to the BS. To prevent this confusion and ensure proper assignment, the protocol mEER-RL employs a two-step approach:
 - Clearing previous CM assignments: Before re-clustering, all clusters data is reset from the network, allowing for an effective clearance of previous cluster information and assigned nodes data. It also prevents newly dead nodes from the consideration.
 - Assigning nodes: A specific parameter “Node.cluster” in each node is reset to “False”, indicating it’s not currently assigned to any cluster. After CH election, clusters are formed by considering the distance between each node and the CHs, the closer CH will allow the node to join its cluster. Only alive nodes with the “False” value of “Node.cluster” parameter are included in a cluster, ensuring they belong to only one. This parameter is then set to “True” for each node after successful cluster assignment.

Figure 12 illustrates the above process. It shows how mEER-RL clears all clusters information from the network which is previously broadcasted to all nodes. During cluster formation, the “Node.cluster” parameter of each node is checked before assigning it to a cluster in addition to the minimum distance to a CH. This process continues until all alive nodes in the network are processed.

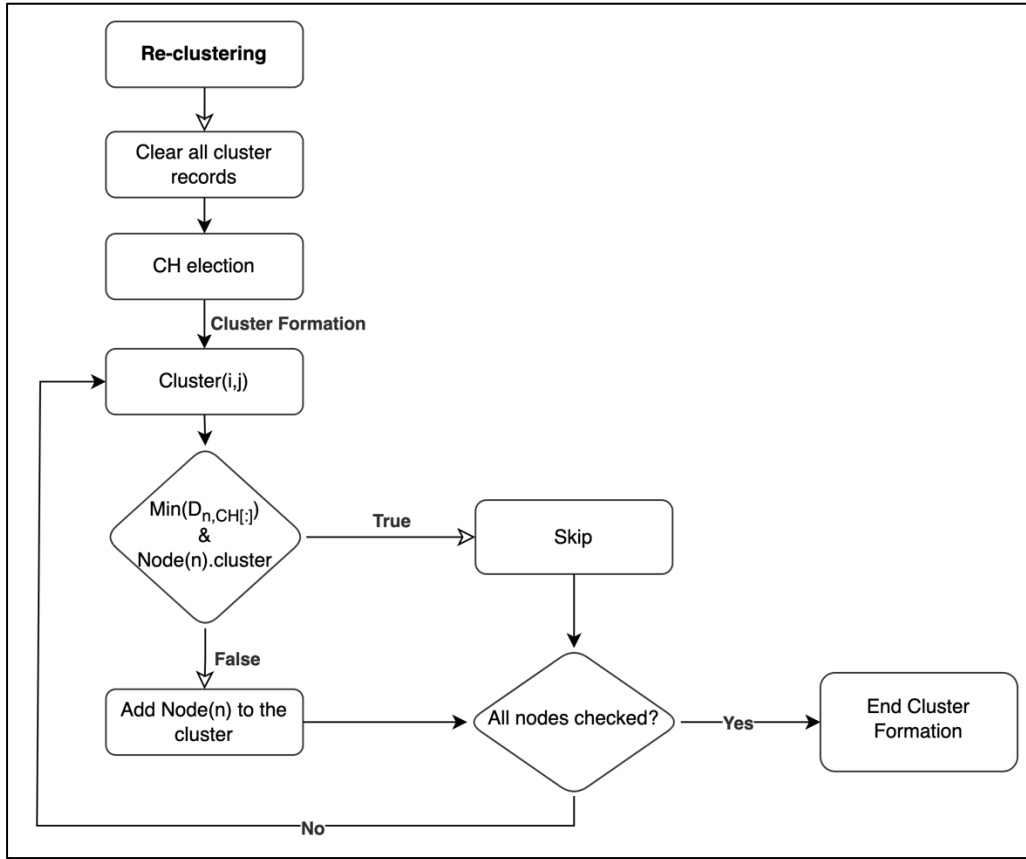


Figure 12 Simplified cluster formation process in mEER-RL

4.5 Intra and Inter-Cluster Multihop Communication

Ensuring timely and efficient data delivery to the BS within a single round is crucial in IoT networks. With the consideration of the TDMA schedule limitations and the varying distances between nodes, mEER-RL addresses this challenge by implementing the following strategies to avoid packet delivery failures and delays:

- Multihop communication optimization: mEER-RL employs an enhanced Multihop communication mechanism that aims to reduce energy consumption and enables the RL agent to discover more effective data routing strategies with larger action and state spaces. This mechanism operates on two levels:
 - Intra-cluster Multihop: This allows CMs outside the CH's direct transmission range to relay data for each other, facilitating data

delivery to the CH from distant nodes. This approach avoids excessive energy consumption and minimize it compared to individual long-range transmissions.

- Inter-cluster Multihop: This involves communication between CHs and with NS nodes. Depending on the network configurations and defined range of NS nodes, a CH within the BS range might transmit data directly, eliminating the need for relaying through NS nodes. This further optimizes energy consumption and network efficiency.
- Hierarchical communication organization: leveraging the established cluster structure, mEER-RL organizes data transmission hierarchically for efficient routing. This is done by allowing nodes within a cluster to transmit data to the CH in a predefined order. Prioritizing CMs furthest from their CH, followed by closer ones, in order to minimize potential packet delivery delays within the TDMA schedule of intra-cluster communication. Similarly, for inter-cluster communication, prioritization between CHs and NS nodes depends on their relative distance to the BS. Therefore, each node (CM, CH, NS) verifies data reception from other nodes before transmitting. This ensures the CH gathers all data from its CMs before sending an aggregated packet towards the BS. This hierarchical structure ensures all nodes transmit their data within their designated time slots, preventing data delivery delays and missed packets.

By combining these strategies, characterized by hierarchical organization and Multihop communication, mEER-RL aims to achieve timely and reliable data delivery within a single round, minimizing energy consumption and allowing the RL agent to discover and learn optimal data routing policies.

Figure 13 illustrates the intra-cluster Multihop communication mechanism implemented within mEER-RL. This mechanism addresses the challenge of CMs located beyond the direct transmission range of their designated CH. The primary objective of this mechanism is to optimize energy efficiency, minimize packet delivery delay and facilitate the exploration of diverse packet routing strategies by the RL agent.

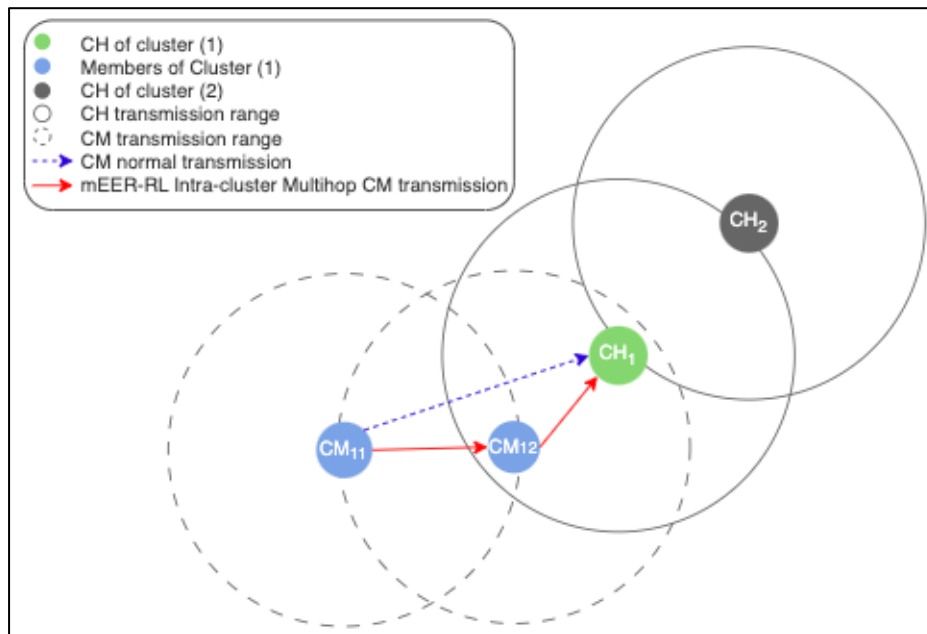


Figure 13 mEER-RL Intra-cluster Multihop communication

The intra-cluster Multihop communication process works as follows:

1. CM range limitation detection: Once a CM determines its inability to transmit data directly to the CH due to exceeding transmission range, it searches for another CM situated closer to the CH. This identified CM serves as a relay for data transmission.
2. Priority-based transmission: Farther nodes are prioritized for data transmission within their designated time slots assigned according to the distance to CH, all CMs are updated with any node state change within the cluster. This prioritization minimizes potential delays for nodes further from the CH.

3. Relay confirmation and data collection: Before a relay CM proceeds with its own transmission to the CH, it verifies the reception of any data packets from farther CMs, then safely transmits to the CH. This step ensures seamless data forwarding and prevents packet losses towards CHs.
4. Energy efficiency and RL exploration: This Multihop approach within the cluster reduces energy consumption compared to the individual long-range transmissions directly from faraway CMs, extending network longevity. Additionally, the ability to explore alternative routing paths through different relay CMs empowers the RL agent to learn and identify the most efficient data transmission policies over time through continuous exploration and data gathering for the Q-table.
5. Direct transmission as a last resort: There might be an uncommon scenario or after most of the network nodes has died, where no closer CM is found to act as a relay for far nodes. The far nodes resorts to direct transmission to the CH, even if this approach requires higher energy consumption. This fallback option guarantees that all CMs data reaches the CH, at the cost of increased energy consumption in such situations.

Figure 14 shows the inter-cluster Multihop communication implemented within mEER-RL to facilitate CH data delivery when the intended destination (NS or BS) falls outside its direct transmission range. Similar to intra-cluster Multihop communication, this mechanism aims to optimize energy consumption and empower the RL agent to discover diverse routing strategies.

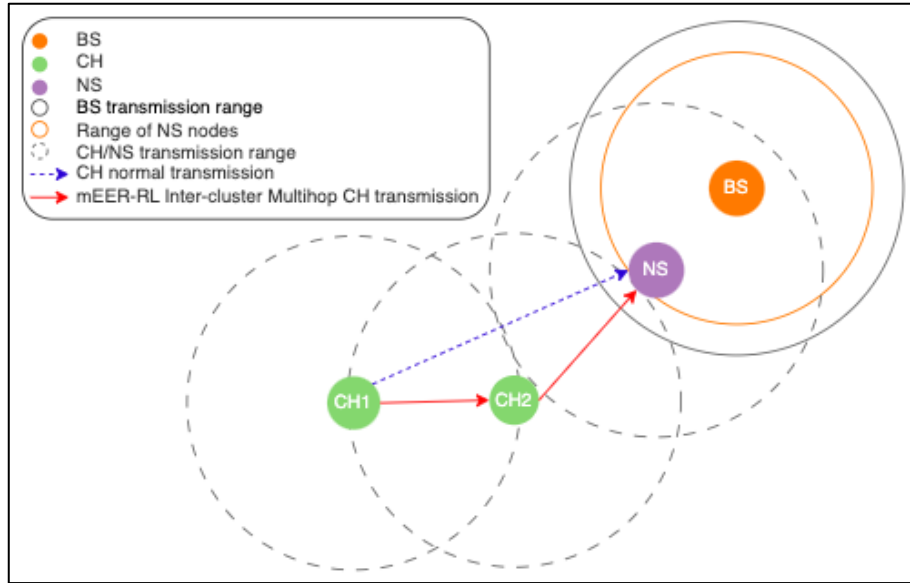


Figure 14 mEER-RL Inter-cluster Multihop communication

The process works the same as intra-cluster Multihop, with minor notes as follows:

1. Destination beyond range detection: The destination in this step to be identified as the BS or NS node. Where the CH searches for another CH to work as a relay once the intended destination is beyond its transmission range.
2. Priority-based transmission: Farther CHs are prioritized for data transmission within pre-established slots that are agreed and calculated by the CHs during clustering phase based on the broadcasted distance to BS.
3. Relay confirmation and data collection: The relay CH and NS nodes verifies the reception of any data packets from other CHs before proceeding with its own transmission towards the BS.
4. Energy efficiency and RL exploration: Similar to intra-cluster Multihop communication, this approach offers energy conservation and extending network lifetime, in addition to offering more alternative routing paths to be explored by the RL agent to identify the most efficient data transmission policy.

5. Direct transmission as a last resort: Far CHs may resort to direct transmission to the destination even when it requires higher energy consumption in order to guarantee the delivery of aggregated data to the BS.

It is important to note that the RL agent cannot discover a “better route” when a direct transmission scenario arises (both intra and inter-cluster). However, both Multihop mechanisms serves as a valuable alternative for energy-efficient data delivery and delay minimization in most situations, offering the RL agent a broader state-action spaces for policy optimization and network performance enhancement.

As the CHs physically closer to the BS, more viable routes are available for the RL agent for exploration. While utilizing NS nodes can be an option, it may not always be the most efficient choice. Figure 15 represents a scenario where mEER-RL identifies an alternative route that bypasses the NS node. In this case, if the BS is already within direct transmission range of the CH, the RL agent has an option of direct data transmission to the BS within its action space. The availability of this route expands the exploration space available for the RL agent, it also allows it to evaluate the current state for more efficient reward maximization. This is aiming to allow the RL agent to evaluate the energy efficiency of direct transmission compared to the Multihop approach, assess potential delays associated with each route considering packet reception and distance to BS, and also to learn and adapt its policy based on the gathered data through exploration for efficient and reliable data routing.

Noting that the availability of a direct transmission option does not reduce the importance of NS nodes, where the CHs that falls outside the BS’s direct range still requires the NS node, that makes it a vital component of the Multihop routing mechanism for efficient data delivery.

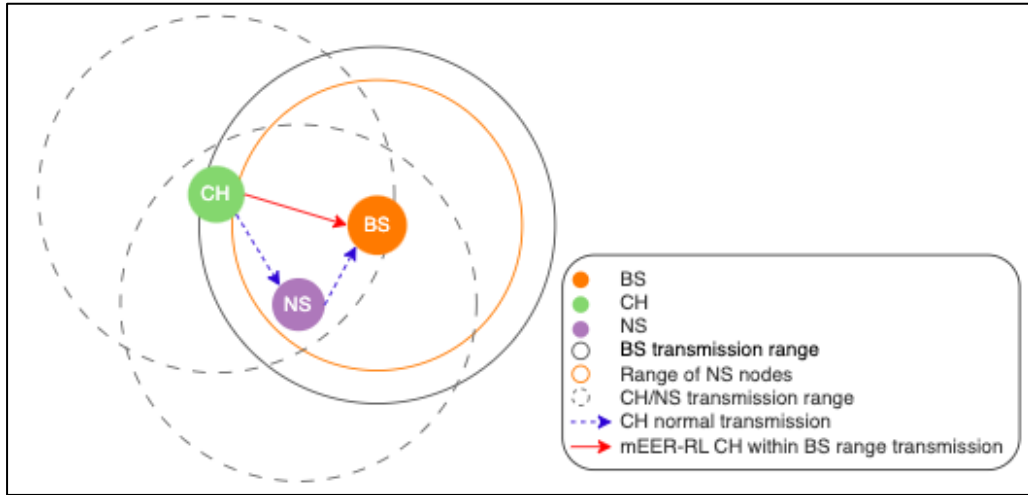


Figure 15 mEER-RL CH within BS range communication

All priorities are being arranged for mEER-RL data routing, and more routes are available for the RL agent to explore:

- Intra-cluster:
 - CMs outa of CH range can transmit through closer CMs, possibly can involve more than 2 hops for long paths.
 - CMs closer and within CH range transmit after checking for received packets, according to the agreed slots.
 - All CMs nodes agent explore routes to the CH.
- Inter-cluster:
 - CHs aggregate CM data and explore routes to the BS.
 - CHs outa of BS range can transmit through closer CHs or NS nodes, possibly can involve more than 2 hops for long paths.
 - All CHs and NS nodes transmit after checking for received packets according to the agreed slots.
 - CHs within BS range have the option of direct transmission.

The expanded exploration space provided by mEER-RL allows the RL agent to achieve improved learning outcomes. This is accomplished through a better trade-off between exploration and exploitation, ultimately accelerating the learning process towards optimal data routing strategies.

By exploring a larger state space, the RL agent can identify diverse solutions by the consideration of various routes and networks configuration. It also can avoid overfitting by not only relying on a limited set of few routes.

4.6 Dynamic Learning

In section 2.3.2, we have learnt that a dynamically adjusted learning rate (α) can encourage exploration which stabilizes the learning and agent policy convergence, this can be achieved while keeping the exploitation in the consideration. α (alpha) is a learning rate parameter that controls the balance between exploiting existing knowledge (stored in the Q-table) and exploring new routing options.

Our RL agent is working in a complex environment which tends to have unsteady learning performance, as we have seen through the previous chapter. Therefore, a function has been added to the method that decreases the learning rate value during the learning process. Equation (4.4) shows the simple method followed to decrease the learning rate value by time.

$$\alpha(r + 1) = \alpha(r) \times \varepsilon \quad (4.4)$$

Where r = round number, and ε is a decreasing factor that it is optimized and found to be equal to 0.75, more details are discussed in the simulation section. However, the decreasing value is allowing the RL agent to improve convergence to the optimal policy by the exploration and exploitation while it progresses, it allows to adapt to the changing data distribution to avoid making large updates based on irrelevant data during the training process, which can be identified as spikes through the evaluation and monitoring.

A constant learning rate leads to overshooting the optimal solution and fluctuating around it and does not allow proper exploitation. Thus, by gradually decreasing the learning rate, it helps the model to fine-tune its

parameters and converge more precisely towards the optimal policy. In addition, this helps to improve stability and makes the optimization process smoother and more stable in such complex environments, which is full of changing dynamics that performs similar to noisy data.

During the simulation (discussed in the next section), we observed that the performance of mEER-RL was improved before reaching the FND. This observation led us to investigate a method for resetting the value of α within the RL agent. We explored two strategies for α reset:

- Reset after each re-clustering phase: This approach assumes that re-clustering significantly alters network topology. Resetting alpha encourages the agent to re-evaluate existing routes and potentially discover more efficient paths after the network reconfigures.
- Reset at the end of a round that has dead nodes: This strategy focuses on adapting to node failures. Dead nodes indicate changes in network connectivity. Resetting alpha allows the agent to explore alternative routes that bypass failed nodes.

Wireless IoT networks are dynamic environments. As nodes drain energy and die, or network conditions fluctuate, existing routes might become less efficient. Continuing with the decreased alpha value may not adequately balance exploitation and exploration for optimal learning during the rounds following FDN. Resetting alpha allows the agent to prioritize exploration in response to environmental changes, potentially leading to the discovery of improved routing strategies towards the destination.

This approach differs from resetting the Q-table entirely. Here, we're specifically adjusting the learning rate to encourage exploration when the environment changes.

4.7 Modified Q-function and Reward: Balancing Energy and Reliability

The design of the Q-function and reward function significantly impacts the learning process of the RL agent in routing protocols. These functions determine the information considered by the agent during decision making in a specific network state. Ideally, the model should pull the most relevant network information for optimal learning, leading to well-informed routing decisions taken by the agent.

In this work, we explored various modifications to the original EER-RL Q and Reward functions. These attempts involved incorporating factors like PDR, residual node count or minor changes. However, most modifications resulted in either no performance improvement or decreased performance.

Despite these initial challenges, we succeeded at a modified initial Q-function (Equation 4.5) and reward function (Equation 4.6) for a noticeably improved network lifetime and delayed the FND (the higher value, the better). An essential addition to these functions is the inclusion of PDR. This ensures that the agent prioritizes paths with minimal energy consumption while also enabling most nodes to successfully deliver their data. Focusing on minimizing energy consumption can lead to negative performance, as the agent might learn to choose paths that compromise data delivery which was observed during the test runs and protocols evaluations.

$$Q = \begin{cases} \frac{1}{N_h^{CO}}, & \text{if } E_{max} = E_{min} \\ p \times \left(\frac{E_r - E_{min}}{E_{max} - E_{min}} \right) + q \times \frac{1}{N_h^{CO}}, & \text{if } E_{max} \neq E_{min} \end{cases} \quad (4.5)$$

$$r_{t+1} = \begin{cases} \frac{1}{N_h^{CO}}, & \text{if } E_{min} = E_{max} \\ p \times \left(\frac{E_r - E_{min}}{E_{max} - E_{min}} \right) + q \times \left(\frac{1}{N_h^{CO}} \right) - \beta \times \left(1 - \frac{SentP_{(r-1)}}{Aln} \right), & \text{if } E_{min} \neq E_{ma} \\ -100, & \text{if } E_r \leq 0 \end{cases} \quad (4.6)$$

Within the above equations, CO is a constant value determined with the initial network parameters, the determined best value of the constant is found to be equal to “7”. β represents a probabilistic value for the PDR, integrated for the learning process. Additionally, $SentP_{(r-1)}$ denotes the number of packets sent in the previous round, and Aln represents the number of alive nodes after the last round completed. By incorporating the PDR and carefully balancing it with energy consumption, the modified Q-function and reward function enable the RL agent to make more accurate routing decisions that optimize both network lifetime and data delivery success.

4.8 Node Mobility Consideration in mEER-RL

The core functionality of mEER-RL operates in a model-free IoT network, where each node aims to transmit data to the BS while concentrating on the goals of minimizing energy consumption, minimizing routing hops, and maximizing PDR at the same time. However, the consideration of mobility adds a layer of complexity to the environment. To maintain efficiency, the RL agent within mEER-RL must consider additional parameters that influence improved routing decisions in a dynamic setting.

We propose an algorithm integrated into mEER-RL that specifically addresses node mobility throughout the network lifetime. Node movement directly impacts the protocol’s efficiency as nodes enter and leave clusters due to changes in location and distance to the BS and CHs. This is

continuously changing the network structure and cluster arrangements. These movements also invalidate the previously learned Q-values, which guides the RL agent's decision making.

The proposed mobility model employs a simple random walk approach. In each round, the location of each alive node is updated using a random bounded acceleration value. This approach shares similarities with common methods for exploring the state space within an RL environment. However, in this context, the RL agents are not actively exploring the area but rather moving randomly while simultaneously exploring optimal packet routing options as moving.

Equations (4.7) and (4.8) represents the method of updating the new location of a mobile node for both x and y coordinates. Here, V_{t+1} represents the new vector point for both x and y coordinates, V_t is the current round vector point, and ε_t is a conditional random value. This equation models the uncertainty in the of mobile nodes movement.

The conditional random value is used to introduce random changes in the node's velocity, consequently updating its location. This value increments or decrements the x and y coordinates by a value between -0.5 and 0.5 meters within the defined network area. In case a node approaches the edge of the network area, then the value is adjusted to move the node closer to the center of the controlled area. " $edge_-$ " and " $edge_+$ " represents the lower/left and upper/right edges of the environment, respectively. " $rand \{ \}$ " is a function used to generate the random value within the defined movement limits.

$$V_{t+1} = V_t + \varepsilon_t \quad (4.7)$$

$$\varepsilon_t \in \begin{cases} rand \{0,0.5\}, & \text{if } V_t \leq edge_- + 0.5 \\ rand \{-0.5,0\}, & \text{if } V_t \geq edge_+ - 0.5 \\ rand \{-0.5,0.5\}, & \text{otherwise} \end{cases} \quad (4.8)$$

However, once a new location point is detected, then each node calculates and updates its distance to the BS. This updated distance directly affects the next Q-value, ensuring the RL agent makes the best learnt routing decisions based on the current network configuration. Subsequently, nodes within the BS transmission range checks if they remain within that range. If not, they join the closest CH cluster. Similarly, a node within a cluster checks if it became within the BS transmission range which will make it opts out of the cluster and transmit its data directly to the BS.

Also, nodes evaluate if they have moved closer to a different CH compared to their current one. If the distance difference exceeds a pre-defined threshold (accounting for movement speed to prevent unnecessary cluster switches), the node opts out of its current cluster and joins the closer CH's cluster. This threshold helps maintain network stability while allowing for adaptation to significant movement.

All nodes periodically check their distance to their assigned CH. If a node discovers a closer CH exceeding the threshold, it initiates a re-clustering process. The threshold value of changing to a new cluster is chosen for the consideration of the movement speed, that the node may stay within the same range of the current cluster in the next few rounds or may change the direction. Finally, the protocol will continue to calculate the Q-value and update the nodes by re-clustering through the previously discussed phases of CH election, cluster formation and data transmission.

Figure 16 shows the flow of modified condition of inert-cluster communication within the mEER-RL protocol. Each CH performs two checks before allowing any data exchange through the network, CH energy compared to the threshold and cluster change trigger. The latter is a

parameter that is initially set to “0”, serves as a flag for mobility-motivated re-clustering. If any node detects significant movement that necessitates re-clustering within the network, this flag is triggered, changing its value to "1". These checks are conducted periodically within the communication phase before any data exchange to ensure the CM remains connected to its designated or new CH. Communication attempts with an unreachable CH (e.g., due to the CH being out of range or belonging to a different cluster after movement) could lead to unnecessary disconnections. Also, NS nodes may require joining new clusters, or new nodes could be set as NS node. This encourages the capability of this protocol to be adaptive for new nodes joining the network.

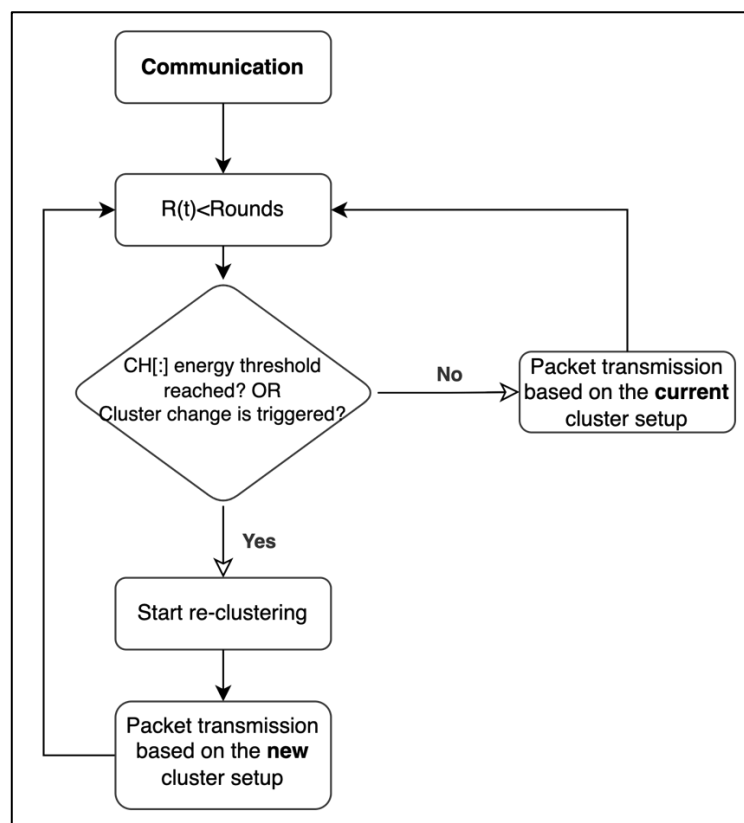


Figure 16 Modified inter-cluster communication condition for mobile mEER-RL

At the end of each round, Equations (4.7) and (4.8) are utilized to update the location (x and y coordinates) of each alive mobile node within

the network. Consequently, the distance between each node and the BS is recalculated based on the updated positions. This change in node location may or may not require re-clustering. Three specific scenarios trigger a re-clustering process, as illustrated in Figure 17 flow:

- **BS range exit:** If a node that was previously within BS's transmission range moves out of range, it can no longer communicate directly with it. This node will need to join a cluster for network access.
- **BS range entry:** Contrariwise, a node within a cluster may move closer to the BS and enter its transmission range. In this scenario, the node no longer requires cluster membership and can communicate directly with the BS for improved efficiency. It will therefore leave its current cluster.
- **Closer CH:** If a node detects a different CH that is at least 2 meters closer than its current CH, it will initiate a cluster switch request by setting the parameter "change" to "1". The node will leave its current cluster and join the cluster associated with the closer CH for potentially improved routing options.

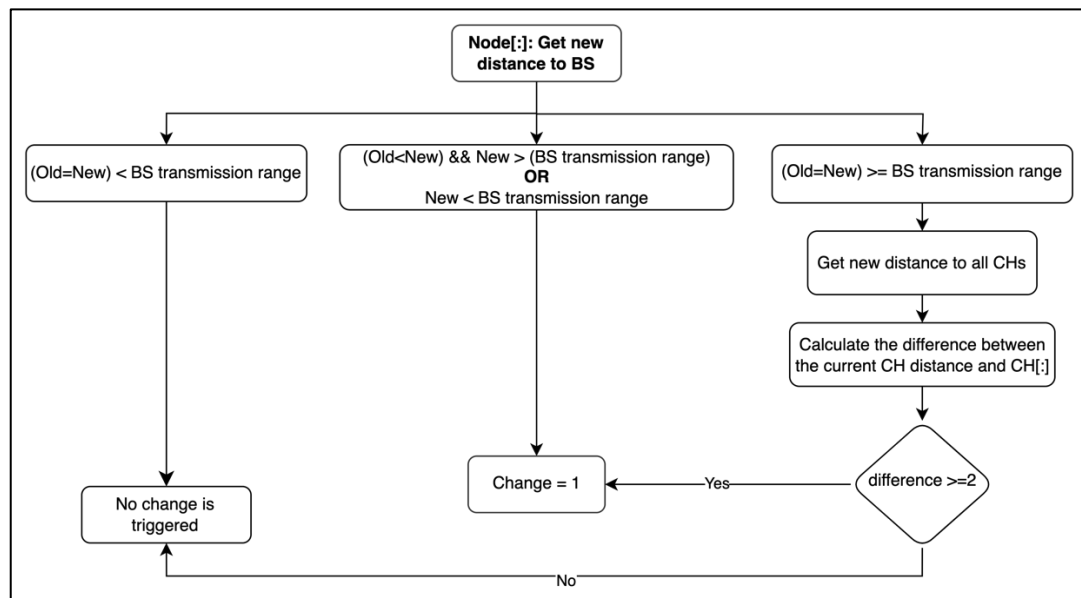


Figure 17 mEER-RL mobile nodes location and distance re-clustering trigger

Following a re-clustering event, all nodes are notified with the network changes. This ensures that Q-values, which guide routing decisions for each node, are updated based on the current network configuration. This leads to an efficient data transmission within the network.

4.9 Chapter summary

This chapter presented the modifications and contributions applied to the enhanced routing protocol mEER-RL; a modified protocol based on EER-RL. mEER-RL is specifically designed for energy-efficient and reliable data transmission in fixed and mobile IoT networks. This protocol incorporates several enhancements to the original protocol, including automated CH number calculation, NS nodes exclusion, enhanced distance threshold consideration between CHs, alive and Dead nodes parameters update technique, re-clustering phase rearrangement, enhanced Intra and Inter-cluster Multihop communication, dynamic learning and node mobility consideration.

These enhancements collectively contribute to mEER-RL's improved performance in terms of energy efficiency, network lifetime, and data delivery reliability.

Chapter 5: mEER-RL Simulation Results and Discussion

5.1 mEER-RL Simulation Setup

5.2 mEER-RL Simulation Results and Performance Evaluation

5.2.1 CH Election and Nodes Parameters Update

5.2.2 CH Distance Threshold Formula

5.2.3 Mitigating Packet Drops and Duplications in mEER-RL-v2

5.2.4 Modified Q-function and Reward: Balancing Energy Efficiency and Reliability in mEER-RL

5.2.5 Dynamic Learning

5.2.6 Mobile mEER-RL

5.3 Performance Comparison: Fixed vs. Mobile mEER-RL and Existing Protocols

5.3.1 RL-based Protocol Parameters

5.3.2 Network Setup and Clustering

5.3.3 Network Lifetime and Energy Consumption Evaluation

5.3.4 Distance Traveled Evaluation

5.3.5 Packet Delivery Evaluation

5.3.6 Results Summary and Discussion of mEER-RL Performance

5.4 Chapter Summary

Building upon the evaluation of LEACH and EER-RL protocols, we have employed MATLAB for performance insight, while methodically evaluating and improving the performance of mEER-RL. The simulation environment mirrors the framework established in Chapter 3, prioritizing the analysis of energy consumption and hop count optimization with the packet delivery through RL-driven routing.

5.1 mEER-RL Simulation Setup

For the simulation parameters, we follow a strategic balance between flexibility and focus. Table 9 presents the initial simulation parameters for the general analysis, differentiating between those that remain fixed and those open to tuning. This approach supports both a targeted analysis and adaptability.

Table 9 Simulation parameters for mEER-RL evaluation

Parameter	Initial value	Tuning
Sensing field	100*100 m	Yes
No. of sensor devices	100	Yes
No. of CHs	10% of total nodes	Yes
Sink location	Center (50,50)	No
Min. CH to sink range	15m	No
Transmission range	20 m	No
Data size	4000 bits	No
E_{elec}	50×10^{-9} joules/bit	No
E_{amp}	100×10^{-12} joules/bit/m ²	No
Initial energy	1-2 joules	No
α (learning rate)	1	Yes
γ (discount factor)	0.95	Yes
p (energy probabilistic)	0.5	Yes
q (hops probabilistic)	0.5	Yes
Total rounds	5000	Yes

Through the following sections, only the parameters accepting tuning in our simulation are presented, which are related to data routing and energy focus. Those parameters are the field size, number of nodes in the field, CH%, total number of rounds (episodes) and Q-function parameters within the RL agent (refer to Equations 3.11). Other parameters including E_{elec} , E_{amp} , transmission range, minimum distance of a CH to BS, data packet size and the initial heterogenous energy (1-2 Joules), are intentionally maintained as constants. This decision serves two purposes:

- Concentrated Focus: By isolating data routing-related parameters, we can accurately examine their impact on energy consumption dynamics, yielding to stronger insights.
- Default Node Capabilities: The fixed parameters reflect essential node capabilities, ensuring consistency and comparability across simulations.

The evaluation metrics are used to capture performance differences. To carefully evaluate mEER-RL protocol performance, we employ a multi-dimensional approach, combining the following metrics:

- EC: Maximum Consumed Energy (MCE) and Average Consumed Energy (ACE).
- Network lifetime: FND, LND and number of alive nodes after a specified number of rounds.
- Routing efficiency: average distance traveled by packets during routing, and total number of delivered packets throughout the journey.

A visualized simulation is presented with each comparison to provide an overview of the simulation parameter setting, in order to help in understanding the experimental design and following results.

5.2 mEER-RL Simulation Results and Performance Evaluation

In this section, we perform a comparison of outcomes, precisely examining the factors influencing mEER-RL's performance compared to the established protocols like LEACH and EER-RL. We employ a multifaceted approach, by studying the resulted performance of the modifications discussed in section 4.1 and analyzing the sensitivity of mEER-RL to both network parameters and its internal RL agent configurations. Each network parameter adjustment or addition to the protocol is examined, revealing its unique impact on performance metrics and overall network behavior. This analysis provides valuable understandings into the complex interaction between network dynamics and mEER-RL's decision-making agent.

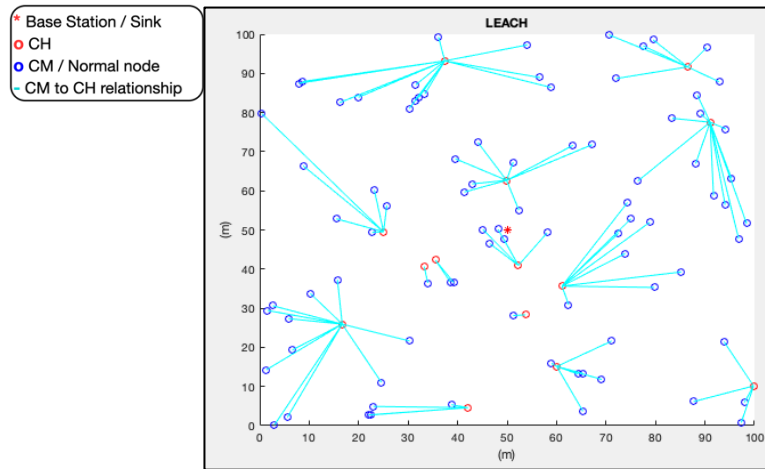
By analyzing the individual factors that influence mEER-RL performance, we gain a clearer understanding of the protocol's capabilities. We also prepare the base for future improvements and potential adaptations of its core principles. This comprehensive examination promotes a deeper understanding of mEER-RL's potential within the complex environment of energy-efficient IoT networks.

5.2.1 CH Election and Nodes Parameters Update

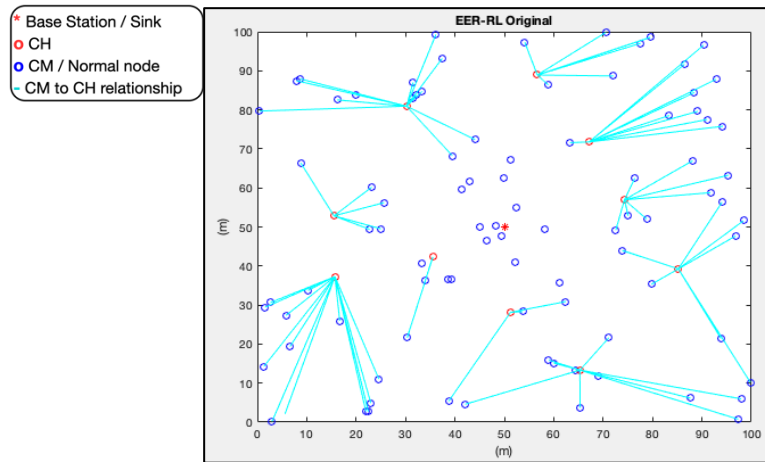
To measure the effectiveness of the modifications introduced in subsections 4.1, 4.2 and 4.3, we have conducted simulations using a common network configuration for all three protocols: LEACH, EER-RL and mEER-RL. All simulation parameters were set according to Table 9 (refer to the previous section).

Once the protocol receives the network parameters, it starts the initial calculation for CH election and cluster formation, Figure 18 shows the

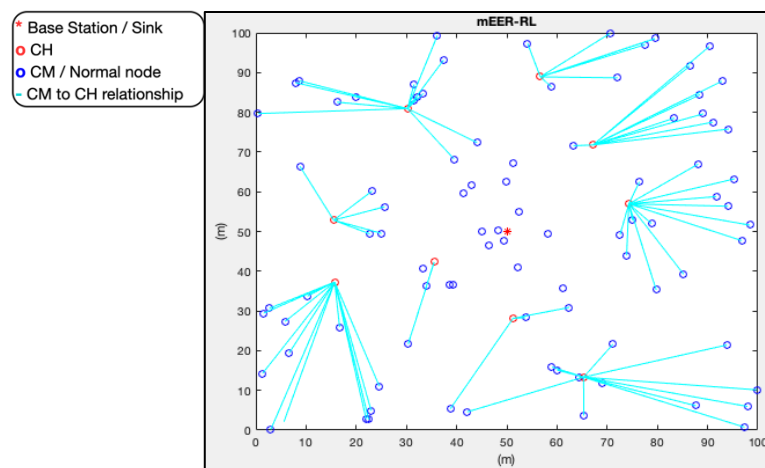
clusters formed by each protocol, with an observable difference in the behavior between them.



(a) LEACH clusters



(b) EER-RL clusters



(c) mEER-RL clusters

Figure 18 Protocols cluster formation during the setup phase comparing mEER-RL CH election method

LEACH formed 13 clusters, exceeding the desired percentage. This behavior is recognized to the dependence on the random value found by each node within the network for CH election. This is also potentially leading to elect CHs closer to the BS than their associated nodes.

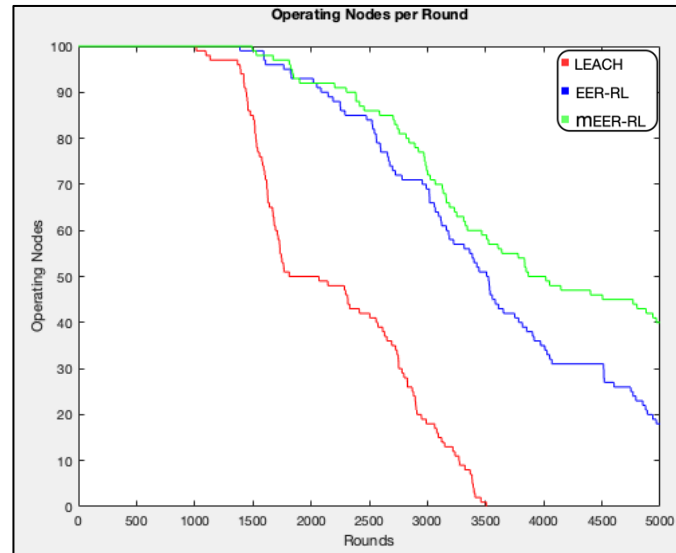
EER-RL formed 10 clusters, aligning with the calculated required CH number of total nodes. mEER-RL formed 9 clusters, satisfying the desired CH percentage while excluding the number of NS nodes from total nodes that are allowed to join clusters.

During the following rounds, nodes transmit data through their CHs. LEACH employs single-hop communication to the BS, while EER-RL and mEER-RL utilize multi-hop communication for improved efficiency. Re-clustering occurs in each round for LEACH, whereas EER-RL and mEER-RL trigger re-clustering based on the CH reaching energy threshold. The communication process consumes energy, and once the threshold is reached, re-clustering is required to maintain network functionality. The clusters are formed based on the CH election method of each protocol.

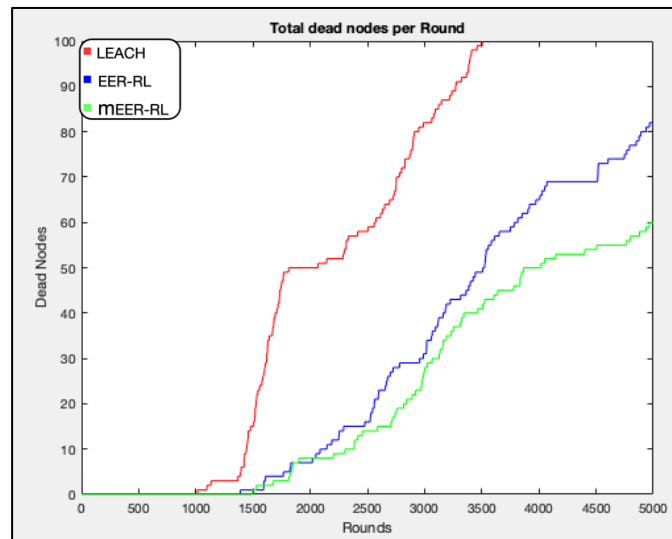
Equations (4.1) and (4.2) are used by mEER-RL for CH election, enabling a more accurate calculation of CHs and clusters before clustering compared to other protocols. This approach also provides more accurate Q-value consideration for routing decisions. However, the protocol may sometimes skip electing one or two CHs from the desired percentage only if there are no more nodes that meets the minimum distance requirement between CHs. Our observations suggest this occurs infrequently, and the desired CH percentage is usually achieved. Equation (4.3) for distance threshold adaptation is explored in the next section.

Figure 19 compares the performance of the protocols sampled for 5000 rounds by observing the number of alive and dead nodes. LEACH shows a rapid increase in dead nodes after 1000 rounds, ending the network

lifetime after 3500 rounds. Conversely, both EER-RL and mEER-RL determined superior performance with a slower rise in dead nodes, resulting with a longer network lifetime due to lower energy consumption.



(a) Operating nodes per round



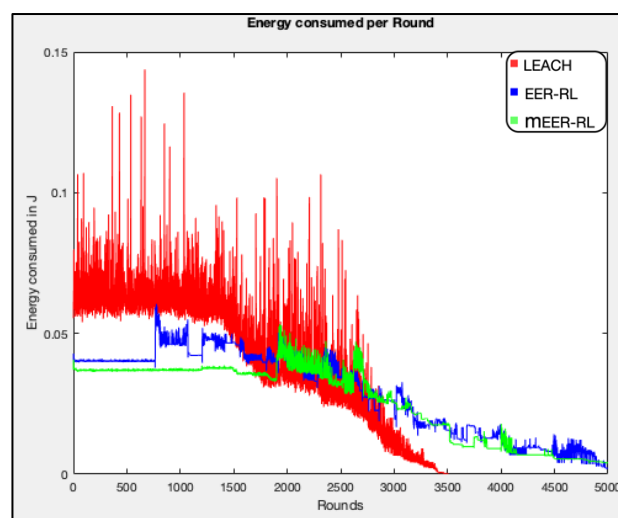
(b) Dead nodes per round

Figure 19 Operating and dead nodes per round for evaluating CH election and distance threshold consideration while accurately updating the parameters, 5000 rounds

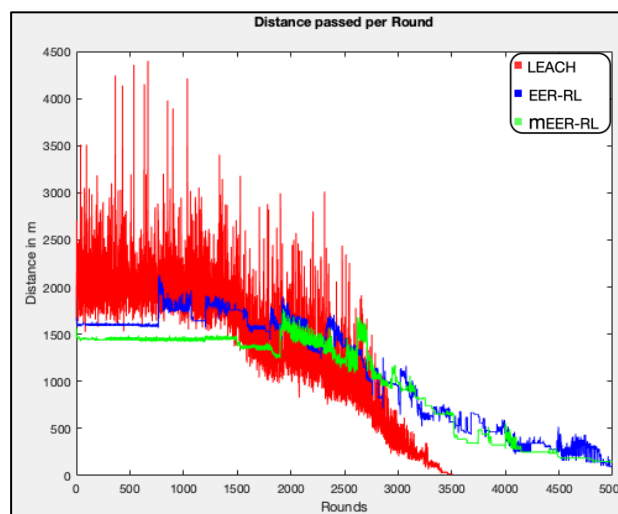
We can further observe how mEER-RL's implemented methods enhance the performance and maintain a higher number of alive nodes until the final round. This is due to the combined effect of the CH election

method, consideration of the distance threshold between CH nodes during election, and accurate updates to the alive and dead nodes parameters.

Figure 20 analyzes the consumed energy and traveled distance for 5000 rounds. LEACH shows the highest energy consumption for not considering the distance between CHs and the BS. This can lead to significant energy consumption and longer traveled distance, especially for faraway CHs receiving data from distant CM and sending it to the BS separately, without multi-hop aggregation techniques.



(a) Consumed energy per round



(b) Traveled distance per round

Figure 20 Consumed energy and traveled distance per round for evaluating CH election and distance threshold consideration while accurately updating the parameters, 5000 rounds

Additionally, LEACH's inefficient clustering results in unbalanced network behavior. mEER-RL outperforms the other protocols in terms of energy consumption, it proves that CH election and cluster formation occur more efficiently, leading to RL agent discovery of shorter routes that requires less energy.

The observed trends in energy consumption and traveled distance is referring to the design of the reward Q-value. Where the likelihood of selecting the path is balanced with the energy consumption. A more detailed examination of the RL parameters' influence is presented in a separate section.

Table 10 summarizes the simulation results, highlighting mEER-RL's superiority across all evaluation metrics. Notably, mEER-RL achieves an FND at round 1496, significantly later than the other protocols. This indicates a longer network lifetime due to efficient energy utilization. Additionally, mEER-RL maintains a considerably higher number of alive nodes at the end of the simulation (40 nodes), showing a 220% improvement compared to EER-RL. The results of energy and distance also show that the RL agent effectively learns shorter routes with lower energy consumption.

Furthermore, the higher number of alive nodes with lower energy consumed shows the positive impact of CH percentage calculation in mEER-RL. Selecting an accurate number of CHs means efficient data transmission within the clusters.

While mEER-RL shows promising results, it is important to include an additional evaluation metric: PDR. Although mEER-RL presented better performance in other aspects, packet delivery was not considered during the evaluation of the surveyed protocols. Lower energy consumption might indicate a preference for the routes with lower cost but

potentially lower packet delivery success, which is against network efficiency goal. To balance between them and address this potential trade-off, the following sections explore improvements to mEER-RL that aim to prevent packet duplication while achieving a higher PDR.

Table 10 Protocols performance for evaluating CH election and distance threshold consideration while accurately updating the parameters, 5000 rounds

Evaluation metric	LEACH	EER-RL	mEER-RL
FND at round	1013	1389	1496
LND at round	3506	5000+	5000+
Alive nodes after 5000 rounds	0	18	40
Avg energy (J)	0.03	0.0291	0.026
Max energy (J)	0.1438	0.0596	0.0529
Avg distance (m)	984.5	1118	993.3
Packets delivered	221526	341750	337122

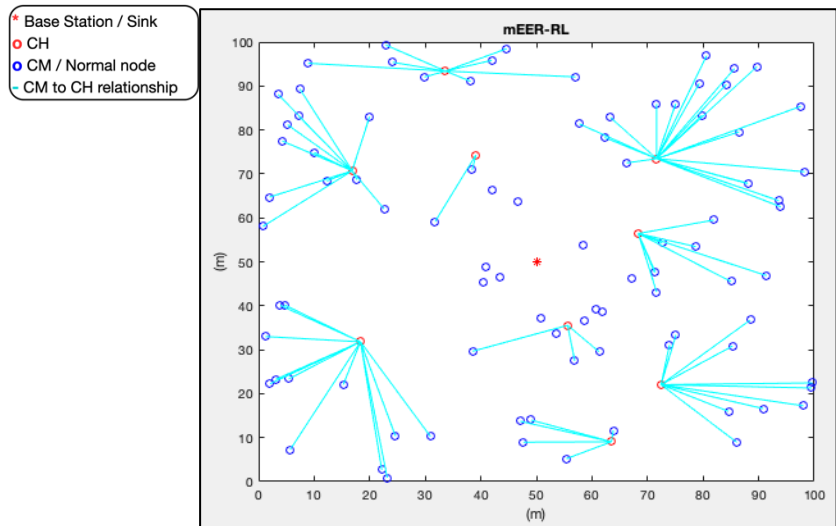
5.2.2 CH Distance Threshold Formula

This section evaluates the performance of mEER-RL after including a dynamic CH distance threshold formula Equation (4.3) to enhance its adaptability. We compare two versions:

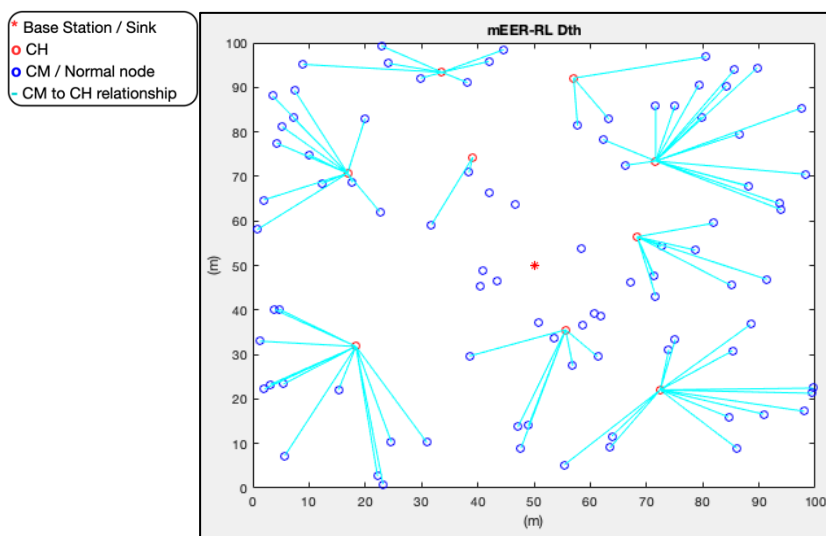
- mEER-RL with fixed threshold: This version utilizes the same fixed threshold values (15m for initial clustering and 20m for re-clustering) as EER-RL.
- mEER-RL with distance threshold formula: This version employs the dynamic CH distance threshold ($D_{threshold}$) formula in both initial clustering and re-clustering phases.
 - a. 10% CHs of total nodes number:

The simulations were conducted using the same default parameters for both versions. Figure 21 shows the clusters formed by each version for

a 10% CH percentage. We observe a difference in cluster arrangements. mEER-RL with the formula selects a CH in a different location due to the calculated $D_{threshold}$ during election, impacting cluster formation.



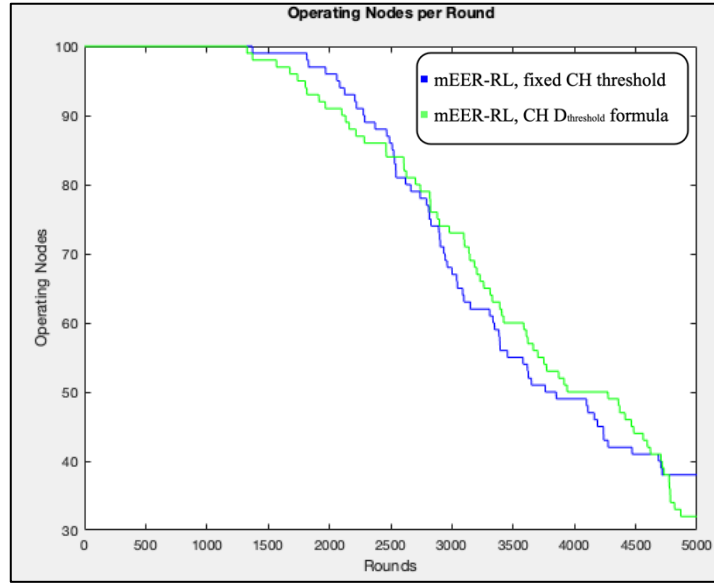
(a) mEER-RL with fixed CH distance threshold



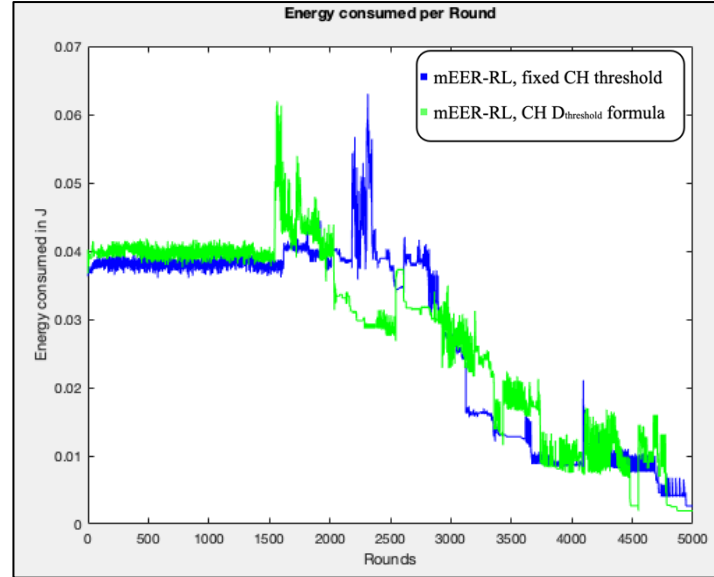
(b) mEER-RL with $D_{threshold}$ formula

Figure 21 mEER-RL performance with fixed and CH $D_{threshold}$ formula, 5000 rounds and 10% CH

Figure 22 (a) and (b) compare the number of alive nodes and energy consumption, respectively. While the performance is very similar, the $D_{threshold}$ formula introduces a slight increase in energy consumption, which in turn affects the number of alive nodes at the end of the simulation.



(a) Operating nodes per round



(b) Energy consumed per round

Figure 22 Operating nodes and energy consumed per round for evaluating mEER-RL CH $D_{threshold}$ formula, 5000 rounds

Figure 23 highlights unbalanced packet delivery behavior, with packet duplication occurring between rounds 2000-2500, impacting the performance of both versions.

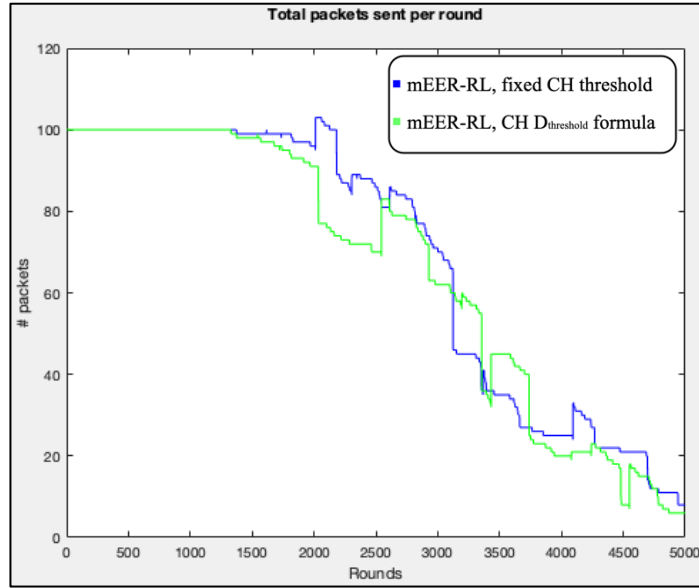


Figure 23 Total sent packets per round for evaluating mEER-RL CH $D_{threshold}$ formula, 5000 rounds

Table 11 summarizes the quantitative results. The new formula leads to an earlier FND at round 1328 compared to the fixed threshold version (FND at 1373) and fewer alive nodes at the end (32 vs. 38). This corresponds with the higher average energy consumption and longer traveled distances observed with the formula.

Table 11 Protocols performance for evaluating mEER-RL CH $D_{threshold}$ formula, 5000 rounds

Evaluation metric	mEER-RL, fixed CH threshold	mEER-RL, $D_{threshold}$ formula
FND at round	1373	1328
LND at round	5000+	5000+
Alive nodes after 5000 rounds	38	32
Avg energy (J)	0.0274	0.0278
Max energy (J)	0.063	0.062
Avg distance (m)	1093.4	1099.2

Repeated tests demonstrate inconsistent results with the formula, sometimes achieving better outcomes than the fixed threshold. This may show that the formula might be more beneficial for network configurations

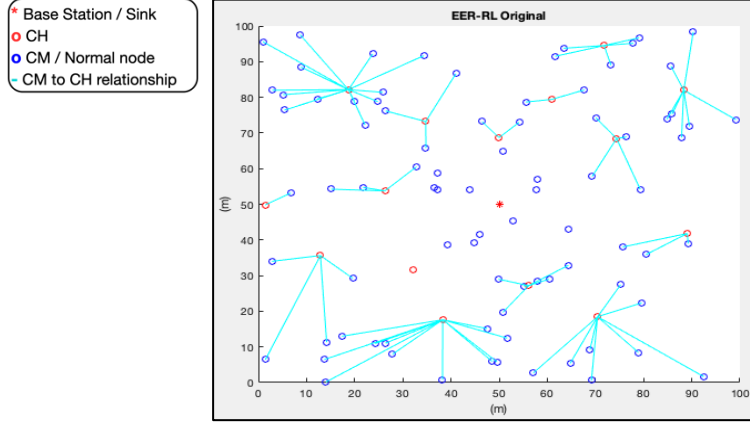
requiring higher or different CH percentages (other than 10%) where it can dynamically adjust the threshold distance. Further investigation is needed to refine the formula and measure its effectiveness under various network conditions.

b. 15% CHs of total nodes number:

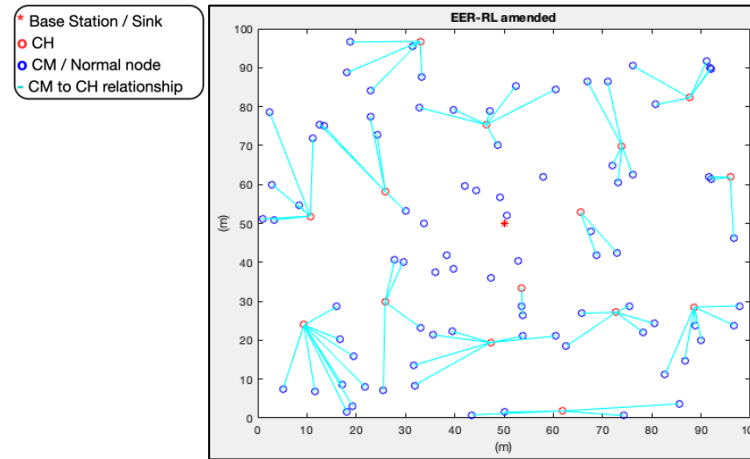
To further evaluate the dynamic CH $D_{threshold}$ formula, we investigated its behavior under a different network configuration. We increased the CH percentage to 15% and used the same network for EER-RL, mEER-RL with fixed threshold, and mEER-RL with $D_{threshold}$ formula. The observations were as followed:

- EER-RL showed a lack of adaptability to the modified parameters (increased CH percentage and potentially different network size due to a varying number of nodes). This resulted in area size and CH number inconsistencies. Numerous test runs were necessary to avoid infinite loops caused by the fixed distance threshold. During re-election, the protocol struggled to find suitable CH candidates due to the fixed threshold limitations. Manual intervention was required to adjust the distance threshold to “15” and resolve the issue.
- mEER-RL with fixed distance threshold showed a better adaptability. In cases where no suitable CH candidates remained, it effectively skips the completion of election, keeping the existing cluster structure based on the current fixed threshold (equals to the transmission range “20”). This approach might lead to the formation of 1-2 fewer clusters during the first few re-clustering phases compared to the ideal number for a 15% CH percentage.

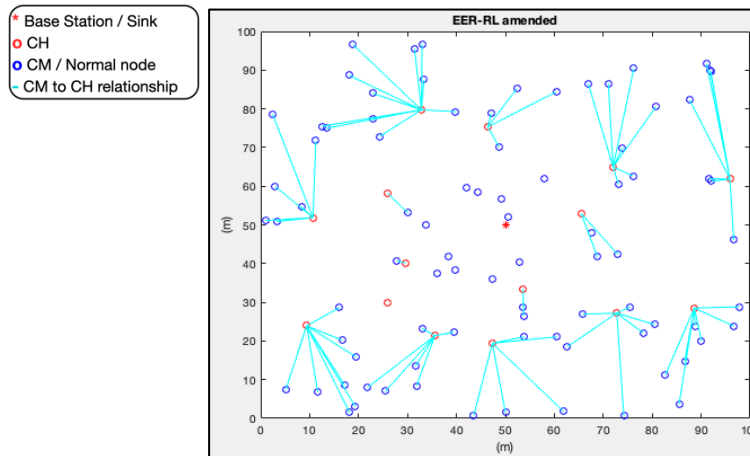
- The initial cluster arrangements differed between EER-RL and both mEER-RL versions due to the varying influences of the distance threshold calculation in each protocol, see Figure 24.



(a) EER-RL with manually set CH election parameters



(b) mEER-RL with fixed CH distance threshold



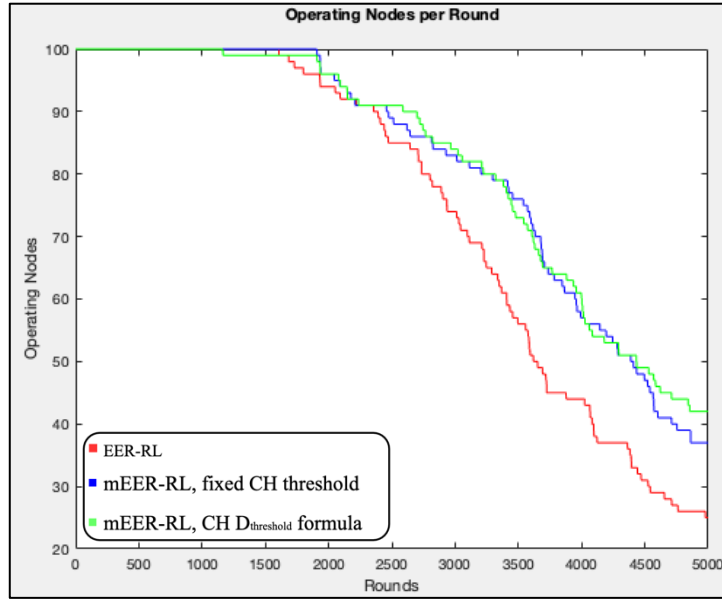
(c) mEER-RL with $D_{threshold}$ formula

Figure 24 mEER-RL performance with fixed and $D_{threshold}$ formula compared to original EER-RL, 5000 rounds and 15% CH

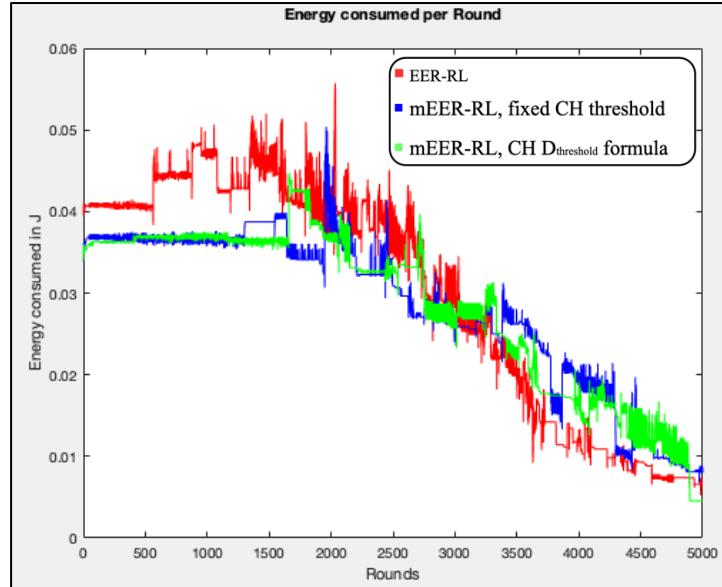
- mEER-RL with $D_{threshold}$ formula resulted with the lower FND at round 1165 but a greater number of alive nodes (42) at the end of the simulation compared to the other protocols. It also consumed the lowest maximum energy per round (0.0444 J). While mEER-RL with fixed CH distance threshold was able to keep all nodes alive until round 1909, both mEER-RL versions showed similar average energy consumption, average distance traveled, and total number of delivered packets. However, $D_{threshold}$ formula extended network lifetime. See Table 12 and Figure 25.

Table 12 Protocols performance for evaluating mEER-RL CH fixed and $D_{threshold}$ formula compared to original EER-RL, 5000 rounds and 15% CH

Evaluation metric	EER-RL	mEER-RL, fixed CH threshold	mEER-RL, $D_{threshold}$ formula
FND at round	1608	1909	1165
Alive nodes after 5000 rounds	25	37	42
Avg energy (J)	0.0302	0.0283	0.0284
Max energy (J)	0.0557	0.0503	0.0444
Avg distance (m)	1147.9	1087.8	1087.4
Packets delivered	360053	388927	387420



(a) Operating nodes per round



(b) Energy consumed per round

Figure 25 Operating nodes and energy consumed per round for evaluating mEER-RL CH fixed and $D_{threshold}$ formula compared to original EER-RL, 5000 rounds and 15% CH

- By comparing the total number of sent packets with the average distance traveled, EER-RL shows that its RL agent was unable to discover shorter paths compared to both mEER-RL versions. This resulted in higher energy consumption with fewer delivered data packets, leading to the lowest number of alive nodes at the end of the

simulation. Both versions of mEER-RL have achieved better routing, showing a noticeable balance between minimizing energy consumption and minimizing hop count distance, see Table 12.

- All protocols experienced some packet duplication and packet drops, as shown in Figure 26. This observation was made by manually comparing the number of alive nodes with the total number of packets sent during a single round.

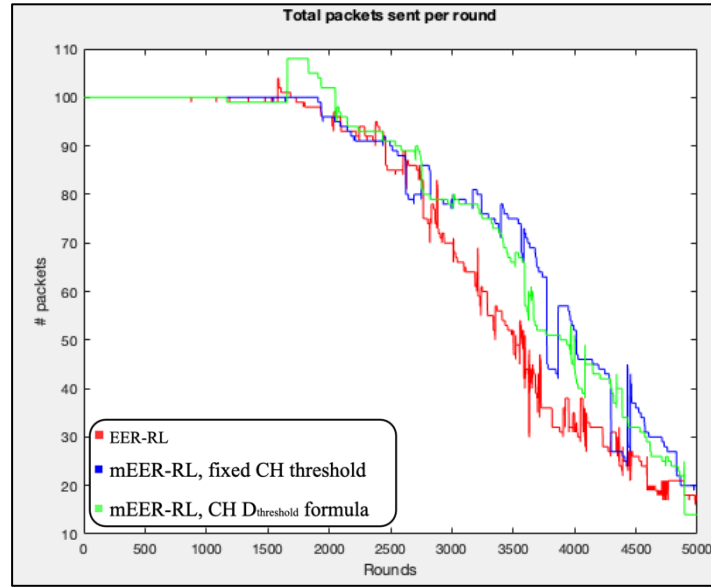


Figure 26 Total sent packets per round for evaluating mEER-RL CH fixed and $D_{threshold}$ formula compared to original EER-RL, 5000 rounds and 15% CH

This section shows the ability of $D_{threshold}$ formula to adapt to the network size, total number of nodes, and CH percentage. While the formula resulted in slightly lower performance compared to the fixed distance threshold for mEER-RL with total of 10% CHs, it offers potential benefits for different CH percentages and network parameters. Manually adjusting the threshold based on trial and error is no longer necessary. Further investigation is recommended to refine the formula and enhance its adaptability across diverse network configuration, including variations in CH percentage and network size for efficient clustering and routing. For

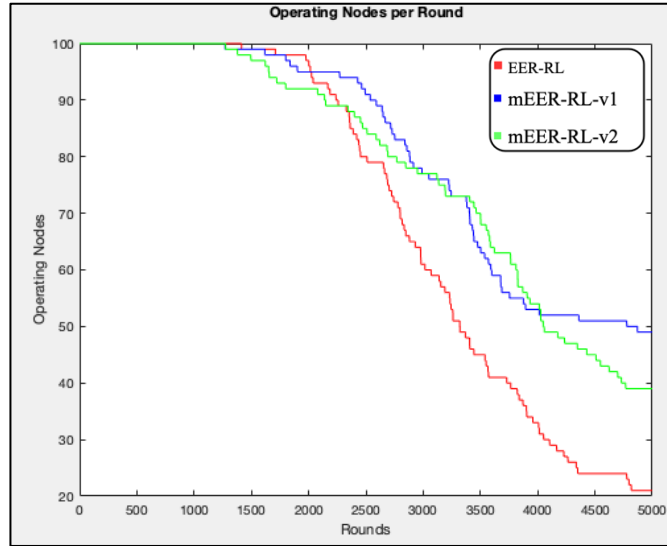
the following sections, we will maintain a fixed distance threshold with 10% CHs.

5.2.3 Mitigating Packet Drops and Duplications in mEER-RL-v2

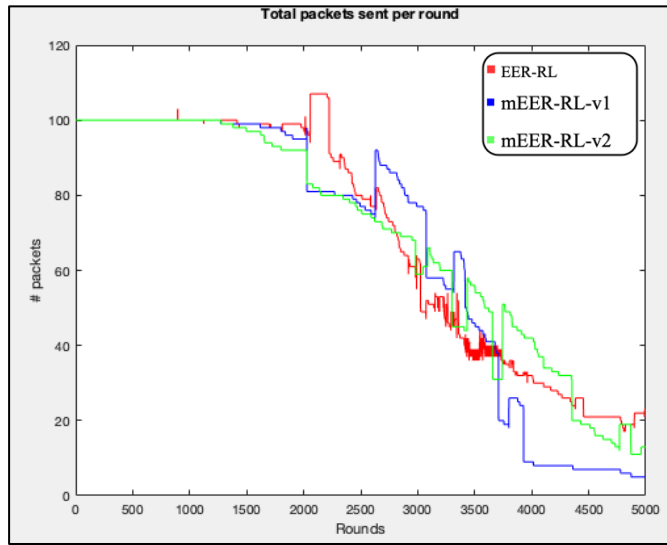
The goal of this section is to evaluate the effectiveness of modifications made to the re-clustering phase and the multi-hop scheme in mEER-RL-v2. These changes aim to minimize packet drops and duplications, as discussed in sections 4.4 and 4.5. We compare the performance of EER-RL and mEER-RL-v1, and the improved mEER-RL-v2 using the same simulated network.

In Figure 27, mEER-RL-v2 successfully transmits packets without duplication while maintaining good overall performance. However, there's still room for improvement in packet delivery efficiency by encouraging the RL agent to prioritize paths that guarantee delivery beside low-cost.

Notably, mEER-RL-v2 shows the lowest energy consumption, particularly within the initial 1000 rounds when all the nodes are alive (Figure 28 (a)). This behavior trend continuous even after the first node death, with mEER-RL-v2 showing steadily lower energy consumption despite potentially longer traveled distances (Figure 28 (b)).

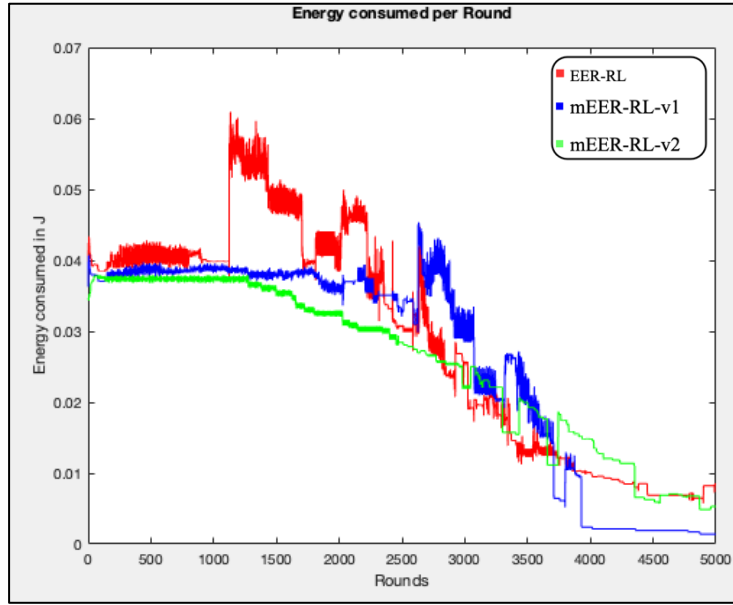


(a) Operating nodes per round

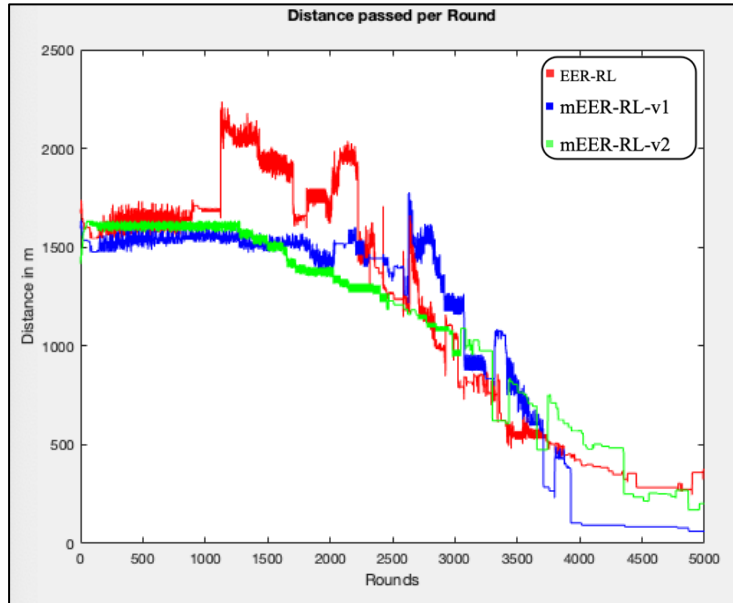


(b) Sent packets per round

Figure 27 Operating nodes and total sent packets per round for evaluating updated mEER-RL performance for Multihop advanced and packet duplication prevention, 5000 rounds



(a) Energy consumption per round



(b) Traveled distance per round

Figure 28 Energy consumption and traveled distance per round for evaluating updated mEER-RL performance for Multihop advanced and packet duplication prevention, 5000 rounds

The results in Table 13 confirms that mEER-RL-v2 achieves the lowest average and maximum energy consumption. The improved multi-hop mechanism reduces the energy required for packet transmission to the next hop. Additionally, the duplication prevention ensured only valid packets are considered in the comparison of total delivered packets. This is

as compared to EER-RL and mEER-RL-v1, which suffer from packet duplications and missing transmissions. The validity of delivered packets can be further verified by comparing the number of alive nodes in a round with the corresponding number of delivered packets. This creates a more accurate environment for the RL agent to learn optimal routing policies.

Table 13 Protocols performance for evaluating updated mEER-RL performance for Multihop advanced and packet duplication prevention, 5000 rounds

Evaluation metric	EER-RL	mEER-RL-v1	mEER-RL-v2
FND at round	1415	1271	1275
Alive nodes after 5000 rounds	21	49	39
Avg energy (J)	0.0289	0.0263	0.0255
Max energy (J)	0.0609	0.0452	0.0376
Avg distance (m)	1173.7	1055.9	1081.3
Packets delivered	348053	331257	345449

The improvements implemented in mEER-RL-v2 noticeably reduce packet drops and duplications, leading to lower energy consumption and potentially more reliable packet delivery. However, further investigation is needed to refine the RL agent's decision-making process to prioritize paths that ensure successful delivery in addition to minimizing transmission costs.

5.2.4 Modified Q-function and Reward: Balancing energy Efficiency and Reliability in mEER-RL

This section explores how mEER-RL-v3 achieves a balance between energy efficiency and reliable data delivery. We address the potential conflict between minimizing hop count with the reduction of energy consumption and guaranteeing successful packet transmission, in order to mitigate short-sighted routing decisions.

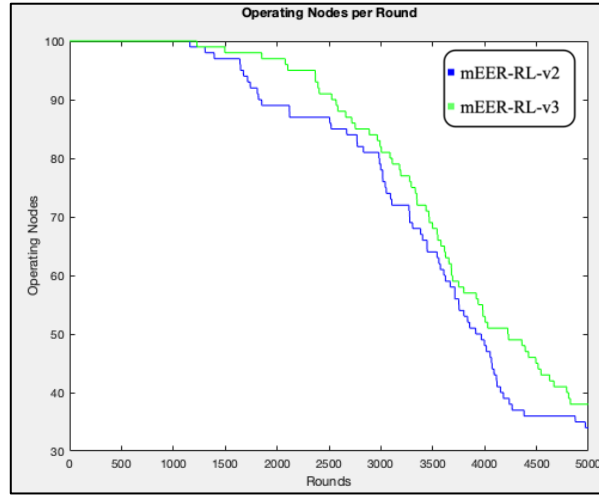
In previous versions (mEER-RL v1 and v2), the RL agent’s decision-making process might prioritize low cost paths that neglect packet delivery success. This could lead to dropped packets despite minimizing energy consumption. To address this, we introduced a modification to the Q-function and reward calculations (introduced in section 4.1.7 with Equations 4.5 and 4.6) by incorporating a constant to the hop count (constant = 7 found the best based on trial and error) and PDR metric. The probabilistic values were set as Table 14, where the new probabilistic parameter is added within the Q-value and reward functions for PDR. This empowers the RL agent to consider energy efficiency, hop count, and packet delivery success when selecting routes.

Table 14 Probabilistic values of mEER-RL Q-value and Reward functions including PDR

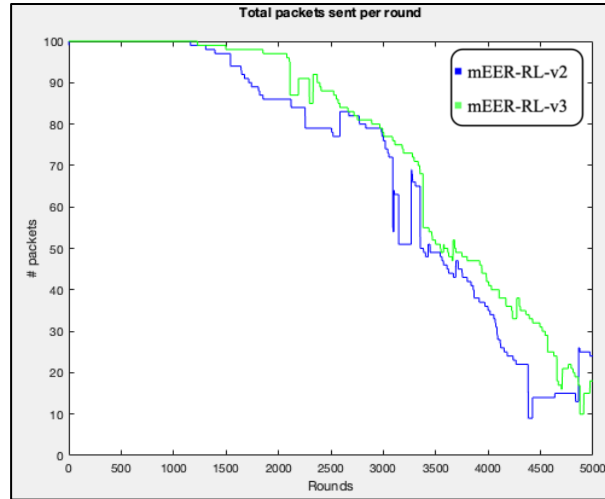
Parameter	Value
$p(\text{energy probabilistic})$	0.4
$q(\text{hops probabilistic})$	0.4
$\beta(\text{PDR probabilistic})$	0.2

In simpler terms, the agent is no longer only incentivized by shorter paths with potentially lower energy consumption. Instead, it learns to prioritize routes that guarantee successful data delivery to the sink, even if they require slightly more energy. The effectiveness of the modified Q-function and reward is shown in Figure 29 and Table 15.

Figure 29 illustrates a significant improvement in packet delivery success rate and higher number of alive nodes achieved by mEER-RL-v3 compared to v2. This confirms the effectiveness of incorporating PDR into the agent’s decision-making process by learning better routing policy.



(a) Operating nodes per round



(b) Sent packets per round

Figure 29 Operating nodes and total sent packets per round for evaluating mEER-RL-v3 for the new Q-function and reward performance, 5000 rounds

As expected, including PDR to the RL learning process leads to a slight increase in energy consumption, (both average and maximum) for mEER-RL-v3 shown in Table 15. This is a trade-off for achieving higher packet delivery success. However, the benefits extend beyond just successful transmissions. mEER-RL-v3 show a longer network lifetime, marked with a higher FND value and a greater number of alive nodes at the end of the simulation (5000 rounds). This indicates that the RL agent in mEER-RL-v3 effectively balances routing efficiency with the overall health and reliability of the network.

Table 15 Protocols performance results for evaluating mEER-RL-v3 performance for the new Q-function and reward, 5000 rounds

Evaluation metric	mEER-RL-v2	mEER-RL-v3
FND at round	1160	1228
Alive nodes after 5000 rounds	34	38
Avg energy (J)	0.0257	0.0281
Max energy (J)	0.0398	0.0427
Avg distance (m)	1096.8	1223
Packets delivered	345920	371700

The modifications implemented in mEER-RL-v3 reveals a successful approach to balancing energy efficiency and reliable data delivery. By incorporating PDR into the Q-function and reward calculations, the RL agent learns to prioritize routes that not only minimize energy consumption and hop count but also guarantee successful packet transmission. This trade-off results in a more robust and reliable routing protocol for IoT networks.

5.2.5 Dynamic Learning

In this section, we explore the concept of dynamic learning adaptation in mEER-RL-v4. By adjusting the learning rate α (alpha) throughout the simulation, we aim to achieve a better balance between exploration (discovering new routes) and exploitation (utilizing existing knowledge) for the RL agent.

a. Impact of the decreasing factor (ϵ):

We investigate the effect of the decreasing factor (ϵ) in Equation 4.1.4, which controls the rate at which alpha is reduced over time. Different (ϵ) values were used in separate simulations of the same network (Table 16).

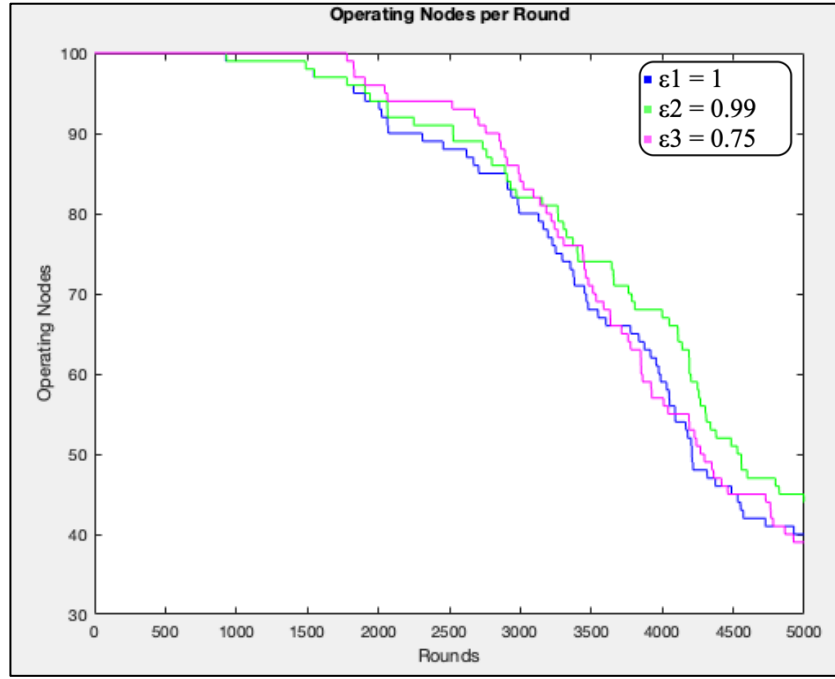
Table 16 ϵ values for dynamic learning simulation

Simulation	ϵ	ϵ value
1	ϵ_1	1
2	ϵ_2	0.99
3	ϵ_3	0.75

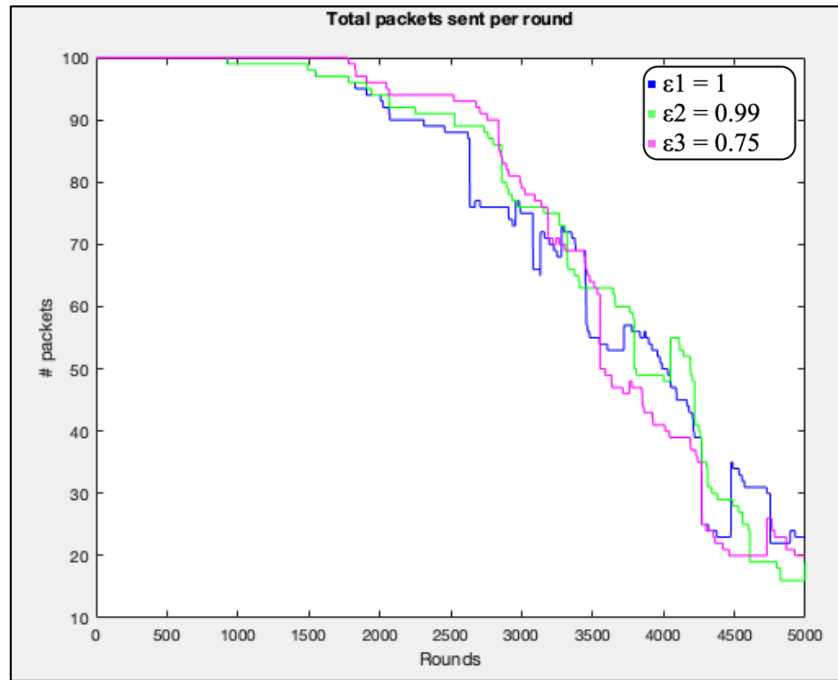
For ϵ fine-tuning in mEER-RL, a value of $\epsilon=1$ was initially used to allow for thorough exploration of the environment by the RL agent. While lower ϵ values lead to faster alpha decay, prioritizing exploitation in later stages. The results in Table 17 and Figure 30 demonstrate the influence of ϵ on network performance.

Table 17 Protocol performance results for evaluating mEER-RL using different ϵ values for dynamic learning effect

Evaluation metric	$\epsilon = 1$	$\epsilon = 0.99$	$\epsilon = 0.75$
FND at round	926	930	1779
Alive nodes after 5000 rounds	40	44	39
Avg energy (J)	0.0265	0.0258	0.0255
Max energy (J)	0.0364	0.0364	0.0356
Avg distance (m)	1149.8	1095.5	1082.6
Packets delivered	374933	380026	376620



(a) Operating nodes per round



(b) Sent packets per round

Figure 30 Operating nodes and total sent packets per round for evaluating mEER-RL using different ϵ values for dynamic learning effect

By observing FND and packet delivery, $\epsilon = 0.75$ (ϵ_3) yielded the best performance, achieving a significantly later FND at round 1779 compared to ϵ_1 (926) and ϵ_2 (930). This suggests that ϵ_3 effectively balanced

exploration and exploitation, enabling the RL agent to discover efficient routes and maintain network health for a longer duration. The number of delivered packets also remained high until FND for ϵ_3 , indicating successful data transmission.

Also, we compare the energy consumption and distance traveled by each of them. Before reaching FND, ϵ_3 led to the lowest energy consumption (average and maximum) and shortest average traveled distances (Figure 31 and Table 17). This highlights the benefit of ϵ_3 in promoting efficient exploration that leads to the discovery of energy-saving routes.

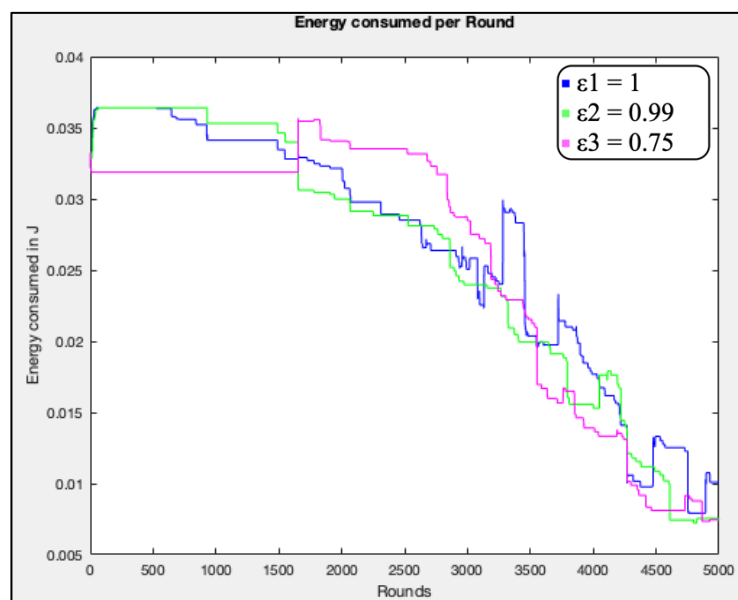


Figure 31 Energy consumption per round for evaluating mEER-RL using different ϵ values for dynamic learning effect

b. Resetting α strategies:

Based on the findings above, lower ϵ values (like ϵ_3) seem to achieve a good balance between exploration and exploitation while all nodes are alive. However, after FND, exploration becomes less effective due to the already reduced alpha value. New routes might be missed, especially after topology changes caused by re-clustering and dead nodes. To address this,

we examine the alpha reset strategies introduced in Section 4.1.6 and evaluate their impact.

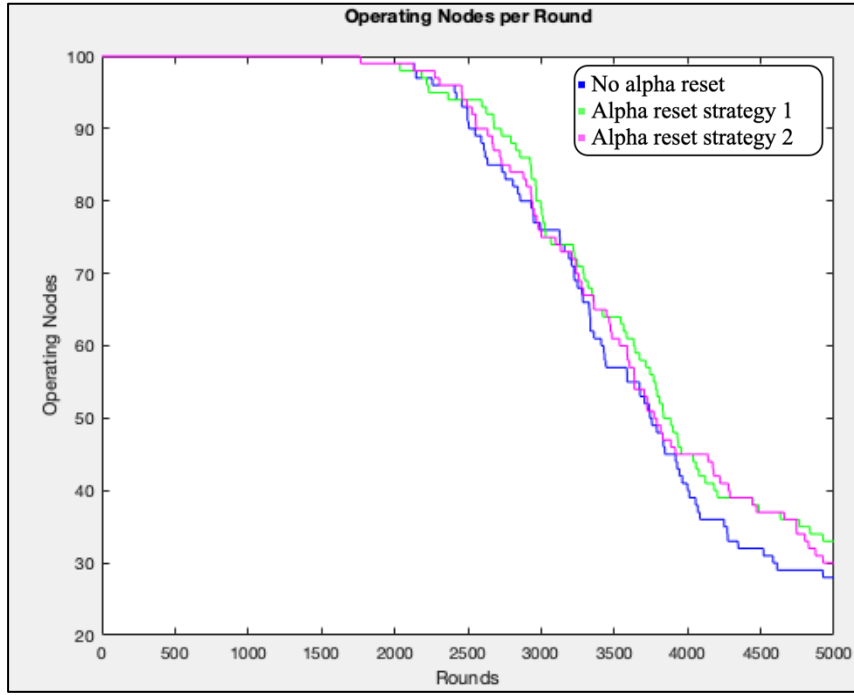
For performance comparison during the simulation, we ran the below mEER-RL versions on the same network setup:

- mEER-RL v4.1 (No Reset): This version includes dynamic learning with $\varepsilon = 0.75$ but no alpha reset.
- mEER-RL v4.2 (Reset Strategy 1): This version employs dynamic learning ($\varepsilon = 0.75$) with alpha reset to the original value after each re-clustering phase.
- mEER-RL v4.3 (Reset Strategy 2): This version uses dynamic learning ($\varepsilon = 0.75$) with alpha reset to the original value only when dead nodes are detected in a round.

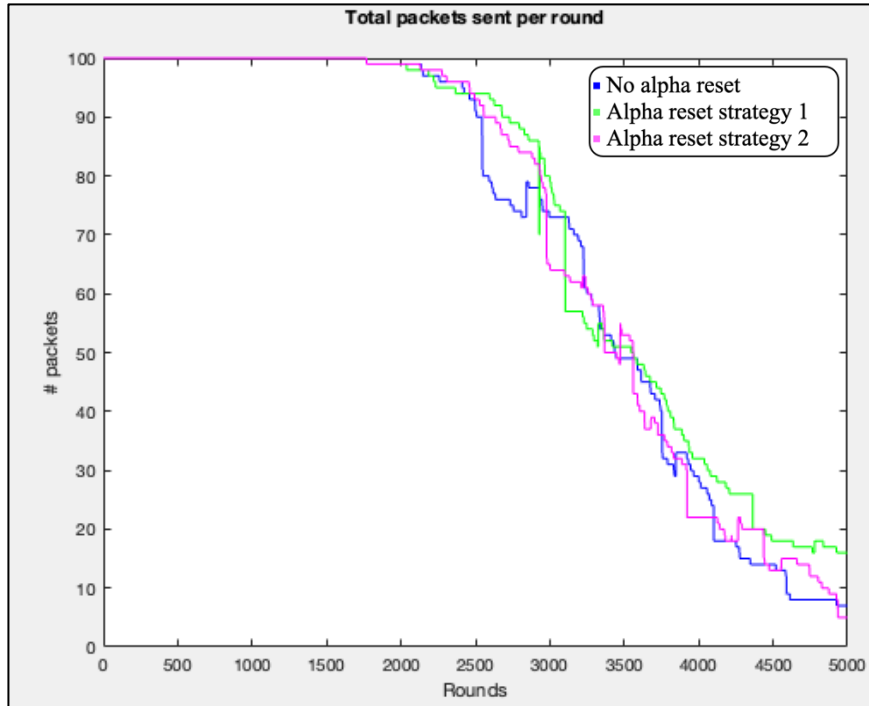
The results show that alpha reset strategy 1 (reset after re-clustering) maintained the highest number of alive nodes and delivered the greatest number of packets throughout the simulation (Table 18 and Figure 32). This indicates that resetting alpha after re-clustering allows the RL agent to adapt to significant network topology changes and discover new efficient routes.

Table 18 Protocols performance results for evaluating mEER-RL Alpha reset strategies for dynamic learning effect

Evaluation metric	No alpha reset strategy	Alpha reset strategy 1	Alpha reset strategy 2
FND at round	1768	1768	1768
Alive nodes after 5000 rounds	28	33	30
Avg energy (J)	0.0255	0.0262	0.0258
Max energy (J)	0.0383	0.0397	0.0388
Avg distance (m)	1124.3	1138.7	1125
Packets delivered	350669	363809	353682



(a) Operating nodes per round



(b) Sent packets per round

Figure 32 Operating nodes and total sent packets per round for evaluating mEER-RL Alpha reset strategies for dynamic learning effect

For the energy consumption, while strategy 1 resulted in slightly higher energy consumption compared to the others (Figure 33 and Table 18), this is likely due to the higher PDR achieved. This trade-off is

considered positive as it denotes successful data transmission despite the slight increase in energy usage. Additionally, the higher energy consumption did not lead to a shorter network lifetime because the RL agent effectively adapted to new routes after re-clustering.

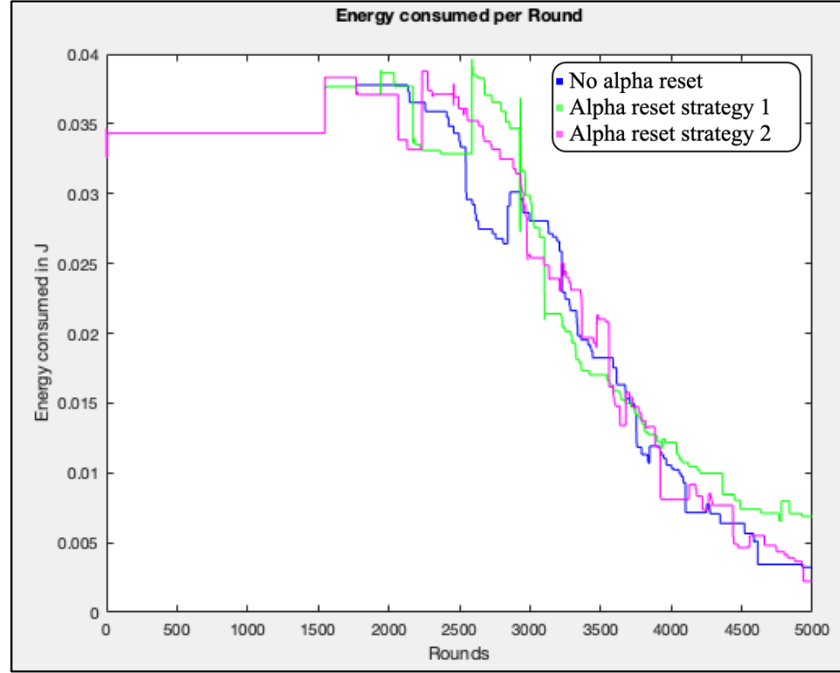


Figure 33 Energy consumption per round for evaluating mEER-RL Alpha reset strategies for dynamic learning effect

Interestingly, the results indicate that strategy 2 (reset only on dead nodes) might not be necessary. Since re-clustering inherently involves significant topology changes, resetting alpha after re-clustering seems to be sufficient for effective adaptation. Dead nodes primarily affect intra-cluster communication, resetting alpha after each dead node might be less impactful compared to the broader changes introduced during re-clustering (impacting both intra and inter-cluster communication).

We explored dynamic learning rate adaptation in mEER-RL. We observed that a lower ϵ value (e.g., $\epsilon = 0.75$) promotes a balance between exploration and exploitation, leading to efficient route discovery and improved network performance. Furthermore, resetting the learning rate after re-clustering phases (mEER-RL v4.2) proved to be an effective

strategy for adapting to network topology changes and maintaining high performance throughout the simulation. Based on these findings, mEER-RL-v4.2 is considered the most suitable choice for deployment as the production routing protocol for IoT networks with fixed nodes. This version effectively balances exploration and exploitation, adapts to network changes, and achieves efficient and reliable data transmission.

5.2.6 Mobile mEER-RL

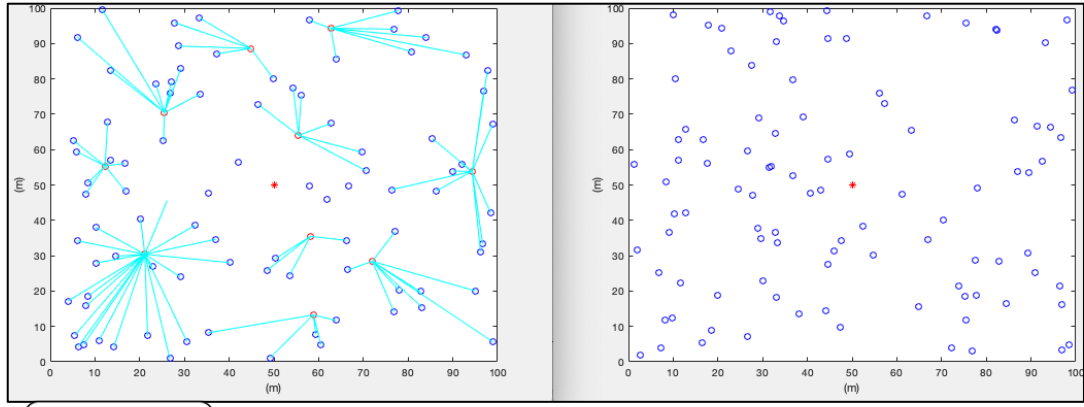
This section explores the deployment of mEER-RL in Wireless based IoT networks with mobile nodes. As discussed in Section 4.1.8, the mobility model involves nodes moving randomly in speed and direction at the end of each round. The protocol triggers location changes to evaluate re-clustering needs alongside the CH threshold energy condition.

Mobile networks are naturally more dynamic than fixed networks due to constant node movement. This can lead to frequent topology changes, potentially requiring more alpha resets compared to the fixed network scenario. Here, we investigate suitable dynamic learning parameter configurations for the mobile version of mEER-RL (mobile mEER-RL).

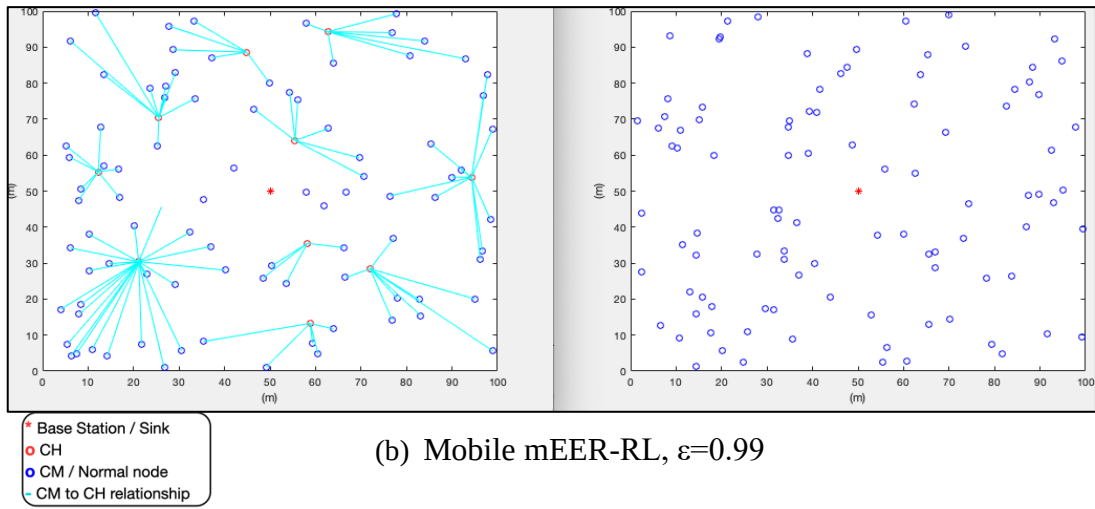
a. Impact of the decreasing factor (ϵ):

Similar to the fixed network case, we first evaluate the effect of the decreasing factor (ϵ) on learning performance before applying alpha reset strategies. We use the same ϵ values from Table 16 ($\epsilon = 1, 0.99$, and 0.75).

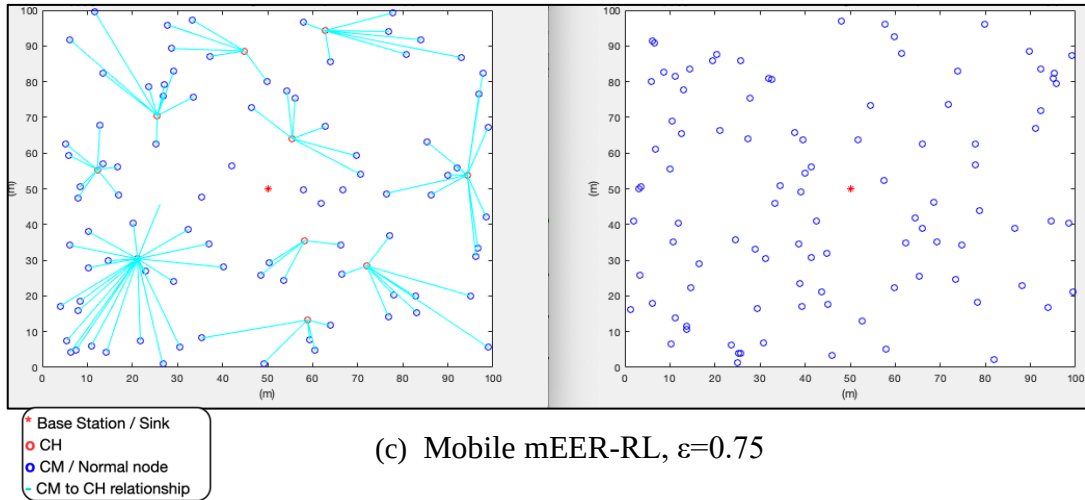
To visualize node mobility, Figure 34 confirms that nodes are moving randomly within the network area across all ϵ values, by showing the first and last round nodes location. This validates the mobility model.



(a) Mobile mEER-RL, $\epsilon=1$



(b) Mobile mEER-RL, $\epsilon=0.99$

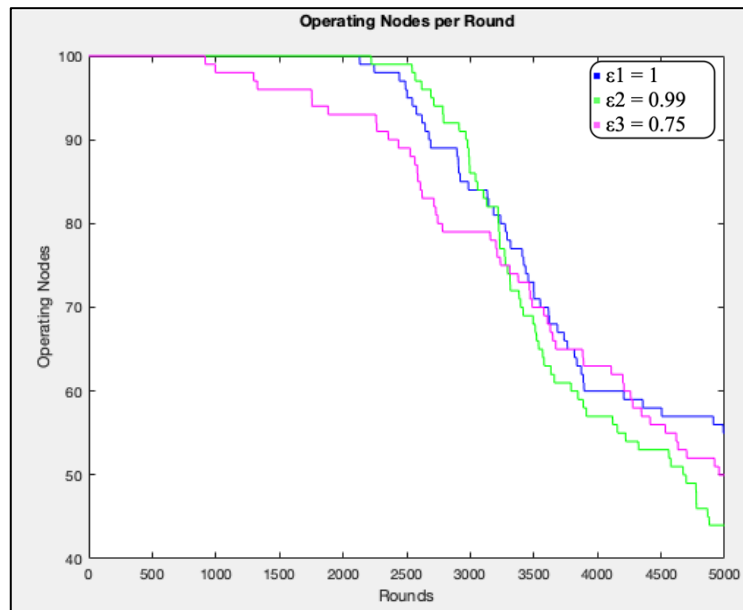


(c) Mobile mEER-RL, $\epsilon=0.75$

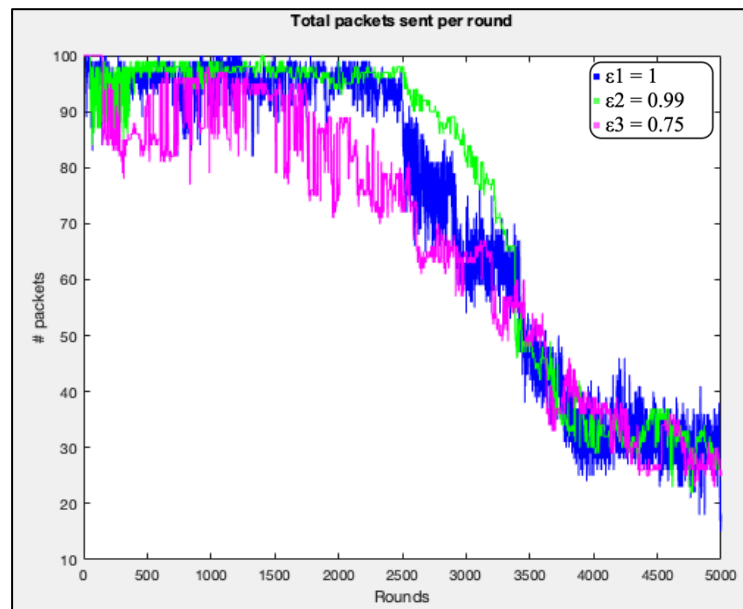
Figure 34 First and last round nodes location for mobile mEER-RL, with different ϵ values

The effect of mobility on learning is shown in Figure 35. $\epsilon = 0.99$ appeared as the most suitable decreasing factor due to the uncertainty in

mobile networks. Both $\varepsilon = 1$ and $\varepsilon = 0.75$ presented greater instability in packet delivery success compared to $\varepsilon = 0.99$. While $\varepsilon = 0.99$ achieved a more stable packet delivery rate, it also resulted in a lower number of alive nodes at the end of the simulation. In mobile networks, routes change frequently, requiring the RL agent to adapt to the randomness of topology changes and cluster requirements.



(a) Operating nodes per round



(b) Sent packets per round

Figure 35 Operating nodes and total sent packets per round for evaluating mobile mEER-RL using different ε values for dynamic learning effect

Energy consumption behavior in Figure 36 aligns with the number of alive nodes. When $\varepsilon = 0.99$ is used for decreasing alpha, the energy consumption is more stable. This indicates that with a decreased alpha, the RL agent prioritizes learned knowledge (existing routes) over exploration (discovering new routes), especially after the initial period of frequent topology changes.

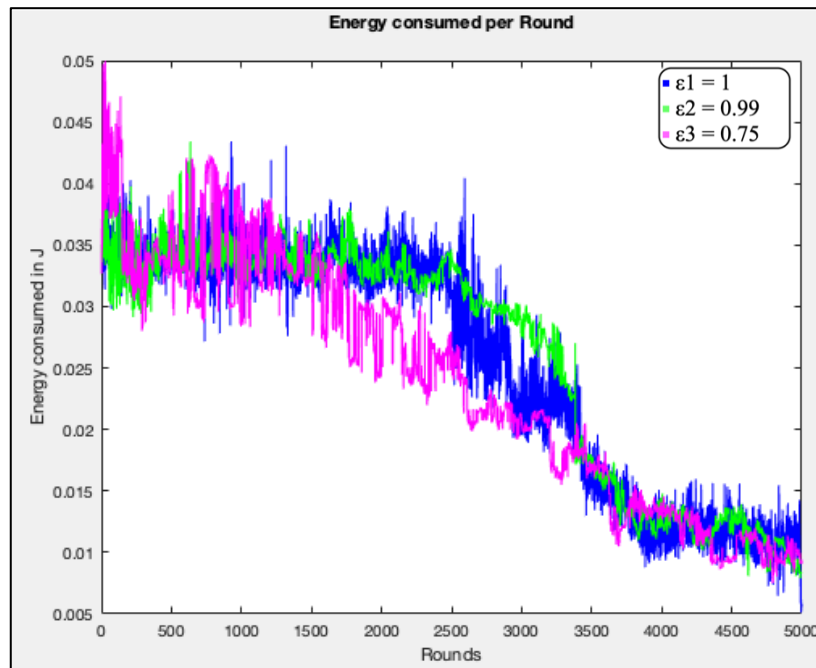


Figure 36 Energy consumption per round for evaluating mobile mEER-RL using different ε values for dynamic learning effect

The detailed results in Table 19 confirm these observations. $\varepsilon = 0.99$ achieves a later FND and higher delivered packets compared to other values. However, it also has a lower number of alive nodes and a higher maximum energy per round. These results indicate efficient energy consumption for a higher packet delivery success rate, despite the challenges of comparing performance in a mobile environment.

Table 19 Protocol performance results for evaluating mobile mEER-RL using different ε values for dynamic learning effect

Evaluation metric	$\varepsilon = 1$	$\varepsilon = 0.99$	$\varepsilon = 0.75$
FND at round	2131	2222	917
Alive nodes after 5000 rounds	55	44	50
Avg energy (J)	0.0254	0.0261	0.0239
Max energy (J)	0.0434	0.0434	0.0497
Avg distance (m)	1076.5	1127.5	1002
Packets delivered	358177	372071	332307

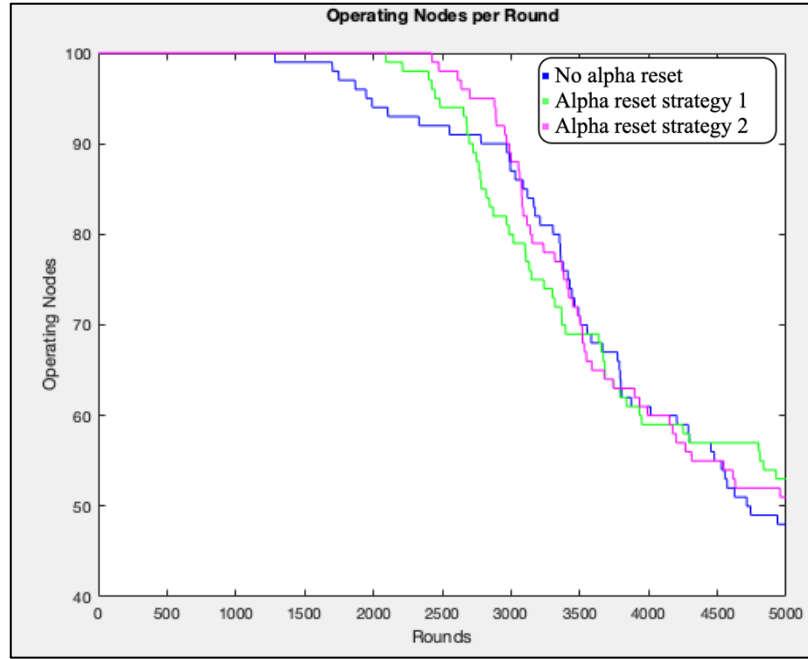
b. Resetting α strategies:

Based on the findings above, $\varepsilon = 0.99$ is considered the most suitable value for the decreasing factor in mobile mEER-RL. We now evaluate different alpha reset strategies similar to the fixed network case:

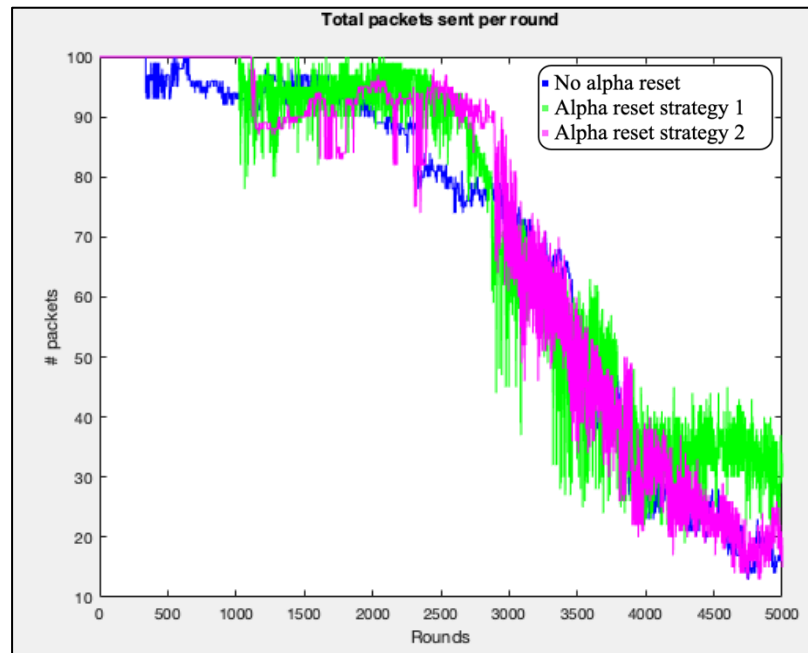
- Mobile mEER-RL v1 (No Reset): This version includes dynamic learning with $\varepsilon = 0.99$ but no alpha reset.
- Mobile mEER-RL v2 (Reset Strategy 1): This version employs dynamic learning ($\varepsilon = 0.99$) with alpha reset to the original value after each re-clustering phase.
- Mobile mEER-RL v3 (Reset Strategy 2): This version uses dynamic learning ($\varepsilon = 0.99$) with alpha reset to the original value only when dead nodes are detected in a round.

Both Mobile mEER-RL v2 and v3 (with alpha reset strategies) achieved a later FND, higher number of alive nodes by the end of the simulation, and greater packet delivery success compared to Mobile mEER-RL v1 (no reset) (Figure 37 and

Table 20).



(a) Operating nodes per round



(b) Sent packets per round

Figure 37 Operating nodes and total sent packets per round for evaluating mobile mEER-RL Alpha reset strategies for dynamic learning effect

Table 20 Protocols performance results for evaluating mEER-RL Alpha reset strategies for dynamic learning effect

Evaluation metric	No alpha reset strategy	Alpha reset strategy 1	Alpha reset strategy 2
FND at round	1285	2091	2427
Alive nodes after 5000 rounds	48	53	51
Avg energy (J)	0.0246	0.0262	0.0255
Max energy (J)	0.0453	0.0528	0.0529
Avg distance (m)	1056.6	1128.2	1101.3
Packets delivered	346904	366901	355124

The energy consumption for v2 and v3 is understandably higher than v1 due to their higher packet delivery success rate (Figure 38 and

Table 20). Additionally, the presence of alpha reset strategies introduces some instability in energy consumption as the RL agent explores for potentially better routes during network topology changes.

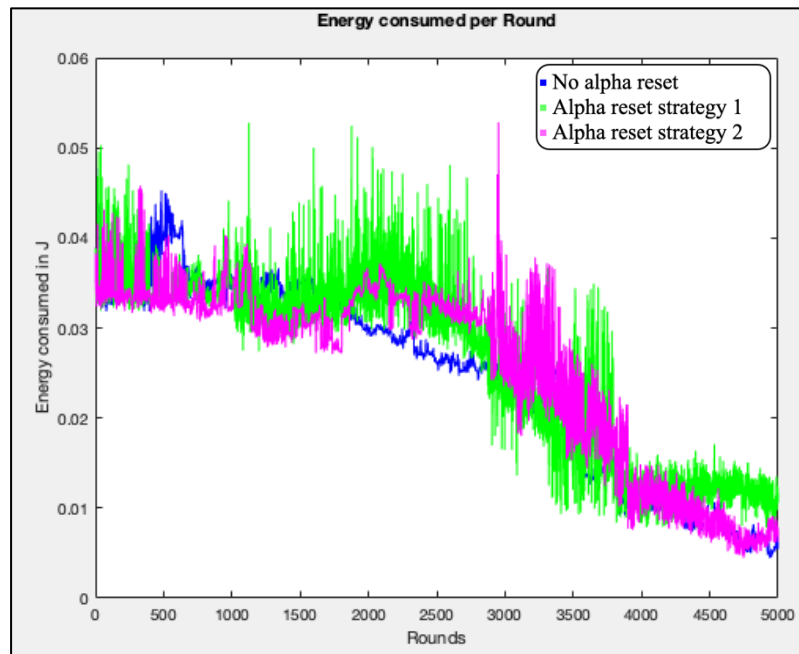


Figure 38 Energy consumption per round for evaluating mobile mEER-RL Alpha reset strategies for dynamic learning effect

These results suggest that mobile mEER-RL benefits from alpha reset strategies (v2 or v3) to handle the dynamic nature of mobile networks. However, the specific choice between reset strategies 1 and 2 might require further investigation depending on the trade-off between exploration cost (energy consumption) and achieving higher packet delivery success rates. Overall, mobile mEER-RL with $\varepsilon = 0.99$ and alpha reset strategy 1 is a promising approach for routing in mobile IoT networks.

5.3 Performance Comparison: Fixed vs. Mobile mEER-RL and Existing Protocols

This section analyzes the performance of mEER-RL (both fixed and mobile versions) for the wireless IoT networks compared to existing routing protocols, LEACH and EER-RL. Here, we evaluate the final versions of the protocols using the established performance metrics discussed earlier in the previous chapter.

5.3.1 RL-based Protocol Parameters

Table 21 summarizes the key parameters used for the RL-based protocols (EER-RL and mEER-RL) in the simulations. LEACH, being a non-RL protocol, does not have these parameters. The table highlights the following:

- Decreasing Factor (ε): The value that controls the learning rate decay over time. mEER-RL uses different ε values for fixed (0.75) and mobile networks (0.99) due to their varying levels of dynamism.
- Reward Function Parameters: The reward function has been modified for mEER-RL to influence the RL agent's behavior. These details are already discussed in section 4.1.7, which focuses on energy consumption, hop count (distance based on transmission range), and packet delivery success.

Table 21 Simulation parameters for RL based protocols

Parameter	EER-RL	mEER-RL	Mobile mEER-RL
α (<i>learning rate</i>)	1	1	1
ε (<i>decreasing factor</i>)	N/A	0.75	0.99
γ (<i>discount factor</i>)	0.95	0.95	0.95
p (<i>energy probabilistic</i>)	0.5	0.4	0.4
q (<i>hops probabilistic</i>)	0.5	0.4	0.4
β (<i>Packet delivery probabilistic</i>)	N/A	0.2	0.2

5.3.2 Network Setup and Clustering

This section examines how each protocol forms clusters during the network setup phase. Figure 39 illustrates the resulting clusters for each protocol.

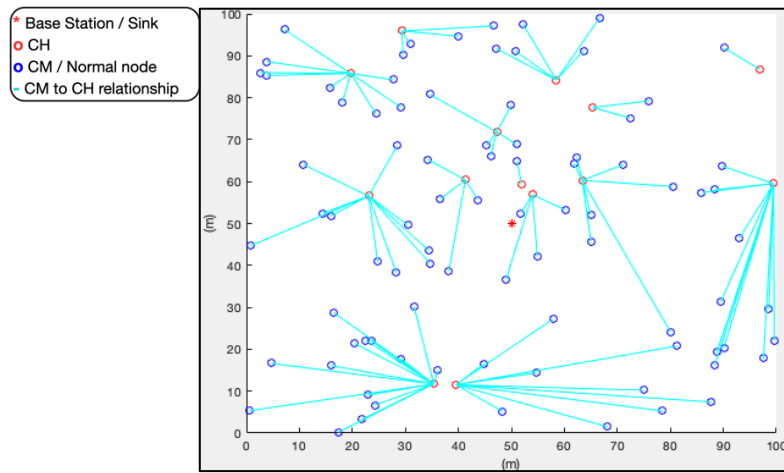
- **LEACH:** LEACH exhibits unbalanced cluster distribution. Most CHs are located close together in one network area (upper area portion of Figure 39 (a)), keeping the lower portion with 2 clusters only. Additionally, LEACH may create more or less clusters than required due to the random nature of its CH election process.
- **EER-RL:** EER-RL demonstrates well-balanced clusters (Figure 39 (b)) that change based on predefined CH energy thresholds. It forms 10 clusters based on the total number of nodes.
- **mEER-RL (Fixed and Mobile):** Both versions of mEER-RL initially form the same clusters (Figure 39 (c)) due to identical initial node locations and Q-values. Re-clustering occurs based on CH energy thresholds, with an additional location trigger for mobile mEER-RL. The initial number of clusters is 8 based on the number of available nodes for clustering (84 nodes), reflecting the efficiency of the mEER-RL CH election method considering the number of NS nodes (16 node).

However, the number of the included nodes in clustering adjusts dynamically throughout the network lifetime as dead nodes increase and node movement (mobile mEER-RL) affects the number of CMs. Mobile mEER-RL will adapt the number of clusters as nodes move closer to or further from the BS, following the CH election method.

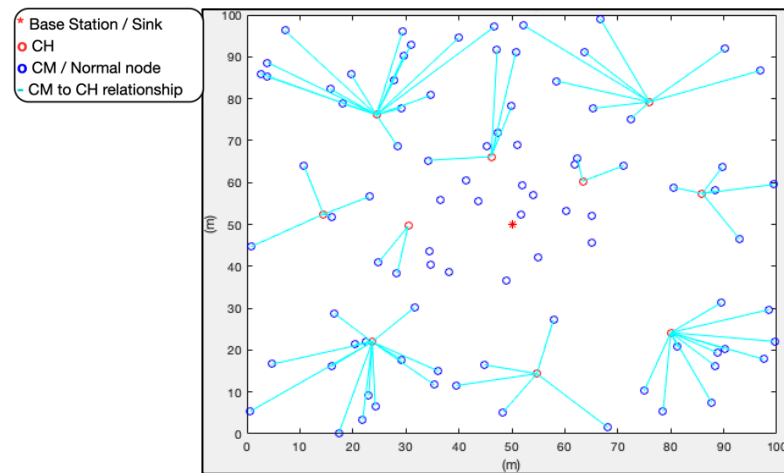
The comparison highlights the advantages of mEER-RL over LEACH and EER-RL:

- **Balanced Clustering:** mEER-RL ensures balanced cluster distribution, unlike LEACH's uneven cluster formation. While also excluding the NS nodes from CH election and clustering calculations unlike EER-RL.
- **Dynamic Re-clustering:** mEER-RL's re-clustering based on CH energy thresholds and nodes location changes (mobile mEER-RL) allows for network adaptation compared to EER-RL's reliance solely on energy thresholds, and LEACH randomly changing each round.

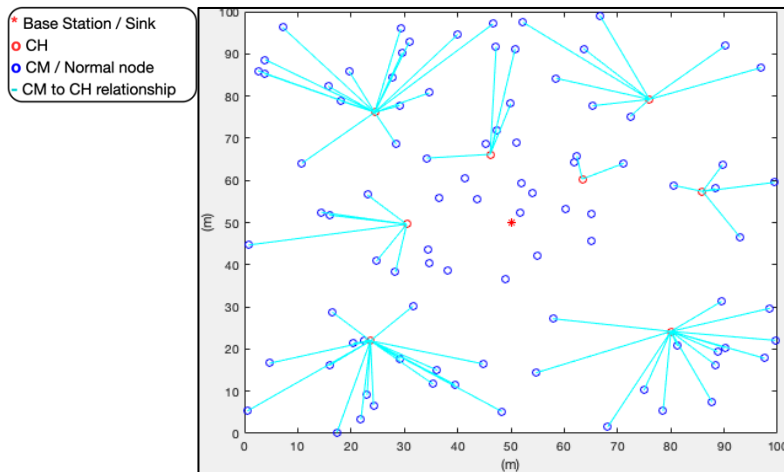
The trade-off between fixed and mobile mEER-RL lies in the additional overhead of managing node movement in the mobile version. However, the ability to adapt to network changes through location-triggered re-clustering can be crucial for mobile IoT networks.



(a) LEACH



(b) EER-RL



(c) Fixed and mobile mEER-RL

Figure 39 Formed clusters by each protocol in network setup phase, for fixed and mobile mEER-RL performance comparison with existing protocols

5.3.3 Network Lifetime and Energy Consumption

Evaluation

This section analyzes how each protocol influences network lifetime and energy consumption. Since these factors are complicatedly linked together, minimizing energy consumption is crucial for extending network lifetime in wireless IoT networks and WSNs due to the challenges in replacing or recharging batteries in deployed nodes.

Figure 40 compares the number of operational nodes per round across the protocols. Both fixed and mobile mEER-RL outperform LEACH and EER-RL by maintaining a higher number of alive nodes throughout the simulation. This suggests that the RL agent in mEER-RL learns to optimize the policy by selecting routes that minimize energy consumption and promote network longevity.

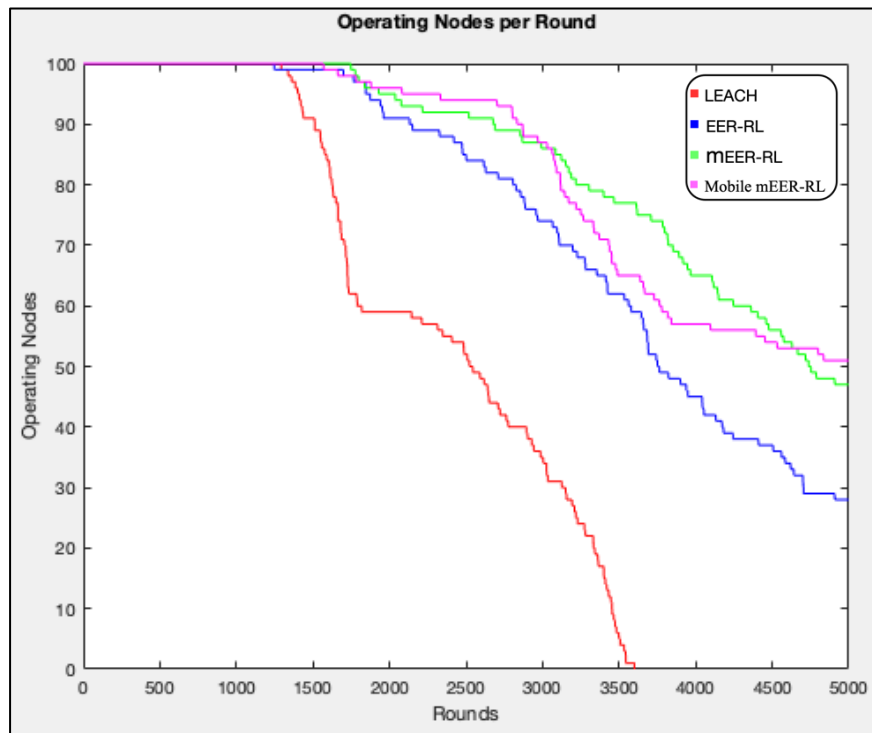
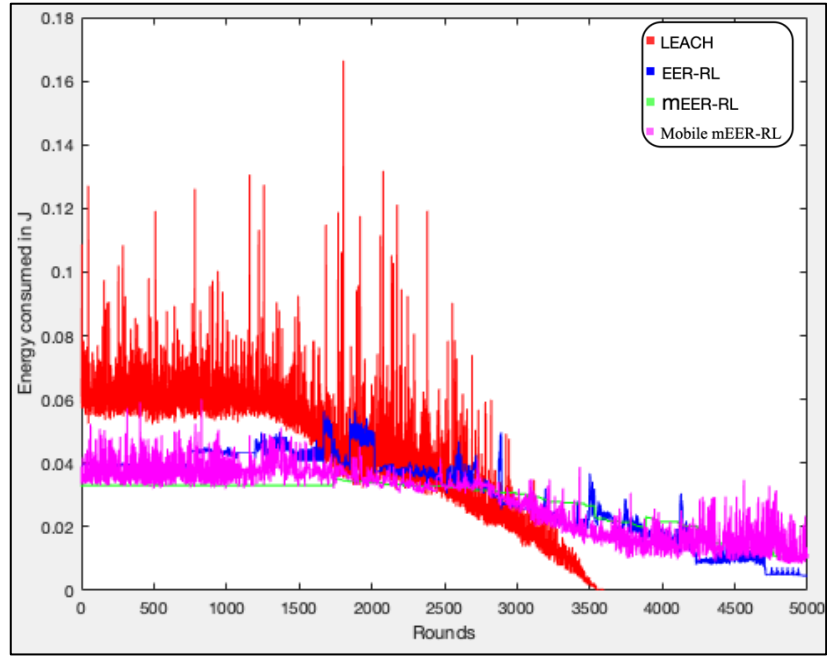


Figure 40 Operating nodes per round for fixed and mobile mEER-RL performance comparison with existing protocols

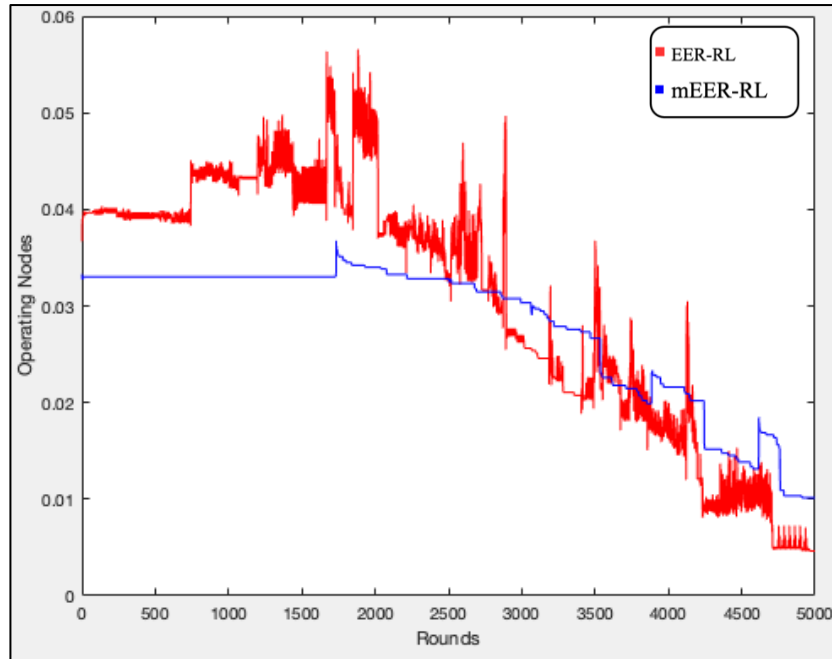
Figure 41 (a) shows the energy consumption behavior for each protocol:

- LEACH: LEACH shows the highest energy consumption per round, resulting in the shortest network lifetime (as observed earlier).
- EER-RL: EER-RL shows the energy consumption is unstable and fluctuates considerably.
- Fixed mEER-RL: Fixed mEER-RL demonstrates the most stable energy consumption pattern. This indicates the effectiveness of the RL agent's learning process with the modified reward function. The balance between exploration (trying new routes) and exploitation (utilizing learned efficient routes) contributes to this stability. Additionally, the figure indicates that the methods within the protocol efficiently manage network communication while updating the RL agent with relevant information for learning.
- Mobile mEER-RL: Mobile mEER-RL exhibits some instability in energy consumption due to the dynamic nature of mobile networks, where routes change frequently. The RL agent needs to explore and learn new routes as network conditions evolve.

Figure 41 (b) provides a closer look at the energy consumption comparison between EER-RL and fixed mEER-RL. Here, we see a significant difference, where fixed mEER-RL displays a steadier and lower energy consumption pattern, especially during the initial 1500 rounds, compared to the fluctuations in EER-RL behavior.



(a) All protocols of the simulation



(b) EER-RL vs mEER-RL

Figure 41 Energy consumption per round for fixed and mobile mEER-RL performance comparison with existing protocols

Table 22 summarizes the network lifetime and energy consumption metrics for all protocols, with 1 more metric added: network lifetime end, it is the estimated round at which the network becomes inoperable due to insufficient nodes.

However, mEER-RL outperforms other protocols in all these metrics:

- **FND and Alive Nodes:** Both fixed and mobile mEER-RL exhibit a later FND and a significantly higher number of alive nodes at the simulation's end (almost 50% of total nodes) with 34.42% improvement compared to LEACH and 40.19% compared to EER-R. This confirms their effectiveness in extending network lifetime.
- **Network Lifetime End:** This is mainly detecting the last round a protocol keeps the network operational until all nodes die, LEACH is the only protocol that failed to keep the network operational up to the end of simulation. mEER-RL is able to increase the number of surviving nodes by 67.86% over EER-RL.
- **Avg Energy:** mEER-RL protocols achieve the lowest average energy consumption, directly contributing to extended network lifetime. Fixed mEER-RL has the lowest overall average energy consumption.
- **Max Energy:** Fixed mEER-RL also demonstrates the lowest maximum energy consumption per round. This indicates that the RL agent successfully identified energy-efficient routes towards the BS. The higher maximum energy consumption in mobile mEER-RL is likely due to the exploration overhead caused by route changes.

Table 22 Network lifetime and energy evaluation metrics for fixed and mobile mEER-RL performance comparison with existing protocols

Evaluation metric	LEACH	EER-RL	Fixed mEER-RL	Mobile mEER-RL
FND at round	1294	1247	1749	1570
Alive nodes after 5000 rounds	0	28	47	51
Network lifetime end	3621	5000+	5000+	5000+
Avg energy (J)	0.0442	0.0305	0.0279	0.0286
Max energy (J)	0.1664	0.056	0.0367	0.0601

These results declare that mEER-RL's RL agent effectively adjusts its policy to minimize energy consumption and maximize network lifetime. The trade-off between fixed and mobile mEER-RL lies in the slightly higher energy consumption of the mobile version due to route adaptation.

The success of mEER-RL in network lifetime and energy efficiency can be attributed to several factors:

- **Energy-Aware Reward Function:** The reward function in mEER-RL likely incorporates energy consumption as a crucial factor, encouraging the RL agent to choose energy-efficient routes.
- **Dynamic Re-clustering:** Re-clustering based on energy thresholds and location changes (mobile mEER-RL) helps distribute the network load and prevents specific nodes from being overworked, thereby extending their lifespan.
- **Learning and Adaptation:** Over time, the RL agent in mEER-RL learns from experience and refines its routing strategies to further optimize energy usage and network longevity.

While mobile mEER-RL experiences some overhead due to managing node movement, its ability to adapt to network changes can still lead to a longer network lifetime compared to fixed mEER-RL in mobile IoT networks deployments. However, the benefits it brings in mobile IoT network deployments are as following:

- **Increased learning opportunities:** As nodes move, they encounter new network conditions and routes. This provides the RL agent with more data and learning opportunities.
- **Improved routing policy:** By exploring these new routes, the RL agent can refine its routing policy and potentially discover even more energy-efficient paths for data transmission. This continuous learning helps mobile mEER-RL adapt to the dynamic network environment.

- Longer network lifetime: The ability to adapt and discover new, efficient routes can ultimately lead to a longer network lifetime in mobile IoT networks compared to fixed mEER-RL.

Therefore, despite the overhead, mobile mEER-RL remains a valuable choice for scenarios where network dynamics are a significant factor, such as in mobile IoT networks.

5.3.4 Distance Traveled Evaluation

This section analyzes the distance packets travel within the network using each protocol and its relation to energy consumption. Minimizing the distance traveled translates to lower energy consumption for packet transmissions.

Figure 42 compares the average distance traveled by packets per round across the protocols:

- LEACH: LEACH exhibits the highest average distance traveled by packets. This is because LEACH doesn't consider distance when forming clusters, leading to potentially long paths from Cluster CMs to CHs and then directly to the BS.
- EER-RL and mEER-RL: Both EER-RL and mEER-RL utilize multi-hop communication with data aggregation to reduce the number of transmissions and the distance requiring higher energy to transmit packets to a far destination. Additionally, the reward function in mEER-RL likely incorporates modified hop count function, encouraging the RL agent to discover shorter paths to the BS. This combined approach leads to shorter traveled distances compared to LEACH and EER-RL.
- Fixed vs. Mobile mEER-RL: Notably, fixed mEER-RL demonstrates the steadiest performance in terms of traveled distance. Mobile mEER-RL exhibits some fluctuation due to the dynamic nature of mobile IoT network, where node movement alters path lengths.

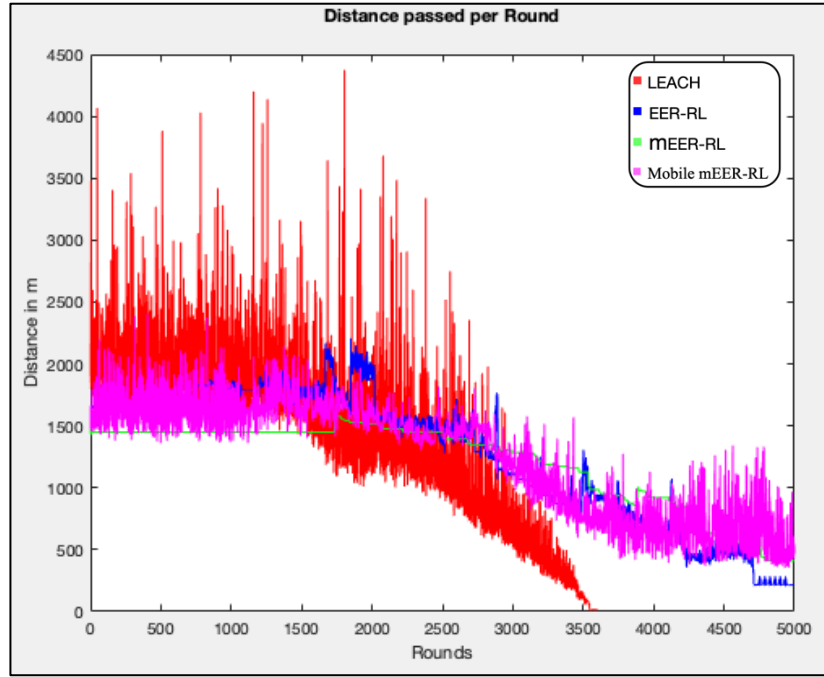


Figure 42 Traveled distance per round for fixed and mobile mEER-RL performance comparison with existing protocols

Table 23 explores the average distance traveled by packets per round for each protocol. These results describe the connection between traveled distance and energy consumption discussed earlier:

- LEACH: As expected, LEACH, with the highest average distance, also has the highest energy consumption (refer to previous sections).
- mEER-RL: mEER-RL achieves the lowest average distance and energy consumption 15.27% shorter distance than LEACH and 3.72% compared to EER-RL. This is demonstrating the effectiveness of its approach in finding shorter paths and conserving energy. This holds true even though mEER-RL has more alive nodes transmitting more packets.
- EER-RL and Mobile mEER-RL: Interestingly, EER-RL and mobile mEER-RL have comparable average traveled distances. This is a positive sign for the mobile version of mEER-RL, indicating its ability to learn shorter paths despite node movement and maintaining a high number of alive nodes (similar to fixed mEER-RL). This indicates that

exploration of new routes by the RL agent in mobile mEER-RL is highly beneficial.

Table 23 Traveled distance evaluation metric for fixed and mobile mEER-RL performance comparison with existing protocols

Evaluation metric	LEACH	EER-RL	Fixed mEER-RL	Mobile mEER-RL
Avg distance (m)	1431.3	1259.6	1212.7	1260.5

The results highlight the importance of minimizing traveled distance to achieve energy efficiency. mEER-RL is outperforming in this aspect due to the following:

- **Multi-hop Communication:** Utilizing multi-hop communication with data aggregation reduces the overall number of transmissions and traveled distance.
- **Hop Count in Reward Function:** The inclusion of hop count within the reward function incentivizes the RL agent to prioritize shorter paths for efficient packet delivery.

While mobile mEER-RL experiences some variations in traveled distance due to mobility, its ability to adapt and discover new paths through exploration helps maintain energy efficiency comparable to fixed mEER-RL. This makes mobile mEER-RL a suitable choice for scenarios where network dynamics are a significant factor.

5.3.5 Packet Delivery Evaluation

This section analyzes the effectiveness of each protocol in delivering packets to the BS. A high PDR is crucial for reliable data transmission in IoT networks.

Figure 43 compares the total number of packets delivered per round across the protocols:

- **LEACH:** LEACH shows a rapid decay in packet delivery after the FND. This is likely due to the high energy consumption in LEACH, leading to a reduced number of alive nodes with the ability to send packets.
- **EER-RL:** EER-RL shows evidence of packet duplication. The simulation setup specifies that each node should send only one packet per round. However, EER-RL experiences duplicate transmissions. This could be caused by nodes sending packets before re-clustering is complete, leading to outdated routing information for the RL agent.
- **mEER-RL:** mEER-RL addresses the duplication issue through the re-clustering and nodes communication priority schemes. It also utilizes the modified reward function for improved packet delivery performance.

Both fixed and mobile mEER-RL maintain a consistently high PDR throughout the simulation. However, mobile mEER-RL displays some instability due to changing network conditions and policy updates as the RL agent adapts to node movement.

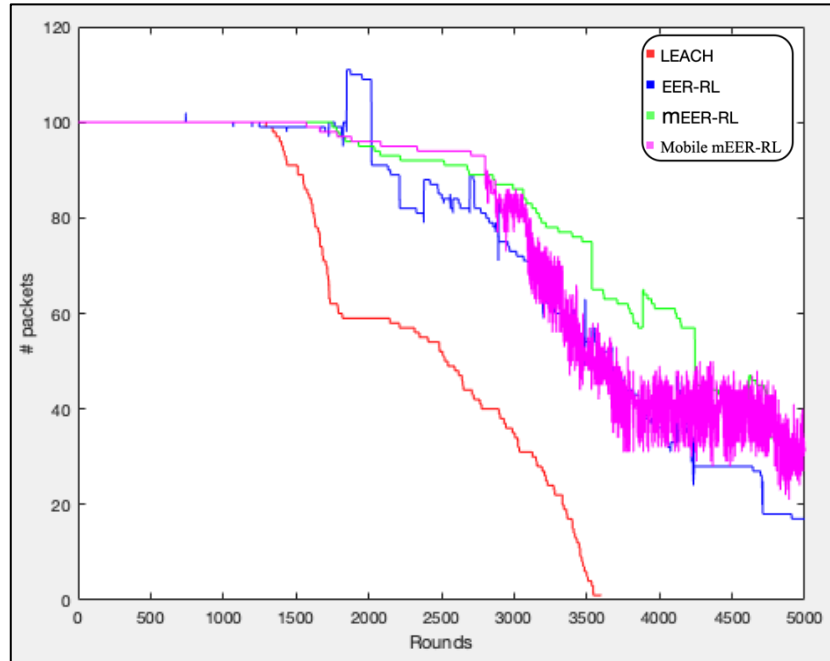


Figure 43 Total number of sent packets per round for fixed and mobile mEER-RL performance comparison with existing protocols

Table 24 presents the total number of packets sent by each protocol during the simulation. While not a direct measure of PDR (as some packets might be dropped), it provides insights into delivery success.

The total number of sent packets is the metric that reflects the overall number of packets transmitted by all nodes during the simulation. The results are as following:

- mEER-RL achieves the highest number of delivered packets, indicating its superior performance in packet delivery by 66.47% compared to LEACH and 11.03% compared to EER-RL.
- While mobile mEER-RL performs better than LEACH and EER-RL in terms of sent packets, it delivers fewer packets compared to fixed mEER-RL, with 56.79% improvement over LEACH and only 5.08% compared to EER-RL. This can be attributed to the overhead associated with managing node movement in mobile IoT networks.

The results indicating that the inclusion of packet delivery probability within the reward function of mEER-RL likely inspires the RL agent to prioritize reliable routes with a higher chance of successful packet delivery.

Table 24 Packet delivery evaluation metric for fixed and mobile mEER-RL performance comparison with existing protocols

Evaluation metric	LEACH	EER-RL	Fixed mEER-RL	Mobile mEER-RL
Total number of sent packets	245095	365738	405882	384223

mEER-RL demonstrates significant advantages in packet delivery due to several factors:

- **Re-clustering with Communication Priority:** The re-clustering communication priority scheme in mEER-RL prevents packet duplication and ensures efficient data flow within and between clusters.

- **Packet Delivery in Reward Function:** The reward function plays a crucial role by promoting selection of routes with a higher probability of successful packet delivery.
- **Dynamic Learning:** The RL agent in mEER-RL continuously learns and adapts its routing policy, potentially discovering more reliable paths over time.

While mobile mEER-RL experiences some instability due to network dynamics, its ability to adapt and explore new routes contributes to a good PDR, making it a valuable option for mobile IoT networks deployments.

5.3.6 Results Summary and Discussion of mEER-RL

Performance

This section comprehensively analyzed the performance of mEER-RL (both fixed and mobile versions) compared to existing routing protocols, LEACH, and EER-RL, across various metrics. Here's a consolidated summary of the key findings:

- **Network Lifetime and Energy Consumption:** mEER-RL outperforms other protocols in extending network lifetime and minimizing energy consumption. This is achieved through a combination of factors:
 - **Energy-Aware Reward Function:** The reward function incentivizes the RL agent to select energy-efficient routes.
 - **Dynamic Re-clustering:** Re-clustering based on energy thresholds and location changes (mobile mEER-RL) helps distribute load and prevent node overload, extending lifetime.
 - **Learning and Adaptation:** The RL agent continuously learns and refines routing strategies to optimize energy usage and network longevity.
- **Packet Delivery:** mEER-RL demonstrates superior packet delivery performance compared to other protocols. This is accredited to:

- **Re-clustering Communication Priority:** This scheme prevents packet duplication and ensures efficient data flow within and between clusters.
- **Packet Delivery in Reward Function:** The reward function promotes selecting routes with a higher probability of successful delivery.
- **Dynamic Learning:** The RL agent's ability to learn and adapt to network changes contributes to discovering more reliable paths over time.
- **Distance Traveled:** mEER-RL achieves shorter average traveled distances compared to LEACH due to multi-hop communication and the RL agent's focus on hop count within the reward function. While mobile mEER-RL presents some variation due to mobility, it maintains good performance through exploration of new routes.

5.4 Chapter summary

This chapter evaluated the performance of the presented and enhanced routing protocol mEER-RL, designed for energy-efficient and reliable data transmission in IoT networks. Compared to existing protocols, mEER-RL achieved:

- Extended network lifetime and lower energy consumption through efficient CH election, dynamic re-clustering, and an energy-aware reward function for the RL agent.
- Improved packet delivery reliability by minimizing packet drops and prioritizing routes with high success probability.
- Effective adaptation to network changes via dynamic learning rate adjustments, making it suitable for both fixed and mobile IoT networks.

mEER-RL v4.2 is the most suitable choice for fixed networks, while Mobile mEER-RL with $\varepsilon = 0.99$ and alpha reset strategy 1 shows promise

for mobile networks. Further research is needed to optimize mobile network performance.

Chapter 6: Conclusion and Future Work

This chapter concludes the thesis by summarizing the key findings of the proposed protocol mEER-RL, which is a novel routing protocol for IoT networks that utilizes Reinforcement Learning for dynamic routing and network management. It also highlights potential areas for future exploration to further enhance mEER-RL's performance and applicability.

a) Conclusion

This thesis investigated the effectiveness of mEER-RL, a routing protocol for IoT networks that uses RL for dynamic cluster formation, route selection, and network adaptation. This protocol is built over the foundation of the previous cluster-based protocol EER-RL, by adding and modifying important parameters and methods, including improved multi-hop communication method, packet duplication prevention, increased protocol reliability by incorporating PDR within the RL agent's reward, and the consideration of nodes' mobility.

Through extensive simulations, we compared mEER-RL (both fixed and mobile versions) against existing protocols, LEACH and EER-RL. The evaluation focused on critical metrics such as network lifetime, energy consumption, packet delivery ratio, and traveled distance.

The results demonstrated that mEER-RL outperforms other protocols in all these aspects:

- **Network Lifetime and Energy Consumption:** mEER-RL significantly extends network lifetime and minimizes energy consumption through an energy-aware reward function, dynamic re-clustering, and continuous learning by the RL agent.
- **Packet Delivery:** mEER-RL achieves superior packet delivery performance due to its re-clustering communication priority scheme, focus on packet delivery probability in the reward function, and the RL agent's ability to learn and adapt to network changes.

- **Distance Traveled:** mEER-RL promotes shorter traveled distances compared to LEACH by utilizing multi-hop communication and incorporating hop count within the reward function. While mobile mEER-RL experiences some variation due to mobility, it maintains good performance through exploration of new routes.

These findings suggest that mEER-RL is a promising routing protocol for IoT, especially for mobile network deployments where network dynamics are a significant factor. The dynamic learning and adaptation capabilities of the RL agent in mEER-RL enable it to efficiently manage network resources and overcome challenges associated with changing network conditions.

b) Limitations and Future Work

While mEER-RL presents significant advantages, it's essential to acknowledge limitations and potential areas for future exploration:

- **Limited Control over Speed and Location:** While the RL agent learns to adapt to node movement, it cannot directly control or predict node speeds or locations for more improved performance related to re-clustering decision and route selection. This is an inherent limitation of IoT networks with mobile nodes.
- **Exploration vs. Exploitation Trade-off:** The RL agent balances exploration (trying new routes) and exploitation (utilizing learned efficient routes) to optimize performance. Future work could explore techniques to dynamically adjust this balance based on network conditions.
- **Impact on Specific Applications:** The evaluation focused on general performance metrics. Further investigation into mEER-RL's effectiveness in various IoT application contexts with varying requirements (e.g., delay-sensitive vs. energy-efficient) would be valuable.

Overall, mEER-RL, with its dynamic learning and adaptation capabilities, emerges as a promising routing protocol for IoT networks, particularly for mobile network deployments where network dynamics play a significant role.

In addition to the areas mentioned above, more specific enhancements that could be considered for future development of mEER-RL:

- **Inter-Cluster Communication with Mobile CHs:** Explore the feasibility of CHs using relay CMs from the next cluster to communicate with the distant CH of that cluster which is outside the transmission range.
- **Dynamic CH Range Threshold:** Investigate the possibility of dynamically adjusting the distance threshold for CH selection based on factors like network size, number of nodes, and desired cluster head percentage. This could potentially lead to more efficient cluster formation.
- **CM-to-CM Communication beyond Transmission Range:** Consider scenarios where a distant CM needs to communicate with the CH through another CM that is also beyond the transmission range. In such cases, the protocol could explore the possibility of finding a closer node within another cluster to join it and consume less energy.

By exploring these future work directions and enhancements, we can further improve mEER-RL's capabilities and broaden its applicability in diverse IoT networks deployment scenarios.

References

- [1] Weber, R.H. and Weber, R., 2010. Internet of things (Vol. 12). Heidelberg: Springer.
- [2] Yang, K., 2014. Wireless sensor networks, pp.1-250.
- [3] Sumathi, J. and Velusamy, R.L., 2021. A review on distributed cluster-based routing approaches in mobile wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 12, pp.835-849.
- [4] Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y.C. and Kim, D.I., 2019. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4), pp.3133-3174.
- [5] Mammeri, Z., 2019. Reinforcement learning based routing in networks: Review and classification of approaches. *Ieee Access*, 7, pp.55916-55950.
- [6] Muchintala, P.R. and Hash, L., 2016. Routing Protocols for MANETs.
- [7] Sutton, R.S. and Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.
- [8] Mutombo, V.K., Lee, S., Lee, J. and Hong, J., 2021. EER-RL: energy-efficient routing based on reinforcement learning. *Mobile Information Systems*, 2021, pp.1-12.
- [9] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H., 2000, January. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences* (pp. 10-pp). IEEE.
- [10] Arat, F. and Demirci, S., 2020, April. Energy and QoS aware analysis and classification of routing protocols for IoT and WSN. In *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)* (pp. 221-225). IEEE.

- [11] Lombardi, M., Pascale, F. and Santaniello, D., 2021. Internet of things: A general overview between architectures, protocols and applications. *Information*, 12(2), p.87.
- [12] Jamali, J., Bahrami, B., Heidari, A., Allahverdizadeh, P. and Norouzi, F., 2020. *IoT Architecture. Towards the internet of things*. Springer International Publishing, pp.9-31.
- [13] Roberts, C.M., 2006. Radio frequency identification (RFID). *Computers & security*, 25(1), pp.18-26.
- [14] Tiwari, S., 2016, December. An introduction to QR code technology. In 2016 international conference on information technology (ICIT) (pp. 39-44). IEEE.
- [15] Yang, S.H. and Cao, Y., 2008. *Networked control systems and wireless sensor networks: theories and applications*.
- [16] Lansky, J., Ali, S., Rahmani, A.M., Yousefpoor, M.S., Yousefpoor, E., Khan, F. and Hosseinzadeh, M., 2022. Reinforcement Learning-Based Routing Protocols in Flying Ad Hoc Networks (FANET): A Review. *Mathematics*, 10(16), p.3017.
- [17] Lee, I. and Lee, K., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business horizons*, 58(4), pp.431-440.
- [18] Kotha, H.D. and Gupta, V.M., 2018. IoT application: a survey. *Int. J. Eng. Technol*, 7(2.7), pp.891-896.
- [19] Khanna, A. and Kaur, S., 2020. Internet of things (IoT), applications and challenges: a comprehensive review. *Wireless Personal Communications*, 114, pp.1687-1762.
- [20] Amutha, J., Sharma, S. and Sharma, S.K., 2021. Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: Review, taxonomy,

research findings, challenges and future directions. *Computer Science Review*, 40, p.100376.

[21] Nasser, N., Karim, L., Ali, A., Anan, M. and Khelifi, N., 2017, December. Routing in the internet of things. In *GLOBECOM 2017-2017 IEEE Global Communications Conference* (pp. 1-6). IEEE.

[22] Poluru, R.K. and Naseera, S., 2017. A Literature Review on Routing Strategy in the Internet of Things. *Journal of Engineering Science & Technology Review*, 10(5).

[23] Amiri, G.A., 2021. Routing Issues in Internet of Things (IoT): A Survey (No. 5100). EasyChair.

[24] Dhumane, A. and Prasad, R., 2015. Routing challenges in internet of things. *CSI Communications*, pp.19-20.

[25] Dhumane, A., Prasad, R. and Prasad, J., 2016, March. Routing issues in internet of things: a survey. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, pp. 16-18).

[26] Thuy, T.T.T., Routing protocols in internet of Things. Aalto University.

[27] Bhushan, B. and Sahoo, G., 2019. Routing protocols in wireless sensor networks. *Computational intelligence in sensor networks*, pp.215-248.

[28] Yue, Y.G. and He, P., 2018. A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions. *Information Fusion*, 44, pp.188-204.

[29] Sara, G.S. and Sridharan, D., 2014. Routing in mobile wireless sensor network: A survey. *Telecommunication Systems*, 57, pp.51-79.

[30] Rezazadeh, J., Moradi, M. and Ismail, A.S., 2012. Mobile wireless sensor networks overview. *International Journal of Computer Communications and Networks*, 2(1), pp.17-22.

- [31] Yuan, D., Kanhere, S.S. and Hollick, M., 2017. Instrumenting Wireless Sensor Networks—A survey on the metrics that matter. *Pervasive and Mobile Computing*, 37, pp.45-62.
- [32] Naser, J.I. and Kadhim, A.J., 2020. Multicast routing strategy for SDN-cluster based MANET. *International Journal of Electrical & Computer Engineering* (2088-8708), 10(5).
- [33] Elira, B., Keerthana, K.P. and Balaji, K., 2021. Clustering scheme and destination aware context based routing protocol for VANET. *International Journal of Intelligent Networks*, 2, pp.148-155.
- [34] Manuel, A.J., Deverajan, G.G., Patan, R. and Gandomi, A.H., 2020. Optimization of routing-based clustering approaches in wireless sensor network: Review and open research issues. *Electronics*, 9(10), p.1630.
- [35] Sankar, S., Ramasubbareddy, S., Chen, F. and Gandomi, A.H., 2020, December. Energy-efficient cluster-based routing protocol in internet of things using swarm intelligence. In *2020 IEEE symposium series on computational intelligence (SSCI)* (pp. 219-224). IEEE.
- [36] Shahraki, A., Taherkordi, A., Haugen, Ø. and Eliassen, F., 2020. A survey and future directions on clustering: From WSNs to IoT and modern networking paradigms. *IEEE Transactions on Network and Service Management*, 18(2), pp.2242-2274.
- [37] Li, J., Silva, B.N., Diyan, M., Cao, Z. and Han, K., 2018. A clustering based routing algorithm in IoT aware Wireless Mesh Networks. *Sustainable cities and society*, 40, pp.657-666.
- [38] Abbasi, A.A. and Younis, M., 2007. A survey on clustering algorithms for wireless sensor networks. *Computer communications*, 30(14-15), pp.2826-2841.
- [39] Sholla, S., Kaur, S., Begh, G.R., Mir, R.N. and Chishti, M.A., 2017. Clustering internet of things: a review. *Journal of Science and Technology: Issue on Information and Communications Technology*, 3(2), pp.21-27.

- [40] Sutton, R.S. and Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.
- [41] Kolomvatsos, K. and Anagnostopoulos, C., 2017, June. Reinforcement learning for predictive analytics in smart cities. In *Informatics* (Vol. 4, No. 3, p. 16). MDPI.
- [42] Milan, G., Vassio, L., Drago, I. and Mellia, M., 2021, August. RL-IoT: reinforcement learning to interact with IoT devices. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.
- [43] Iqbal, M.M., Waseem, M., Manan, A., Liaqat, R., Muqet, A. and Wasaya, A., 2021, January. IoT-enabled smart home energy management strategy for DR actions in smart grid paradigm. In *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)* (pp. 352-357). IEEE.
- [44] Firouzi, F., Farahani, B., Panahi, E. and Barzegari, M., 2021, August. Task offloading for edge-fog-cloud interplay in the healthcare Internet of Things (IoT). In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-8). IEEE.
- [45] Nie, L., Ning, Z., Obaidat, M.S., Sadoun, B., Wang, H., Li, S., Guo, L. and Wang, G., 2020. A reinforcement learning-based network traffic prediction mechanism in intelligent internet of things. *IEEE Transactions on Industrial Informatics*, 17(3), pp.2169-2180.
- [46] Uprety, A. and Rawat, D.B., 2020. Reinforcement learning for iot security: A comprehensive survey. *IEEE Internet of Things Journal*, 8(11), pp.8693-8706.
- [47] Malik, T.S., Malik, K.R., Afzal, A., Ibrar, M., Wang, L., Song, H. and Shah, N., 2022. RL-IoT: Reinforcement learning-based routing approach for cognitive radio-enabled IoT communications. *IEEE Internet of Things Journal*, 10(2), pp.1836-1847.

- [48] Farag, H. and Stefanovič, Č., 2021, December. Congestion-aware routing in dynamic IoT networks: A reinforcement learning approach. In 2021 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.
- [49] Heinzelman, W.B., 2000. Application-specific protocol architectures for wireless networks (Doctoral dissertation, Massachusetts Institute of Technology).
- [50] Kim, D.S. and Chung, Y.J., 2006, June. Self-organization routing protocol supporting mobile nodes for wireless sensor network. In First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06) (Vol. 2, pp. 622-626). IEEE.
- [51] Xiangning, F. and Yulin, S., 2007, October. Improvement on LEACH protocol of wireless sensor network. In 2007 international conference on sensor technologies and applications (SENSORCOMM 2007) (pp. 260-264). IEEE.
- [52] Ahlawat, A. and Malik, V., 2013, April. An extended vice-cluster selection approach to improve v leach protocol in WSN. In 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT) (pp. 236-240). IEEE.
- [53] Kulkarni, N., Prasad, N.R. and Prasad, R., 2014, May. MMOHRA: Mobility aware multi-objective hybrid routing algorithm for Wireless Sensor Networks. In 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE) (pp. 1-5). IEEE.
- [54] Moussa, N., Hamidi-Alaoui, Z. and El Belrhiti El Alaoui, A., 2020. ECRP: an energy-aware cluster-based routing protocol for wireless sensor networks. *Wireless Networks*, 26, pp.2915-2928.

- [55] Khabiri, M. and Ghaffari, A., 2018. Energy-aware clustering-based routing in wireless sensor networks using cuckoo optimization algorithm. *Wireless Personal Communications*, 98, pp.2473-2495.
- [56] Gaber, T., Abdelwahab, S., Elhoseny, M. and Hassanien, A.E., 2018. Trust-based secure clustering in WSN-based intelligent transportation systems. *Computer Networks*, 146, pp.151-158.
- [57] Maheshwari, P., Sharma, A.K. and Verma, K., 2021. Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks*, 110, p.102317.
- [58] Sahoo, B.M., Pandey, H.M. and Amgoth, T., 2021. GAPSO-H: A hybrid approach towards optimizing the cluster based routing in wireless sensor network. *Swarm and Evolutionary Computation*, 60, p.100772.
- [59] Sennan, S., Ramasubbareddy, S., Balasubramaniam, S., Nayyar, A., Kerrache, C.A. and Bilal, M., 2021. MADCR: Mobility aware dynamic clustering-based routing protocol in Internet of Vehicles. *China Communications*, 18(7), pp.69-85.
- [60] Salim, A., Khedr, A.M., Alwasel, B., Osamy, W. and Aziz, A., 2023. SOMACA: A New Swarm Optimization-Based and Mobility-Aware Clustering Approach for the Internet of Vehicles. *IEEE Access*.
- [61] Hu, H., Han, Y., Wang, H., Yao, M. and Wang, C., 2021. Trust-aware secure routing protocol for wireless sensor networks. *ETRI Journal*, 43(4), pp.674-683.
- [62] Kiani, F., Amiri, E., Zamani, M., Khodadadi, T. and Manaf, A.A., 2015. Research Article Efficient Intelligent Energy Routing Protocol in Wireless Sensor Networks.
- [63] Kiani, F., 2017, November. Reinforcement learning based routing protocol for wireless body sensor networks. In 2017 IEEE 7th international symposium on cloud and service computing (SC2) (pp. 71-78). IEEE.

- [64] Soni, S. and Shrivastava, M., 2018. Novel learning algorithms for efficient mobile sink data collection using reinforcement learning in wireless sensor network. *Wireless Communications and Mobile Computing*, 2018, pp.1-13.
- [65] Guo, W., Yan, C. and Lu, T., 2019. Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing. *International Journal of Distributed Sensor Networks*, 15(2), p.1550147719833541.
- [66] Krishnan, M. and Lim, Y., 2021. Reinforcement learning-based dynamic routing using mobile sink for data collection in WSNs and IoT applications. *Journal of Network and Computer Applications*, 194, p.103223.
- [67] Gantassi, R., Ben Gouissem, B., Cheikhrouhou, O., El Khediri, S. and Hasnaoui, S., 2021. Optimizing quality of service of clustering protocols in large-scale wireless sensor networks with mobile data collector and machine learning. *Security and Communication Networks*, 2021, pp.1-12.
- [68] Abadi, A.F.E., Asghari, S.A., Marvasti, M.B., Abaei, G., Nabavi, M. and Savaria, Y., 2022. RLBEED: Reinforcement-learning-based energy efficient control and routing protocol for wireless sensor networks. *IEEE Access*, 10, pp.44123-44135.
- [69] Sharma, T., Balyan, A., Nair, R., Jain, P., Arora, S. and Ahmadi, F., 2022. ReLeC: A Reinforcement Learning-Based Clustering-Enhanced Protocol for Efficient Energy Optimization in Wireless Sensor Networks. *Wireless Communications and Mobile Computing*, 2022.
- [70] Firdous, S., Bibi, N., Wahid, M. and Alhazmi, S., 2022. Efficient Clustering Based Routing for Energy Management in Wireless Sensor Network-Assisted Internet of Things. *Electronics*, 11(23), p.3922.
- [71] Moussa, N., Nurellari, E., Azbeg, K., Boulouz, A., Afdel, K., Koutti, L., Salah, M.B. and El Alaoui, A.E.B., 2023. A reinforcement learning based routing protocol for software-defined networking enabled wireless

sensor network forest fire detection. Future Generation Computer Systems, 149, pp.478-493.

المخلص

التصاعد المستمر في عدد أجهزة إنترنت الأشياء يتطلب وجود بروتوكولات فعالة وموثوقة لنقل البيانات. بالرغم من ذلك، فإن بروتوكولات النقل والتوجيه التقليدية تواجه صعوبات في التأقلم مع الظروف المتغيرة والمحدودة الموارد الخاصة بشبكات إنترنت الأشياء بسبب عدة عوامل تتضمن اللامركزية، الطاقة المحدودة، والتغيرات الهيكلية المتكررة للشبكة. تقترح هذه الأطروحة بروتوكول توجيه جديد: التوجيه الموفر للطاقة استناداً إلى التعلم المعزز – المعدل، تم تصميمه خصيصاً لمواجهة هذه التحديات وتعزيز الأداء بتطبيقه في كل من شبكات إنترنت الأشياء الثابتة والمتحركة. تم بناء هذا البروتوكول على الأسس الخاصة ببروتوكول سابق: التوجيه الموفر للطاقة استناداً إلى التعلم المعزز، وهو بروتوكول قائم على نظام عنقودي وموجه للتعليم المعزز ويركز على رفع كفاءة الطاقة أثناء نقل بيانات الأشياء. يحتوي البروتوكول المعدل على تغييرات مهمة لمعالجة المشكلة الناجمة عن تنقل الأجهزة، وهو عامل حاسم غالباً ما يتم إهماله في تصميم البروتوكولات القائمة على النظام العنقودي مع التعليم المعزز لإنترنت الأشياء. ومن خلال دمج الاعتبارات الخاصة بتنقل الأجهزة مع المكافأة الوظيفية للعميل الخاص بالتعليم المعزز، يقوم البروتوكول المعدل بتكييف قرارات التوجيه الخاصة به ديناميكياً مع ظروف الشبكة، مما يؤدي إلى تحسين نسبة تسليم حزم البيانات وموثوقية الشبكة بشكل عام. تقدم هذه الأطروحة دراسة شاملة استقصائية لجميع أنواع بروتوكولات التوجيه القائمة على النظام العنقودي والمصممة لشبكات إنترنت الأشياء. كما وتتم مراجعة تقنيات التعليم المعزز للذكاء الاصطناعي ل يتم تطبيقها على بروتوكول التوجيه. تقوم هذه الأطروحة بتقييم أداء البروتوكول المعدل بشكل شامل من خلال عمليات المحاكاة ومقارنته بالبروتوكول الأساسي للبروتوكولات العنقودية والبروتوكول الغير معدل. يركز التقييم على مقاييس مهمة مثل عمر الشبكة، استهلاك الطاقة، نسبة تسليم الحزم، وتأثير تنقل الأجهزة. توضح النتائج أن البروتوكول المعدل يوازن بشكل فعال بين كفاءة الطاقة والموثوقية والقدرة على التكيف، مما يجعله حلاً واعداً للتوجيه في شبكات إنترنت الأشياء المعقدة والديناميكية. حيث تشمل المساهمات الرئيسية لهذا العمل تعزيز الموثوقية وتحسين نسبة تسليم حزم البيانات وتعزيز القدرة على التكيف لكل من بيئات إنترنت الأشياء الثابتة والمتحركة، وتحسين كفاءة الطاقة من خلال زيادة الجودة لاختيار مسارات التوجيه وتقليل نقل البيانات من خلال تحسين تقنيات الاتصال متعددة القفزات. للحصول على أداء متقدم، يعرض البروتوكول المعدل أداءً متقدماً مقارنةً بالبروتوكولات الأساسية من حيث عمر الشبكة واستهلاك الطاقة ونسبة نجاح تسليم حزمة البيانات.



جامعة البحرين
كلية الهندسة
قسم الهندسة الكهربائية والإلكترونية

التعليم المعزز للشبكات اللاسلكية الذكية

أطروحة مقدمة استكمالاً لمتطلبات درجة ماجستير العلوم في برنامج نظام الذكاء الاصطناعي

إعداد: نجيبة محمد
الرقم الجامعي: ٢٠١١٢٦٩٥

إشراف: أ.د. مهذب منجود

مايو ٢٠٢٤