Performance Comparison of Traditional and Reinforcement Learning Based Routing Protocols

Mohammad Dib, Hasnain Syed
School of Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada

Abstract*— As the field of wireless sensor networks advances, engineers must build more efficient models to fuel our innovations in sensor networks. The low-cost nature of these networks makes it difficult to operate at a large scale, which is why energy efficient routing protocols are needed. In this paper, three classifications of routing protocols are discussed and the benefits of each in a certain scenario. Flat, hierarchical, and Q-learning based routing protocols. The paper aims to study the characteristics of these three categories of routing protocols to make a conclusion of which protocol will perform best depending on the requirements of the application. There is no best protocol for every application, each will have its own advantages and disadvantages. It is up to the engineer designing the WSN, which performance metrics they wish to optimize to get acceptable results.

***Index Terms*— WSN, Flat routing, Hierarchical routing, Reinforcement Learning, Q-learning**

I. Introduction

Wireless Sensor Network (WSN) technology enabled the Internet of Things (IoT) and has grown in the last decade as the emergence of inexpensive sensors and electronics has enabled engineers to develop solutions for many industries. The development of light and efficient software is important to facilitate the development of WSNs with low cost hardware. The advancements in AI have accelerated the advancements of WSNs. Reinforcement learning (RL) is a branch of machine learning related to how software entities in an environment react to different factors based on a reward system [1]. Exploring different types of routing protocols and how different RL algorithms influence WSNs is extremely important to understand to apply IoT for different applications. One of the main applications of RL in WSNs is making routing intelligent [1]. Routing is the process of finding the path from the source to the destination which takes place in the network layer [1].

Routing is an important research area that needs to be taken into consideration when working with WSN. Due to limited resources, there are many aspects that affect the longevity and the reliability of the network, especially power efficiency [2]. Therefore, it is necessary to use protocols that find the route with the optimal energy consumption from the source to the sink in an environment that is constantly changing.

Traditional routing protocols (Flat and Hierarchical), which do not employ machine learning, place significant assumptions on the changes that the network may go through. It is important to note that these assumptions cannot stand in the face of uncertainties in WSNs, since some applications of WSNs operate in a high dynamic and unpredictable environment [1].

Using machine learning (ML) in routing helps find an optimal route that will ensure the energy is consumed efficiently, ultimately increasing the life of sensors in the network and its reliability [1]. Reinforcement learning is significantly used as an ML tool that solves routing problems, compared to supervised and unsupervised learning. This is since RL is an online learning method (does not need a predefined dataset for learning) which is important in routing due to the unpredicted changes of the topology of the network [2].

II. Background

A Wireless Sensor Network (WSN) consists of multiple sensor nodes that are autonomous, small, cheap, and operate at low power [1]. These features along with the ease of deployment, made WSN a suitable technology for real-time applications [2]. Additionally, a WSN usually has a huge number of sensors, which may be deployed in an environment that is changing dynamically [2]. This raises the importance of the need for algorithms that are scalable, adaptive, and efficient [2]. Routing is an important issue that needs to be addressed, because of power constraints, limited transmission bandwidth, low memory, and processing capabilities [2]. This leads us to one of the most important goals of routing, which is reducing the energy waste to increase the lifetime of the network [2].

Machine Learning (ML) is a set of techniques aiming to achieve autonomy without the need for explicit programming, which is done through learning from experiences (e.g. a predefined dataset) [2]. This feature will aid improving the performance of WSN, such as taking the optimal routes and adapting to network changes in a dynamically changing environment [1]. ML is usually classified into sub-categories, but this report will focus on reinforcement learning as this technique does not require any predefined dataset which makes it ideal for networks in dynamic environments [2].

A. Overview of Traditional Routing Protocols

Traditional routing protocols were the first type of protocols when the field of WSNs emerged. The classification of the routing protocols is based on the structure of the network, which then are categorised further into subcategories [3]. The two categories this report will focus on are Flat-Based and Hierarchical-Based Routing.

Flat-Based protocol assumes that every node will play the same role and since there are a large amount of sensor nodes, the network does not need to assign a node identifier [3]. The nodes receive information by a central base station through a form of a query, the base station will then wait for a response from an available node [3]. The act of sending the message to all neighbors of each node to find the destination node is called Flooding.

Hierarchical-Based Routing is when the nodes are split into categories based on the type of work they would do. The simplest Hierarchical-Based architecture will consist of high energy nodes known as cluster heads, designated for processing and sending data to the base station, and low energy nodes which are designated to collect sensor data to send it back to their corresponding cluster head [3].

B. Drawbacks of Traditional Routing Protocols

One of the biggest challenges in WSNs is the scalability of the network. As more sensor nodes are added into the network, the architecture needs to employ techniques that will ensure proper function in scenarios where there is a sudden influx on available network or a sudden drop in the available sensors [3]. In Flat-Based protocols, as the number of sensors increase, each node will have to send the same information to more nodes, causing inefficient use of resources [3]. Hierarchical-Based protocols solve the scalability issue but suffers from a network dependency issue. This architecture creates a binary tree-like structure called clusters, where nodes higher up in the tree are generally high energy nodes. Since there is dependency of nodes within the network, a failure of a node can create disruption [4].

Traditional routing protocols (Flat and Hierarchical), which don't employ machine learning, place significant assumptions on the changes that the network may go through; However, these assumptions are not enough to address the complex and highly dynamic changes of the WSN [5].

C. The Use of RL based Routing Protocols

Q-Learning is one of the widely used types of reinforcement learning introduced in 1989, the model is based on Wakin's function, called the Q-function [5]. In Q-learning, the agent (sensor node in our context) goes through a sequence of steps. Each step corresponds to a state which performs an action, receives a reward and the Q-value is updated. Every possible state and action are assumed to be given, so the action is performed in every state which ensures that the desired Q-value is reached [5].

Routing protocols should take into consideration the required energy, the end-to-end delay, and packet delivery ratio (PDR) [6]. As mentioned before, RL is the most widely used ML algorithm to tackle the routing problem due to its suitability for this task. Thus, many researchers tried to apply RL to enhance routing in WSN. Habib *et al.* [6] compared several RL-based routing protocols where many of them were employing Q-Learning methods.

One of the discussed methods is Feedback routing for optimizing multiple sinks in WSN with RL (FROMS), which concentrates on improving the PDR and the energy consumption, by greatly decreasing the network overhead and allowing a full duplex connection between any two nodes. This method takes multiple sink nodes into account, and it has a process for recovery in case of any node failure. However, this method is susceptible to node failure, and routing errors may be caused by the movement of the sink node.

Another method that was mentioned is the Multi-agent RL based routing protocol with quality of service (QOS) support (MRL-QRP). This method is concerned with improving the PDR, the end-to-end delay, and energy usage of the network by taking advantage of the information available on each node about the network in order to find the best route (i.e. the sensor nodes will cooperate to find the best route). MRL-QRP has the following advantages, it outperforms ad hoc on demand distance vector (AODV) protocol, it gives an exceptional performance when there is a heavy traffic load. The disadvantages of MRL-QRP is that it overuses the available resources and causes a network overhead because of the fluctuations in the QoS requirements. Also, the initial routing table does not ensure QoS.

In addition, adaptive network routing based on RL, which aims at improving the PDR and the latency of the network, is suitable for networks with topologies that go through regular changes. Moreover, this method achieves fast delivery and processing time with the help of synchronized routing. Other advantages of this method is the ease of adapting to changes in the network topology, and changes in the traffic conditions. However, the disadvantages of this method is that it doesn't take the energy consumption into consideration, it wasn't compared with the standardized protocols, and this average processing time is high due to the need for learning the full network.

III. Design Implementation

This section includes the design implementation of the different routing protocols that will be compared. It is worth noting that the proposed protocols were developed using python. Since the routing problem can be formulated as a graph problem, NetworkX package [7], which is used to build and analyze graphs, was used to build, and manipulate the sensor network. The following section will go through the different design choices and implementation of the proposed protocols.

A. Flat Protocol

The most important concept in flat-based routing protocol is that every node in the network plays the same role. This simplifies the implementation which is good in some applications but cannot be applied everywhere. The advantage of having all the nodes play the same role is that it will increase the reliability of the network. If an X number of nodes die, the network will still have other nodes performing the same functions. Flooding is a commonly used flat based protocol. In this protocol, the node that must send a message will send the message to every neighbor, every neighbor will send the message to their neighbors, and so on. This will continue until the message has reached its destination [3]. The approach taken to simulate a flat based protocol is a random routing. Instead of sending the message to every neighbor as it is done in flooding, each iteration of the simulation sends the message to one of its neighbors and continues this until the destination has been reached.

In the simulation, a small network consisting of 11 nodes is set up, as seen in Fig 1, where node 0 is the sender and node 10 is the destination. Each node has 4 neighbors they can send the message to, these 3 neighbors are randomly generated when the network is created.

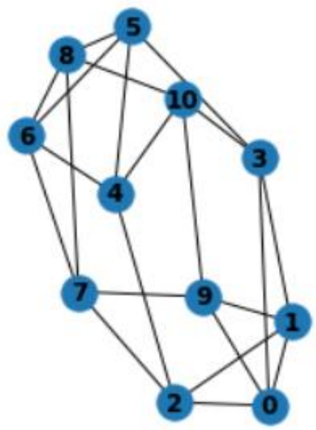

Fig.1. Network topology used in the flat based routing protocol simulation

The following is how the random routing algorithm works:

1. Choose a node to simulate the sender, node 0 in this case
2. The node will then randomly pick a neighbor and send the message
3. A check will be performed to see if the current node is the destination
4. If the current node is the destination, the algorithm will stop and output the path it took
5. If the current node is not the destination, the algorithm will send it to one of its neighbors, while making sure that that node has not been visited to avoid duplicate nodes in the path

The number of hops the message takes to reach its destination will be the main parameter measured. To simulate energy consumption in a simplified model, each node will have 100 battery units, each transmission will take 0.5 units. The simulation will send 100 messages from the sender to its destination and the remaining battery of each node will be analyzed.

To study how this network performs in dynamic environment where a node dies due to hardware failure or energy depletion, one of the nodes will be removed from the network and the same 100 packets will be sent from the same sender to the same destination. This will allow us to elaborate on our theoretical findings that flat based routing protocols are reliable in environments where there are node failures.

B. Hierarchal Protocol

The nodes in hierarchal routing protocols have different roles, as opposed to flat routing protocols, where usually nodes with lower energy will simply collect the data, and then send this data to higher energy nodes (cluster heads), which will operate as gateways to process and transmit the collected data to the base station. LEACH, which is a famous hierarchal routing protocol, runs a clustering algorithm to divide the nodes in the network to several clusters, by first choosing the cluster heads, and then a message is sent to the surrounding nodes [3]. Then depending on the signal strength, sensor nodes will know to which cluster they belong [3]. However, the hierarchal protocol implemented in this paper does not implement LEACH protocol exactly. The designed hierarchal protocol clusters nodes in the same region first, and then a cluster head will be chosen for each cluster. Moreover, the cluster head role will be rotated among the nodes in the same cluster to ensure the same life cycle for all the nodes in the network. The detailed implementation steps are discussed below.

First, the dimensions of the area to be covered and the number of nodes will be specified. After that, the nodes will be assigned different locations, such that they will cover the area of interest uniformly. The base station (BS), which will be the sink of all the messages in the network, was placed in the middle of the area to be covered. This will ensure that the BS will have almost the same distance from the different clusters in the network.

Since this protocol is hierarchal, nodes can have different roles, and they must be categorized into cluster heads, and sensor nodes, where each cluster head and its surrounding nodes belong to the same cluster. Thus, to formulate the different node clusters, the number of clusters was specified, and k-means clustering algorithm was used to cluster nodes that are close together. K-mean clustering algorithm is an unsupervised learning algorithm that aims to group samples that fall in the same regional area [8]. The K-means algorithm used in the implementation, was from scikit-learn python library [8].

Each node will start with 50 energy units and depending on the role of the node on a given rotation, its energy will decrease by a certain amount. To illustrate, if the node is currently operating as a cluster head, its transmission will cost 1.5 units, while if it was a sensor node, its transmission cost will be 1 unit. The cluster head’s energy cost is more compared to the sensor node due to the receiving and the processing of transmitted messages by the sensor nodes. Whereas, sensor nodes will be only sensing and transmitting message. Furthermore, each energy unit is assumed to encompass the energy costs for all the operations in a node, such as transmission and processing.

The way this protocol will operate is as follows. Firstly, the number of cluster heads rotations will be specified. This will prevent the early death of some nodes and will distribute the burden of the cluster node rule among the different nodes on the cluster. Secondly, a cluster head will be chosen for each cluster using one of the two methods, where the desired method can be chosen. The methods are either the cluster heads will be randomly chosen from the cluster of nodes, or the node with the highest energy will be assigned the cluster head role Thirdly, a certain number of messages to send per round is specified. A random sensor node will be randomly chosen as a source to send a message to the base station, given that the chosen node has enough energy to transmit its message. Additionally, the message will be received by the base station only if the cluster head connected to the source node has enough energy. This means that there may be cases where the message will be sent by the source node, and not received by the base station, because its cluster head is dead. Finally, the energy of the involved nodes will be decreased depending on their current role, and their status (i.e. dead or alive) will be checked.

The following parameters are monitored throughout the operation of the network operation. The number of dead nodes, the number of transmitted messages, the number of received messages by the base station, and the total energy of the network, and they will be thoroughly discussed in the results section.

C. *RL Based Protocol*

Machine learning in WSNs is becoming more commonly used as networks become more complicated. In this paper, a basic Q-learning algorithm, a branch of reinforcement learning is applied to a flat sensor network to study its characteristics.

The overview of the Q-learning algorithm is to initialize a Q table with arbitrary values, iterate through as many possible actions and measure the reward for that action. This reward will be used to update the Q table. In the end, the final Q table will hold data for the algorithm to decide which actions to take to achieve the highest reward [5]. In this case, the optimal path taken from the sender to the receiver will yield the highest reward.

These values of reward and the Q-table are calculated from the Q-function:

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\}$$

$$Q^{\pi}(s,a) = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\}$$

Where,
E_{π}: is the expected value of the reward
s_t: is the current state
a_t: is the current action
γ: is the discount rate
r_t: is the reward for that state

The function iterates through k number of times and $Q^{\pi}(s,a)$ is the cumulative reward for that state and is store in the Q-table [6].

A discount rate determines the importance of rewards calculated in future actions. A discount rate of 1, will be good for the algorithm aim to perform to gain rewards for later in the iteration, whereas a discount rate of 0 will only consider the current reward. The discount rate chosen for this simulation is 0.8 to accelerate learning for long term rewards since there will be multiple actions performed to get to the destination [6].

The simulation is inspired by [9] of the implementation of finding shortest path based on Q-learning.

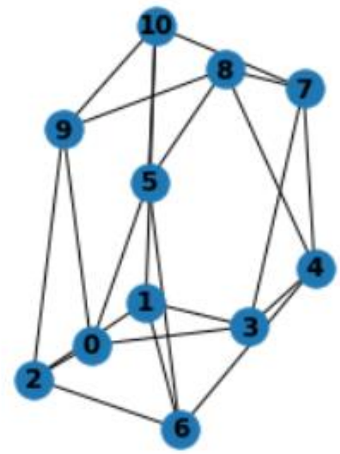

Fig. 2. The topology of the network on which the Q-learning algorithm will be performed

In Fig.2, the network topology seen is what is going to be used to implement Q-learning on.

To start preparing data to perform Q-learning, a reward matrix is needed to keep track of the rewards after every iteration. A Q-table is needed to keep track of the updated Q-values from the Q-function.

	0	1	2	3	4	5	6	7	8	9	10
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 3. Initialization of the reward values

In Fig. 3, all the rewards are set to 0, except the paths for the destination nodes since that is where a high reward is expected.

	0	1	2	3	4	5	6	7	8	9	10
0	-100.0	-100.0	0.0	0.0	-100.0	0.0	-100.0	-100.0	-100.0	0.0	-100.0
1	-100.0	-100.0	0.0	0.0	-100.0	-100.0	0.0	-100.0	-100.0	-100.0	0.0
2	0.0	0.0	-100.0	-100.0	-100.0	-100.0	0.0	-100.0	-100.0	0.0	-100.0
3	0.0	0.0	-100.0	-100.0	0.0	-100.0	-100.0	0.0	-100.0	-100.0	-100.0
4	-100.0	-100.0	-100.0	0.0	-100.0	-100.0	0.0	0.0	0.0	-100.0	-100.0
5	0.0	-100.0	-100.0	-100.0	-100.0	-100.0	0.0	-100.0	0.0	-100.0	0.0
6	-100.0	0.0	0.0	-100.0	0.0	0.0	-100.0	-100.0	-100.0	-100.0	-100.0
7	-100.0	-100.0	-100.0	0.0	0.0	-100.0	-100.0	-100.0	0.0	-100.0	0.0
8	-100.0	-100.0	-100.0	-100.0	0.0	0.0	-100.0	0.0	-100.0	0.0	-100.0
9	0.0	-100.0	0.0	-100.0	-100.0	-100.0	-100.0	-100.0	0.0	-100.0	0.0
10	-100.0	0.0	-100.0	-100.0	-100.0	0.0	-100.0	0.0	-100.0	0.0	-100.0

Fig. 4. Initialization of the Q-table values

In Fig. 4, the Q-table values are all set to 0. All the actions that are not possible are set to -100. An impossible path is the sender also being the destination or no edges to a node. For example, node 0 cannot directly send the message to node 10.

After these initializations, 50,000 actions are performed, exploring the solution space to populate the Q-table and calculate the rewards that will be used to find the optimal path in the routing simulation.

IV. Results

This section includes the performance measures of the different protocols, and the comparison between them. Performance measures that were takin into consideration are the energy of network, the packet delivery ratio, and the end-to-end delay.

A. *Flat based routing protocol*

To test the flat based routing protocol performance, the network discussed in the previous section was used to send message from node 0 to node 10. One iteration equates to 100 messages being sent from node 0 to node 10, and each iteration was run 100 times. The whole simulation will produce 1000 random paths. The battery of each node is set to 100 battery units, and each transmission consumes 0.5 battery units.

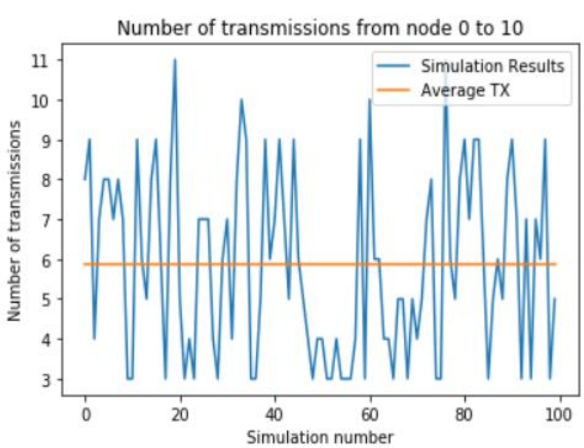

Fig. 5. The number of transmissions the message took to reach the destination of each of the simulation

As seen from Fig. 5, the number of transmissions is very random and independent from each simulation. The average number of hops the message took to get to the destination is approximately 6 hops. The optimal path is 3 hops. The worst possible path is hitting every single node and then reaching the destination, which is 11 hops. Eleven transmissions can be seen for some simulations.

The same simulation was run, but with one node removed to simulate a dying node.

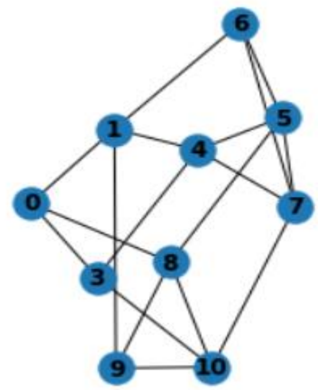

Fig. 6. The network topology after one of the nodes is dead

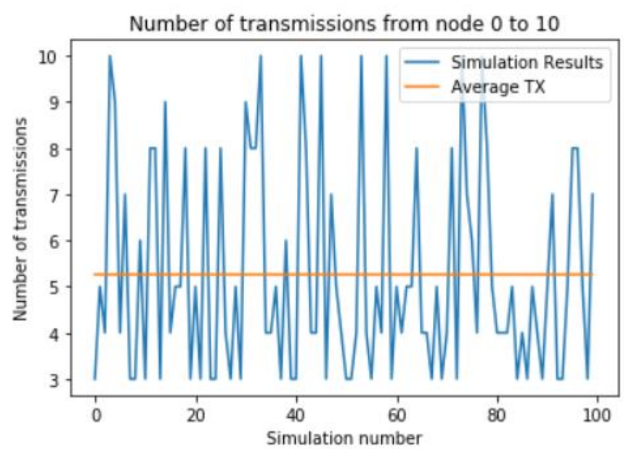

Fig. 7. The number of transmissions the message took to reach the destination of each of the simulation when one node died

As seen from Fig. 6, the graph is like the previous simulation as expected. Both are following the same random routing algorithm. In Fig. 7, The difference we see in this is that the average number of transmissions is lower in this case, close to 5. The reason for this is because a dying node also reduced the number of possible paths the node can take. The ratio of optimal path to all paths have increased when the node died, so there are more instances of taking a less costly path. The worst possible path cost also decreased from 11 to 10, which contributed to the average number of transmissions to go down.

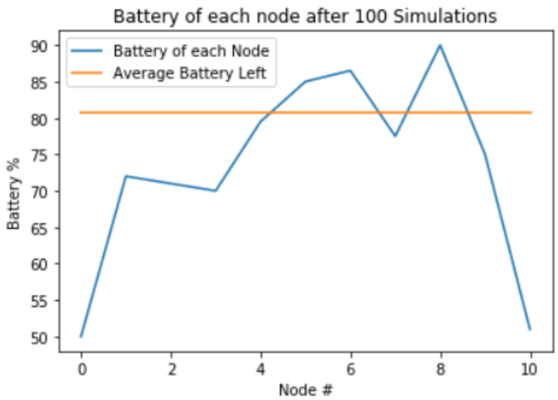

Fig. 8. The remaining battery in each node, and the average battery left in the network

The Fig. 8 is the battery profile of the original network. Node 0 and Node 10 have the lowest battery remaining since they are touched in every simulation as they are the sender and receiver. The battery for the other nodes is randomly distributed since not all paths include them. The average battery remaining is approximately 80%.

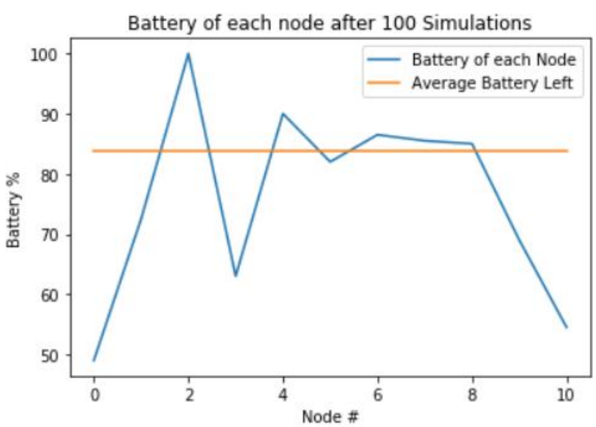

Fig. 9. The remaining battery in each node, and the average battery left in the network when one node dies

Fig. 9 also shows a similar pattern as the network without the dying node; the sender and receiver consuming the most battery. Although what is interesting is that the average battery remaining is approximately 85%, higher than the larger network. As discussed before, lower number of nodes increase the optimal path to total paths ratio, which means that a message is more likely to take an optimal path. The smaller the network gets; the random flat protocol will be more energy efficient.

B. *Hierarchal based routing protocol*

To test the Hierarchal protocol, an area of 30m × 30m was assumed to be the area of interest, in which 30 sensor nodes were scattered randomly. The number of clusters in the k-means algorithm was assumed to be 3, where the nodes will be clustered according to their locations in the network. The simulation of the network was run for 4 rounds of cluster head rotation. The topology of the network for the 4 different rounds can be seen in Fig. 11.

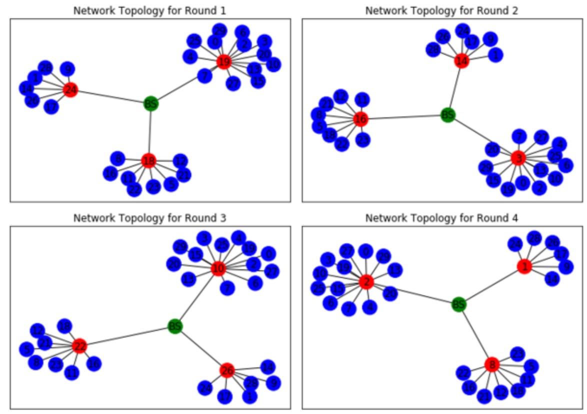

Fig. 10. Network topology for four rounds. Base station is the green node, cluster heads are in red, and normal senor nodes are in blue

It is worth noting that Fig. 10 does not indicate the actual physical location of the nodes in the area to be covered, because this figure aims to show a neat representation of the different clusters in the network. The Fig. 11 shows how the cluster heads (nodes in red), change from round to round. For example, the cluster heads were 18,19, and 24, while the cluster heads became 3,14,16 in round 2. Moreover, all the sensor nodes (nodes in blue) in each cluster are connected to their cluster head, and their cluster head is connected to the base station (node in green).

The energy of network was measured by running the simulation for 4 rounds, and in every round 50 messages were sent. Therefore, a total of 200 messages were sent from random sensor nodes to the base station. The energy of nodes was assumed to be 50 units, and it will decrease by 1 unit per transmission, if the node is a sensor node, and by 1.5 units per transmission, if the node is a cluster head. Fig. 12 shows the total energy profile of the network per message transmission.

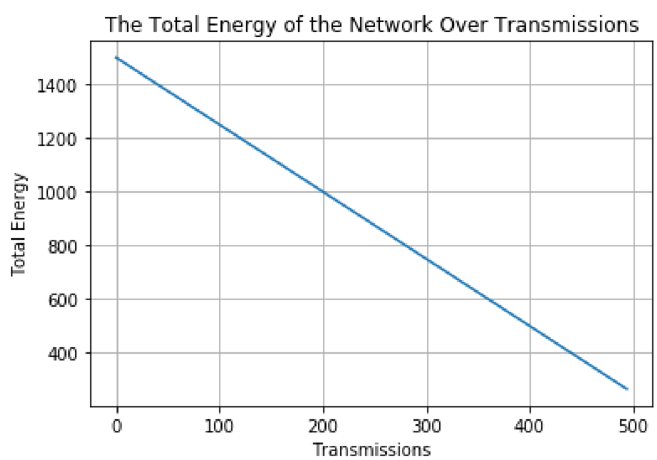

Fig. 11. The total energy of the network over time

Fig. 11 shows that the energy decreases linearly, which is expected because the energy cost for each transmission is the same. To illustrate, if a sensor node wants to transmit a message to the base station, it must send it first to its cluster head, which will cost 1 energy unit. After that, the cluster head will process this message, and then it will send it to the base station, which will cost 1.5 energy units. Thus, each transmission will cost a total of 2.5 units, and this value is constant.

Until now, the network was not pushed to its limit, and the results were acquired for normal operation of the network, where no nodes will die. Thus, the effect of the method by which the cluster head is chosen cannot be noticed. However, when the network is pushed to its limit such that some nodes will die, then the method of choosing the cluster head will be appreciated, which will be discussed below. To compare the two methods, the simulation for both methods was run for 10 rounds (with the same number of messages per round i.e. 50 messages), to ensure that some nodes will die, where the number of dead nodes per round was monitored. Fig. 13 shows the number dead nodes per round for the case of random cluster head assignment, and the assignment of the cluster heads based on the nodes with the highest energy.

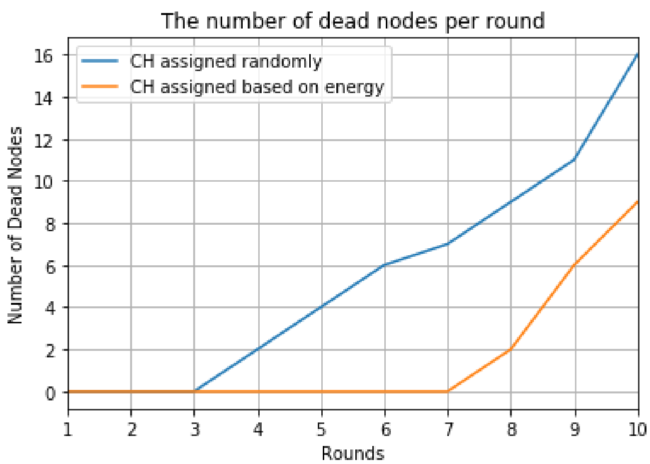

Fig. 12. The number of dead nodes per round

In Fig. 12, when the cluster heads are assigned randomly, the nodes start dying way earlier (round 3) compared to when the cluster head are assigned based on energy (round 7). Then, the number of dead nodes will increase in a steady linear fashion for both cases. Additionally, after 10 rounds, 16 nodes have died when random assignment was used, while only 9 nodes have died when energy-based assignment was used, which significantly lower. This behavior is expected, because some nodes maybe chosen for multiple consecutive rounds in the case of random cluster heads assignment, which will result in their early death. However, this problem is prevented for the case of energy-based assignment of the cluster head because the node with the highest energy in the cluster will be chosen to be the cluster head. This will distribute the burden of the cluster head fairly among the nodes in the same cluster. Therefore, it is more practical to choose the cluster heads based on node energy, because it will increase the longevity of the whole network for as long as possible, compared to choosing the cluster heads randomly.

TABLE 1: Calculating the PDR for both cluster head election methods

	Sent messages	Received messages	PDR
Random assignment	500	434	86.8%
Energy based	500	494	98.8%

Another important aspect of comparing the two different cluster head assigning methods, is the packet delivery ration (PDR). Again, running the simulation for 10 rounds, the following results were achieved for the different methods.
Table 1 shows that the PDR is significantly worse for the random assignment compared to the energy-based assignment. This could be explained by the fact that assigning cluster heads randomly may result in the election of some nodes that do not have enough energy to function as a cluster head for a full round. Thus, such a chosen cluster head will be able to direct messages from the sensor nodes to the base station for a certain number of transmissions, after which it will die before the end of the round. As a result, nodes in the same cluster will not be able to transmit any messages to the base station until the end of the given round, and the election of another cluster head for the next round. This will impose a huge reliability issue, when hierarchal routing protocols with random cluster head assignment is used. However, this issue can be mitigated by using a smarter approach for choosing the cluster heads, which, as seen by table 1, can be done assigning the cluster head role to the node with the highest energy in a given cluster. However, the problem of the cluster head being a single point of failure for a given round will still hold, which could be considered as the drawback of hierarchal routing protocols, but its effect can be reduced as discussed before by choosing the cluster heads intelligently.

Moving to the end-to-end delay in the network. Since the end-to-end delay is directly proportional to the number of hopes between the source and the sink, it can be inferred that the end-to-end delay will be constant for the proposed hierarchal routing protocol, because messages will always go through two hopes to reach their destination. To elaborate, one hope from the source sensor node to its cluster head, and another hope from the cluster head to the base station.

Regarding the adaptability of the hierarchal protocol in a dynamic environment, where nodes may die or added. As discussed before, cluster heads are considered as a single point of failure, so the death of a cluster head will degrade the performance of the network significantly, because the sensor nodes in its cluster won’t be able to transmit any messages to the base station, until the beginning of next round, where another cluster head will be elected. On the other hand, if a sensor head dies, it will not have a huge impact on the performance of the network, since its death will not affect the other nodes in the cluster. Moreover, there is a chance that the covered area by the dead node, is also covered by other nodes in the cluster due to the redundancy characteristic of wireless sensor networks. It is worth noting that nodes may die for various reasons, such as battery death, harsh environment, environmental, or manufacturing deficiencies (low end devices).

When nodes are added to the network, it will be added to suitable clusters depending on its location of deployment. Then, it will be simply treated like the other nodes in the network, where it could operate as either a cluster heads or a sensor node.

Finally, it is worth noting that scalability is achievable when hierarchal routing protocols are used in wireless sensor networks, since the more nodes exist in the network, the more the clusters that can be formulated, where each cluster can operate independently from the other clusters in the network. In addition, the same discussed results will still apply to the scaled version of the network, i.e. the linear decrease of the energy, the effect of the cluster head assignment method on the performance of the network, and the fact that cluster heads are single point of failures for a given round.

C. RL based routing protocol

A similar test as performed for the flat based random routing was performed. The sender node was 0 and the destination was node 10. After iterating through 50,000 solution space, we got this updated Q-table as seen in Fig. 13.

	0	1	2	3	4	5	6	7	8	9	10
0	-100.0	-100.0	-100.0	218.0	138.0	-100.0	-100.0	218.0	174.0	-100.0	-100.0
1	-100.0	-100.0	218.0	218.0	138.0	-100.0	-100.0	-100.0	174.0	-100.0	-100.0
2	-100.0	174.0	-100.0	-100.0	-100.0	-100.0	174.0	-100.0	174.0	-100.0	274.0
3	174.0	174.0	-100.0	-100.0	-100.0	-100.0	-100.0	218.0	-100.0	-100.0	274.0
4	174.0	174.0	-100.0	-100.0	-100.0	-100.0	174.0	-100.0	-100.0	174.0	-100.0
5	-100.0	-100.0	-100.0	-100.0	-100.0	-100.0	174.0	218.0	-100.0	174.0	274.0
6	-100.0	-100.0	218.0	-100.0	138.0	218.0	-100.0	-100.0	-100.0	174.0	-100.0
7	174.0	-100.0	-100.0	218.0	-100.0	218.0	-100.0	-100.0	-100.0	-100.0	274.0
8	174.0	174.0	218.0	-100.0	-100.0	-100.0	-100.0	-100.0	-100.0	174.0	-100.0
9	-100.0	-100.0	-100.0	-100.0	138.0	218.0	174.0	-100.0	174.0	-100.0	-100.0
10	-100.0	-100.0	218.0	218.0	-100.0	218.0	-100.0	218.0	-100.0	-100.0	-100.0

Fig. 13. Updated Q-table

To find the optimal path from node 0 to node 10, the algorithm goes through the Q-table to find the reward using the following method:

1. In row 0, the highest value is for node 3, so the next destination becomes node 3
2. If node 3 is not the destination, so look for next destination
3. In row 3, the highest value is for node 10, so the next destination becomes node 10
4. Node 10 is the destination, so the loop is stopped, and the path is outputted at [0, 3, 10]

Every single simulation will take this optimal path. The number of transmissions from node 0 to node 10 will always be 3. To confirm this, the same test that was done of the flat based protocol was also performed on this, sending 100 messages from sender to receiver 100 times.

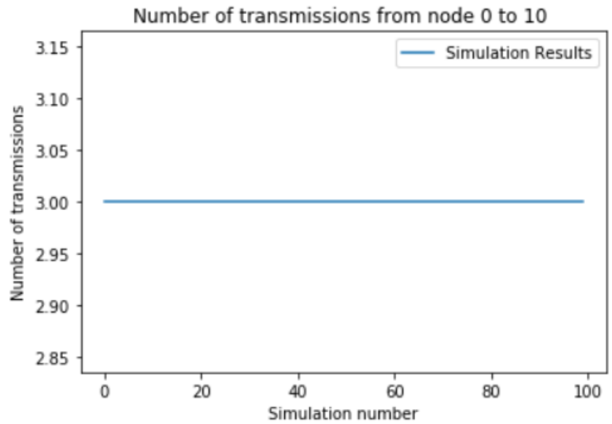

Fig. 14. The number of transmissions the message took to reach the destination of each of the simulation

In Fig. 14, it is seen that every simulation took the optimal path. To study what would happen if one of the node dies, we can remove the column of that node and try to find the optimal path from the updated Q-table. Node 3 is removed, one of the nodes in the optimal path and using the same method, we try to find another optimal path using Fig. 13:

1. In row 0, the highest value is for node 7 (ignoring node 3), so the next destination becomes node 7
2. If node 7 is not the destination, so look for next destination
3. In row 7, the highest value is for node 10, so the next destination becomes node 10
4. Node 10 is the destination, so the loop is stopped, and the path is outputted at [0, 7, 10]

The number of transmissions remain 3, even when there is a node failure in the path of the optimal path. The battery consumption will also be less than the flat based protocol. If 0.5 battery units are used in each transmission, then during the entire simulation knowing the predictable path the message will take, we can calculate the total battery consumption. Only 3 nodes will have used 50% of their battery, while every other node, which was untouched will still be at 100%.

V. Conclusion

When looking at the performance of flat based routing protocols, the implement of this protocol is very simple, which can save costs. This protocol is ideal for smaller networks as there are less possible non-optimal paths to take. However, as the number of nodes increase, this becomes very inefficient. For example, if there are 50,000 nodes, it is possible that the message till take 50,000 transmissions to reach its destination, even if the optimal path is only 2 transmissions. Other than power efficiency, flat based protocols are highly resilient to node failures. Since every node plays the same role, no one node holds enough importance that it will cause the network to fail, unlike hierarchal where a failure of a cluster head can have its packet delivery ratio to drop.

The advantages of hierarchal routing protocols are their scalability, their predictable linear decrease in energy, and their constant end-to-end delay. However, a key disadvantage of hierarchal protocols is the fact that cluster heads are single points of failures. In addition, the method by which cluster head are chosen will impact the performance of the network significantly, especially the packet delivery ration. Thus, an intelligent way of assigning cluster head is required, such as electing the nodes with highest energy to be the cluster heads for their corresponding clusters.

When looking at the performance of RL based routing protocols, they are the most energy efficient since they will take the optimal path. When examining the act of routing, this protocol is more energy efficient, but if the energy consumed while training the network is also included in the overall resource consumption, there will be a rise in consumption. Extra CPU, more memory and storage resources are needed to execute the Q-learning algorithm. This protocol is not as simple to implement as the random routing protocol discussed above. This requires training of the model, exploring possible paths, which has overhead. Like the flat based protocol, this is also resilient to node failures, as the Q-table can be used to find the next most optimal paths.

Each protocol has its advantages and disadvantages. The choice of which routing protocol for a network will depend on the application, and which performance metrics are the most important to optimize to achieve the goals of the network. However, RL based protocols will outperform flat based and hierarchical based in dynamic environments or if only the energy during the act of the messages being passed to nodes is accessed, but due to a more complicated implementation, more resources will be required due to the overhead of the RL algorithm.

VI. Future Work

Routing protocols is a highly researched field, and improvements to these protocols can be made in various ways. For future performance studies, these protocols can be made QoS aware. In this paper, we focused on routing and how the resources are used to have that routing take place successfully. The quality and order of the original message is also a very important aspect in routing which should also be taken into consideration as there are some applications where fragmentation of messages occur. For example, for the hierarchical protocol, the packet delivery ratio was not 100%, and packets were dropped. In QoS aware protocols, the delivery of the message can be assured by resend and acknowledgement methods.

Other Q-learning based routing protocols could be included in the comparison. Also, other routing protocols that fall under the flat and hierarchal routing precools could be considered to have a wider idea about the performance of the different routing protocols will differ.

References

[1] M. Alsheikh, S. Lin, D. Niyato and H. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications", IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996-2018, 2014. Available: 10.1109/comst.2014.2320099.

[2] P. Kumar, T. Amgoth and C. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey", *Information Fusion*, vol. 49, pp. 1-25, 2019. Available: 10.1016/j.inffus.2018.09.013.

[3] H. Mohammed Salman. Survey of Routing Protocols in Wireless Sensor Networks. International Journal of Sensors and Sensor Networks. Vol. 2, No. 1, 2014

[4] J. Yue, W. Zhang, W. Xiao, D, Tang, J. Tang, “Energy Efficient and Balance Cluster-Based Aggregation Algorithm for Wireless Sensor Networks, Procedia Engineering. 2012

[5] Z. Mammeri, "Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches", *IEEE Access*, vol. 7, pp. 55916-55950, 2019. Available: 10.1109/access.2019.2913776.

[6] A. Habib, M. Arafat and S. Moh, "Routing Protocols Based on Reinforcement Learning for Wireless Sensor Networks: A Comparative Study", Journal of Advanced Research in Dynamical and Control Systems, 2019. [Accessed 19 June 2020].

[7] A. Hagberg, P. Swart, and D. S Chult, “Explor-ing network structure, dynamics, and function usingnetworkx,” Los Alamos National Lab.(LANL), LosAlamos, NM (United States), Tech. Rep., 2008.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel,B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer,R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D.Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay,“Scikit-learn: Machine learning in Python,”Journal ofMachine Learning Research, vol. 12, pp. 2825–2830,2011.

[9]]D. Dugmeci, "Finding Shortest Path using Q-Learning Algorithm", Medium, 2020. [Online]. Available: https://towardsdatascience.com/finding-shortest-path-using-q-learning-algorithm-1c1f39e89505. [Accessed: 16- Jul- 2020].