International Journal of Information, Engineering & Technology

Volume 12, Issue 1, PP 1-24, ISSN: 2360-9194, February, 2023
DOI:4272145438121l
Double Blind Peer Reviewed International Research Journal
https://arcnjournals.org
arcnjournals@gmail.com
Journal Series: Global Academic Research Consortium (garc)

Design of an Energy Efficient Clustering Algorithm for Wireless Sensor Networks IoT Devices using Reinforcement Learning

[1]FAGBOHUNMI, Griffin Siji [2]Uchegbu Chinenye E.
[1]Department of Computer Engineering Abia State University, Uturu, Abia State, Nigeria
[2]Department of Electrical and Electronics Engineering, Abia State University, Uturu, Abia State, Nigeria

Abstract: *When designing wireless sensor network for Internet of things (IoT), it is imperative to take into cognizance topology changes resulting from mobility of these devices. A network that uses a flat design where all devices communicate directly to the sink results in a lot of communication overhead which ultimately reduces the network lifetime of the IoT wireless sensor network. For this reason clustering approach is usually employed in recent designs to improve the network lifetime. Recent studies on the clustering approach for wireless sensor networks for IoT usually result in wrong estimation for an optimal cluster size. It is therefore the aim in this paper to design a **W**SN **C**lustering algorithm using **R**einforcement **L**earning (WCRL) which does not necessarily depend on an estimation of optimal cluster size, but rather it dynamically determines the cluster size of the network during topology changes based on its interaction with the environment. This interaction is regularly updated using the reinforcement learning paradigm. Optimal cluster size estimation has the advantage of enabling equitable distribution of energy consumption among all clusters in the wireless sensor network so that energy consumption is not skewed to particular clusters. The purpose of this work is to reduce energy required in assigning cluster head as opposed to the repeated cluster head assignment procedure used in the previous algorithms. This approach is based on reinforcement learning where all nodes (WMNs) in a cluster jointly elect a cluster head among themselves. The WCRL clustering algorithm proposed in this paper is compared to other recently developed algorithms like MemeWSN and PBC-CP . Simulation experiments showed that the proposed algorithm performs better than the compared algorithms using performance metrics such as control packet overload, number of clusters, cluster formation rate and lifetime of clusters by 13%, 24% 17% and 25%respectively.*

Key words: *Service delivery, cluster size, IoT, genetic algorithm, reinforcement learning, Q-learning*

1. Introduction

IoT has become widespread in the communication industries due to the importance of communicating between everyday devices using appropriate algorithm in order to send information gathered from this devices to a sink where computation of data is done for decision making. Applications of WSN in IoT includes rescue mission in times of war, target monitoring, monitoring of flood and disaster management. In these applications, clustering becomes inevitable when the number of IoT devices involved in the network becomes very large. In such cases the use of a single cluster leads to increased overhead due to the number of transmission and redundant transmissions required to send data packets to the sink. It has been proven in (Gupta and Kumar 2010) that when the number of IoT devices involved in a network with a single cluster or a flat routing algorithm is n, the time complexity of the protocol is O(n^2). Subsequently, as the number of IoT devices involved in the network increases, the routing overhead increase in proportion to the square of the number of devices. Other routing techniques like the reactive routing algorithms cause a delay in the route setup phase especially with an increase in the number of devices. Therefore in order to accomplish optimum service delivery in IoT WSN when many devices are involved, it is necessary to employ the clustering approach to WSN routing (Belding-Royer 2012).

The process of assigning optimal cluster head in cluster based routing in WSN is an NP-hard problem, hence appropriate routing algorithm meant to efficiently communicate data packets through the various IoT devices to the sink must be designed (Perkins 2011). The aim of clustering algorithm is to divide the network topology into different segments, so that each segment is requires to have a cluster head. The purpose of the cluster head is to aggregate data packets from the wireless IoT devices in each segment. A number of compression algorithms may be applied on the aggregated data so as to reduce data redundancy before sending such data to the sink from individual cluster heads. The importance of clustering becomes evident when the number of IoT devices in the WSN becomes very large i.e. in hundreds or thousands. The application of flat routing protocol results in high redundancy data communication to the sink. This high redundancy results in increased overhead which ultimately results in data collision and reduction in both throughput and lifetime of the WSN IoT networks (Basagni and Chlamtac 2007).The problem of scalability in WSN IoT becomes more complex when these IoT devices are mobile in which case mobility consideration must be included in the protocol design.

One major challenge in clustering algorithm is the election of appropriate cluster head (CH) during each round of clustering so as to minimize the energy consumption resulting from such procedure. The two most common procedure used involve randomly selecting a cluster head from among the IoT devices in the network while other IoT nodes join the cluster head requiring a minimum number of hops, in this case cluster heads are assumed to have specific distance from each other, so that cluster heads are not skewed to certain areas in the network. Alternately the IoT devices can first be divided into clusters before the election of cluster heads within each cluster. This is particularly used when IoT devices of similar properties are to be assigned same cluster. Different techniques has been employed in electing optimal cluster heads in each cluster of WSN IoT, this includes genetic algorithm, artificial neural network and optimization using particle swarm (Shah et al 2020). Particle swarm optimization (PSO) is not ideal for the election of cluster heads when the number of IoT devices are large or the network topology changes, this is because of increased overhead resulting from synergy agreements between all the agents. Application of bee colony optimization does not incorporate the addition of new IoT devices in the network design, hence the presence of such may cause the algorithm not to converge resulting in high energy drainage. The use of genetic algorithm will not be good to apply in clustering algorithm for certain categories of WSN IoT especially where the properties of the WSN IoT devices vary considerably because of high communication overhead involved in chromosome update.

The use of genetic algorithm is not suited for large WSN IoT network because the combined operation of selection, mutation and cross- over will be too cumbersome for the energy constrained sensors networks and will lead to quick depletion of energy resulting in reduced network lifetime. This paper proposes a clustering algorithm based on Q-learning technique. Here cluster formation needn't be done repeatedly at regularly spaced intervals, but only when the Q-values of the cluster head is below a threshold. This means that all wireless mobile nodes (WMNs) within a cluster will be involved in selecting a cluster head which leads to greater synergy among the cluster members. Secondly, the initial cluster head is not selected randomly, but rather based on initial parameters of remaining energy of the WMN and distance or hop count to the sink. This enables the protocol to converge faster than when random selection of initial cluster head is done.

The rest of the paper is organized thus, section 2 discusses the works of previous researchers in the area of clustering in WSN, in section three a model of the Q-learning technique used in this paper is illustrated, section 4 shows the results and analysis of the protocol used in this paper in comparison with other two state of the art clustering protocols and finally section 5 concludes the paper stating directions for future research work

2. Literature Review

The purpose of any clustering algorithm in WSN is to (i) reduce the overall energy depletion in the WMNs, (ii) evenly spread the energy consumption across the various clusters and (iii) prolong the lifetime of the network. A good number of clustering algorithm in WSN has been proposed in literature, in the work of (Azni et al 2021) the distributed cluster head scheduling was proposed (DCHS). It employs the distributed approach in electing cluster head. In DCHS the network is divided into clusters before an initial cluster head is selected at random from among the cluster members. The shortcoming in the approach is the non-convergence or long convergence time of the clusters. Also the mobility of the WMNs and the possibility of large number of neighbour nodes were not considered in its design. The issue of a secure algorithm as well as equitable distribution of energy consumption across all clusters in the network was not also given a consideration.

(Deb et al 2020) also designed a clustering protocol where in the initial phase, cluster heads are elected at random and subsequently cluster heads are elected based on the remaining energy of the WMNs within the cluster. The shortcoming in their protocol remains the non-convergence or very late convergence of the protocol in finding an optimal cluster size for the network. Mobility and equitable spread of energy consumption across all clusters was not also considered. In the work of (Kannan and Sree 2019), the researchers formed clusters based on the different categories of data required for sensing, in such a way that each cluster is only allowed to access data of the same type. This design leads to increased throughput but at the cost of increased communication overhead resulting from segregation of data and longer communication paths due to the fact that WMNs sensing like data may be far from one another. This scenario will lead to quick energy depletion in the network and reduced network lifetime. In the work of (Behera et al 2019), the speed of mobility of the WMN was used in electing cluster heads with nodes with lower mobility having higher chances of being cluster heads. The shortcoming with their approach is in a scenario where most WMN has unpredictable mobility, such scenarios leads to frequent changes in cluster head and topology. Which may lead to network disconnection. This disconnection will lead to network death. In the work of (Fagbohunmi and Eneh 2015) a cluster head is elected on the basis of a prediction function of the mobility of a node. This ability is used to derive a predicted location of a node with respect to the position of its long term neighbour, with the assumption that a node that moves in direct correlation to the movement of its neighbour nodes will form a stable topology. In this vein a cluster head is elected with a node that has like mobility pattern with its neighbour. The idea here is that such node union will from stable clusters with minimal energy required for packet transmission in terms of node mobility. Their model was able to perform well in mobile assisted nodes irrespective of the speed of node mobility. The shortcoming of this design is of course in a topology with random behaviour, nodes will not be formed with any degree of consistency, such scenario may lead to multiple cluster head election with high communication overhead. In the work of (Venkanna and Leela 2020), the authors proposed a genetic algorithm model that is able to balance the energy consumption across the various clusters n the network. Their work also included a situation where the network topology varies randomly in such case a multi-population model was designed. The idea here was to use a multi-function parameter to determine WMNs belonging to the same cluster. However the main aim of their work was to develop a clustering protocol that can produce high data throughput irrespective of both WMN mobility and random topology variation. The shortcoming was the high data communication overhead involved in the formation of multi-population GA, which leads to quick energy drainage and short network lifetime.

In (Konstantopoulos et al 2017) a clustering optimization protocol was designed that uses a fitness function to calculate the optimal cluster size in different topology situations. The idea in their work was to design a protocol that is able to reduce total packet communication by reducing the number of clusters in a given network size. This will ultimately lead to reduced energy consumption because the network having optimal number of cluster heads will be able to optimally aggregate the WMNs reading and send such data to the sink. Also the optimal cluster size will be able to equitably spread the energy consumption across the various clusters of the network. However the shortcoming is the cumbersome nature of the protocol where a multi-objective cluster organization of the network must be ensured. This complex cluster formation based on resource allocation among the nodes lead to increased communication overhead with high likelihood for data collision. This will eventually reduce the network lifetime. (Cheng et al 2020) designed a clustering algorithm by electing cluster heads as those nodes with lower hop counts to the sink. WMN mobility was also considered in their design, however the design was not resistant to random topology changes. Also there is high likelihood of cluster heads been skewed to certain parts in the network. Such scenario will lead to sink node isolation especially for cluster head that has much data to forward to the sink due to its local optima functionality. In (Ali et al 2018), a Proficient Bee Colony Clustering protocol was designed (PBC-CP). The parameters used in the election of cluster heads were the remaining energy of the WMNs, number of hops to the sink and the number of alternative route from a node to the sinks. A node having a higher number of neighbour nodes to the sink will be chosen ahead of one that has fewer numbers of neighbour nodes. The shortcoming in the PBC-CP protocol is its lack of compatibility to mobile nodes, in which case it doesn't support node mobility especially in case where the movement of a node is at variance with that of its neighbour. Such scenarios lead to unstable cluster formation with the need for frequent cluster head election procedures. Such frequent re-election of cluster heads quickly drains the energy of the network. In the work of (Fagbohunmi and Eneh 2019) a secured routing protocol was designed based on Q-learning protocol. The cluster head was selected based on the location of the nodes. The preferred nodes are those with high Q-values. The idea of the secured routing protocol was to both conserve energy consumption in the network and route through secured paths by using nodes with high Q-values. The protocol does not require the repeated cluster head election procedure as is normally the case with some clustering algorithm. The shortcoming in this approach is the fact that static configuration of the cluster head was assumed which means that the protocol will fare badly under the condition of mobility of the cluster head. In the work of (Pathak 2020), a genetic algorithm approach was proffered where a fitness function was used to calculate the optimal cluster size for a given network area. Subsequently a bypass (gateway) node was used to transmit data packets from one cluster to another (inter-cluster communication). The aim of this gateway node is to reduce the amount of data packets transmitted by the cluster head as some may be far from the sink. It should be noted in the protocol that the CH election procedure is not based on the position wherein the cluster heads can be evenly placed but rather according to the node's transmission energy. The researchers went on to devise means of protecting the identity of the cluster heads as attackers can easily target them through the gateway nodes. In order to forestall such scenario, they proposed a multi-objective function using different parameters to elect the CH according to the population so that the identity of the cluster head is not straight forward. However the shortcoming of this protocol is the cumbersome nature of the multi-objective function. The computation required to elect cluster head will be too high and most importantly may cause delay in electing CH. This will eventually lead to delay in convergence time of the cluster formation leading to increased communication overhead.

In the Q-learning WSN-IoT clustering algorithm proposed in this paper, the election of the cluster head is not based on any particular fitness function. Rather the sinks are expected to flood a control packet to all WMNs in order to get their routing information, The routing information is based on parameters with includes number of hops to the sink, number of neighbour nodes that can be used to reach the sink (alternative routes) rate of energy consumption of the WMN and the remaining energy in the of the WMNs. These parameters are used to calculate the Q-values of the individual WMNs. The advantage of Q-learning is that it does not involve high computation nor is an explicit model of the network topology required.

3.1 Reinforcement Learning

Reinforcement learning is a machine learning technique used to model an intelligent agent which interacts with its environment in order to maximize a cumulative reward (Vimalarani et al 2017). It employs Markov decision process to select actions from a given set of alternatives. Rewards are assigned depending on the action taken at any given point in time. The aim of the agent is to learn the optimal action to take from any state so as to maximize a cumulative reward as it transits from the current position to its destination. In order to achieve its aim, the agent must repeat various permutations from the available actions so as to know the one that will lead to maximum reward.

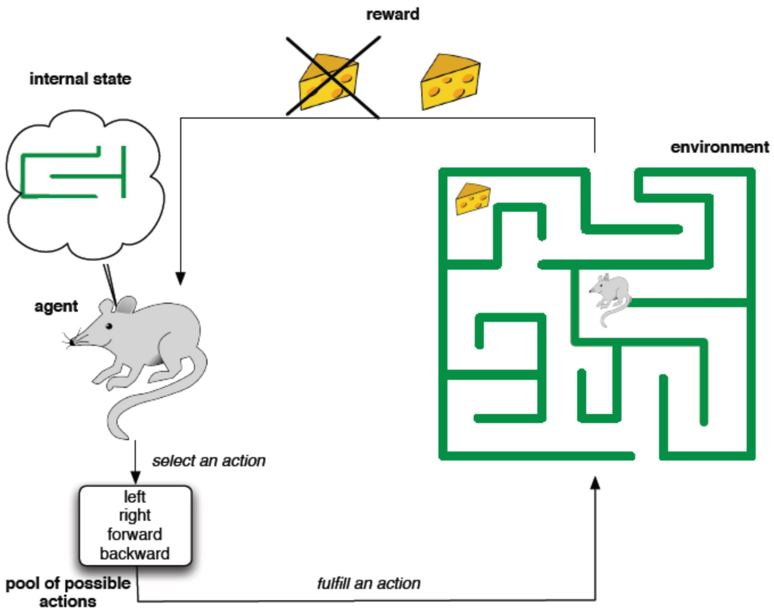

Figure1. A Representative model of Reinforcement learning.

An example shown in figure 1 is that of a mouse attempting to traverse a maze environment. The mouse can select an action among the alternative present and in accordance with its previous knowledge of the maze environment. Every action taken will result in a given reward. Generally a positive reward is assigned when the goal of the agent is fulfilled and negative reward is given when the goal is not been fulfilled by the action taken. The advantage of using reinforcement learning is its low computation and medium memory requirements. In this paper a wise idea will be used to reduce the memory requirements needed to store all routes in the routing table of the agent. The reinforcement option used in this paper is Q-learning.

3.2 The WSN IoT Q-Learning Model

This section describes the steps required in allocating WSN IoT devices into clusters. It should be stated here that the clustering protocol proffered in this paper does not attempt to define an optimal size for a cluster, however the cluster size varies according to the response of the agent in relation to its interaction with the environment hence the protocol is adaptive.

The aggregation (clustering) model defines a role free cluster head election technique thereby resulting in an equitable distribution of the cluster heads in the network. Being role free means that, there are no particular or definite properties expected of the cluster head. The election of nodes to be used as cluster head is dependent on the agent's interaction with the environment culminating from the computation of Q-values for the WMNs. This was used in order to reduce the overhead caused by in-networking communication usually resulting from the computation for the election of cluster heads at every round of packet routing (used by previous researchers) as well as quick depletion of WMNs closer to the sink as they are involved in forwarding all sensor readings to the sinks.

In contrast to compared protocols, it does not repeatedly compute the optimal cluster head (in terms of cost) at every round of routing, and to inform cluster members about it, but incrementally *learns* the best behaviour without knowing where and who the real cluster heads are. The result of this is that many WMNs will initially act as cluster heads resulting in the formation of many clusters. However as shall be seen later, the communication overhead resulting from this will be minimal compared to the regular computation for cluster heads at the beginning of every round. After this initial operation, the real cost or each WMN to transmit data packets optimally to neighbour nodes will be learned and cluster heads will be formed in fairly equal partition of the network area.

This feature is achieved as follows: The cluster formed in the protocol is a variable size square cluster between 80 -110m. Each WMN attempts to transmit its data packets directly to the sink in the network. In order to achieve this, a WMN will either try to act as a cluster head or transmit its data packet to a better suited neighbour WMN using the computation of the Q-value using equation 4 (to be defined later in this section). In the cluster-head routing scenario, every WMN is an independent learning agent, and its actions includes energy efficient routing using different fit neighbours for the next hop toward the cluster-head. The cluster-head is defined as the node in the cluster with the best (lowest) routing cost to the sink. This cluster-head is elected based on the computation of Q-values described later in this section. The following provides the parameters used for the Q-learning clustering solution.

The Q-learning model used in this paper consists of the following:
Agent states: This defines the states an agent (WMNs) can be at any instance in time. The agent can be either in the active, listening or the sleep mode. The active state refers to when the WMN is transmitting or receiving data packets, The listening state is when the WMN waits for data packets from neighbour nodes and the sleep mode is when the WMN is inactive.. For routing to the sink, the state of an agent is defined as a tuple {S_p, $routes\ _{S_p}^{N}$ }, where S_p *is* the sink the packet must reach and $routes\ _{S_p}^{N}$ is the routing information about all fit neighbouring nodes N that will lead to the sink.

Actions: This is the model of the transition from a state to the next available state. It should be noted here that action is only taken when the WMNs are in either active or listening state. The WMN can take any of the following two actions, i.e. transit to a neighbour node or stay in its present state while waiting for data packet from a neighbour node, in this case since there are no transition to a new state, there will not be any improvement of the Q-value. In the model, an action can include transmission of data packets to one or more different fit neighbours as next hops (as each node is expected to have more than one neighbour). The action is defined by *H*= (n_i, *S*) which defines a single fit neighbour n_i where i can be from 1 to 3 and the destinations *S* indicating that neighbour n_i is the intended next hop for routing to destinations S. The value of action is as shown in equation 1

$$\text{H} = \left(\textstyle\sum_S hops_S^{n_i}\right) \qquad (1)$$

where $hops_S^{n_i}$ are the number of hops to reach the destination S using neighbour n_i, This translates to the transition equation shown in equation 2

$$\text{E}(s') = [p(s_{t+1}|\ s_t,A_s)] \qquad (2)$$

The number of alternate paths to the sink through a given neighbour node is given by equation 3

$$\textstyle\sum_{s=1}^{k}[ps_{t+1}\ |s_{t,}A_s] \qquad (3)$$

where k is the number of neighbour nodes. In this paper, the upper bound for the number of neighbour nodes is three (3). The reason for this will be stated later in the section.

Q-Values. These represent the goodness of actions and the goal of the agent is to learn the *actual* goodness of the available actions. Here as opposed to the original Q-learning, which randomly initializes Q-Values, here Q-Values will be bound to represent the real cost of the routes, for example, in this paper the cost function is the combination of five parameters defined by equation 4. To initialize these values, a more sophisticated approach will be employed, which gives an estimate of the cost based on the individual information about the involved neighbours and sink. This approach significantly speeds up the learning process and avoids oscillations of the Q-Values as is the case with most Q-learning model. The Q-value of an action is depicted by $Q(a_i)$ is as shown in equation 4

$$Q(a_i) = \left(\textstyle\sum_S hops_s^{n_i}\right) + \beta\text{Re} + \psi\text{T} + \acute{\eta}\text{Mo} + \lambda\text{Rt} \qquad (4)$$

where $hops_S^{n_i}$are the number of hops to reach a destination S using neighbour $n_{i,,}$ $\beta = 5^{(1 - x)}$ where x is the percentage of remaining battery energy of the WMN, Re is the value of remaining battery energy of the WMN, $\psi\text{T} = 5^{(3-y),}$ where y is the number of neighbour nodes, $\acute{\eta}\text{Mo} = 5^{\frac{t}{100}}$, where t is the speed of the WMN and $\lambda\text{Rt} = 5^{(1-\frac{z}{5})}$, where z is the transmission energy of the WMN in μJ. The exponential values used for the different parameters will be explained in the next paragraph.

Equation 4 consists of five parts, the first part ($\sum_S hops_s^{n_i}$) of the equation accounts for energy efficiency, it defines the number of hops to reach the sink. The minimum number of hops is selected after all alternate paths to the sink have been computed, this results in minimizing data packet transmissions in the network. The second parameter (Re) is the routing through the WMNs with the maximum remaining battery energy, this is necessary to avoid routing through very low powered nodes even when such nodes have shorter hop counts to the sink. The third parameter (T) consists of routing through WMN with higher number of neighbour nodes as opposed to a node with just a single neighbour node. The importance of this is that routing through a WMN with higher number of neighbour node will improve transmission throughput as the alternative routes will enable alternate paths of transmission in the event of link or WMN failure involving one of the neighbour nodes. In this paper, an upper limit has been placed on the number of neighbour nodes. The limit is set to three (3), this is important so as to limit the number of state-actions transitions that can be stored in the routing table. This is because an increase in the number of neighbour nodes to a WMN leads to a polynomial increase in the state-actions transitions, which invariably leads to more data being stored in the routing table. It is common knowledge that the embedded sensors in most WMNs are memory and energy constrained, so innovative techniques must be applied in order to optimally store data in its memory. Relaxing the number of neighbour nodes to any value will result in a very large state action space resulting in increased number of entries into the routing table, which will overwhelm the memory constrained embedded sensors of the WMNs. The fourth parameter of the equation (Mo) chooses a WMN with low mobility as cluster heads in preference to those with high mobility. This is because the mobility of a WMN will result in erratic change in the cluster formation, an incidence that leads to topology change.. Therefore it is always preferred to elect WMNs with low mobility as cluster heads. The fifth parameter of the equation (Rt) elect cluster heads with high transmission energy which in turn has wider coverage area. This part of the equation results in lower number of cluster heads being formed which ultimately leads to lower number of transmitted data packets in the network.. These five elements of the equation are weighted with specific scalar values as explained in the next paragraph. The weighted values (WV) for each parameter grows exponentially with (i) increase in the number of hops to the sink, (ii) decreasing battery levels, (iii) decrease in the number of neighbour nodes (iv) decrease in the transmission energy and (v) increase in mobility.

The consequence of this is that these parameters are weighted with different exponential functions. In the case of routing through WMN with higher remaining battery energy, the exponential function is $5^{(1-x)}$ where x is the percentage of remaining battery energy of the WMN. This means that a WMN with full energy level will have the lowest weighted value (WV) of 1, ie $5^{(1-1)}$or $5^{0,}$ while a WM with 70% remaining battery energy will have a weighted value of 1.62, i.e. $5^{(1-0.7)}$ and the WV of a WMN with 80% remaining battery will be 1.38. the purpose here is to decrease the Q-value with increase in the percentage of remaining battery energy. In the case of the number of neighbour nodes, the exponential function is $5^{(3-y),}$ where y is the number of neighbour nodes. The idea here is to fix the highest number of neighbour nodes attainable for a WMN to three (3). This is necessary because an increase in the number of neighbour nodes increases polynomially the possible ways of routing data packets to the sink resulting in late convergence of the protocol in electing cluster heads. With this model, a WMN having number of neighbours as three (3) will have the lowest weighted value of 1, i.e. 5^0. while the weighted value of a WMN having one neighbour will be 25 (5^2). In the case of mobility The weighted value function is given by $5^{\frac{t}{100}}$. It should be noted here that that speed of WMNs used in the simulation was in the range of 0 km/h – 100 km/h, therefore a WMN with no mobility i.e. speed of 0km/h will have the least WV of 1 i.e. $5^{\frac{0}{100}}$, $5^0 = 1$ while a WMN with speed of 40km/h will have a weighted value of 1.90, i.e. $5^{\frac{40}{100}} = 5^{\frac{2}{5}}$ and a WMN with a speed of 75km/h will have a weighted value of 3.34 i.e. $5^{\frac{3}{4}}$. This shows that the weighted value increase exponentially with an increase in mobility, giving preference to WMN with low mobility. Finally in the case of the transmission energy a WV is assigned such that a WMN with high transmission energy will have a low weighted value. It is assumed here that a WMN with the highest transmission energy is 5μJ with a WV of 1. The transmission energy of every WMN is placed as a fraction of 5μJ. Therefore the exponential function used for the computation of WV for the WMN transmission energy is $5^{(1-\frac{z}{5})}$, where z is the transmission energy of the WMN in μJ. The effect of this is that as the transmission energy of the WMN becomes lower its WV increases thereby reducing its probability of being elected a cluster head as the computed Q-value will be higher. The reason why exponential function rather than a linear function was used is because the tendency here is to give high weighted value to parameters at the upper spectrum of the parameter range. This means that there will be a lower difference in weighted value (WV) for WMN with percentage of remaining energy of battery in 90% - 70% $5^{1-0.9} - 5^{1-0.7}$(1.62 – 1.17 = 0.45) than to a WMN whose difference in percentage of remaining energy of battery is 30% and 10%. $5^{1-0.3} - 5^{1-0.1}$ (4.26 – 3.09= 1. 17) The essence here is give lower differential value of WV for WMN to be skewed towards the upper spectrum of the parameter value so as to discourage electing cluster heads using nodes in the lower spectrum of the parameter value.

Q-Value Update. This denotes the reward values for each action taken in the environment for a particular state. In this case, after the sink sends the announcement control packet to the WMNs in the network, each fit neighbour to which a data packet is forwarded sends the reward (Q-value) as feedback with its evaluation of the goodness to the sink. The new Q-Value of the action is as shown in equation 5

$Q_{new}(a_i) = Q_{old}(a_i) + \gamma (R(a_i) - Q_{old}(a_i))$ (5)

Where $R(a_i)$ is the immediate reward value and γ is the learning rate of the algorithm. $\gamma = 1$ is used here because the initial Q-value represents an upper bound of actual value (i.e. maximum Q-values corresponds to WMN with lowest energy as explained earlier in the section. This will be the initial Q-value to be used in the computation for the sink announcement to all WMNs to reach selected destinations through all neighbours K and hence it is expected to reduce during learning. A lower learning rate (between 0 – 1) is usually used with randomly initialized Q-Values. This causes the Q-value to oscillate heavily in the beginning of the learning process. Therefore, with $\gamma = 1$, the formula is as shown in equation 6

$Q_{new}(a_i) = R(a_i)$ (6)

which directly updates the Q-Value with the reward.

Reward function. This is the downstream WMNs (i.e. nodes farther from the sinks) opportunity to inform the upstream neighbours of its actual cost for the requested action. Hence, when calculating this, the node selects its lowest (best) Q-Value for the destination node and adds the cost of the action itself. This is shown in **equation 7,**

$R(s,a_i) = c_{a_i} + \overset{min}{a}(Q)(a)$ (7)

where$c_{a_i}$is the action's cost, as shown in equation 8

$c_{a_i} = 1 + \beta Re + \psi T + \acute{\eta} Mo + \lambda Rt$ (8)

This is because as the node transits to its neighbour, the downstream WMN increment the hop count by 1 and subsequently update the action cost with the number of neighbour nodes, the remaining battery energy, WMN mobility and the transmission energy of the WMN. The flowchart for the Q-value update procedure which is necessary for the election of cluster heads is shown in figure 2.

Policy (Model): The Q-learning model is then modified as shown in equation 9

$V(s) = \min [R(s,a_i) + \sum_{s' \in S} P(s'|s,a)V(s')]$ (9)

*w*here $R(s,a_i)$ is the current estimate i.e. current reward value, $V(s) = \min Q^*(a)$ i.e. the value function is the new estimate i.e the minimum Q-value of all routes (considering all alternate neighbour routes) starting from state (s) and action a to the destination, γ is the learning rate,$\gamma = 1$ here, hence it is omitted in the equation.

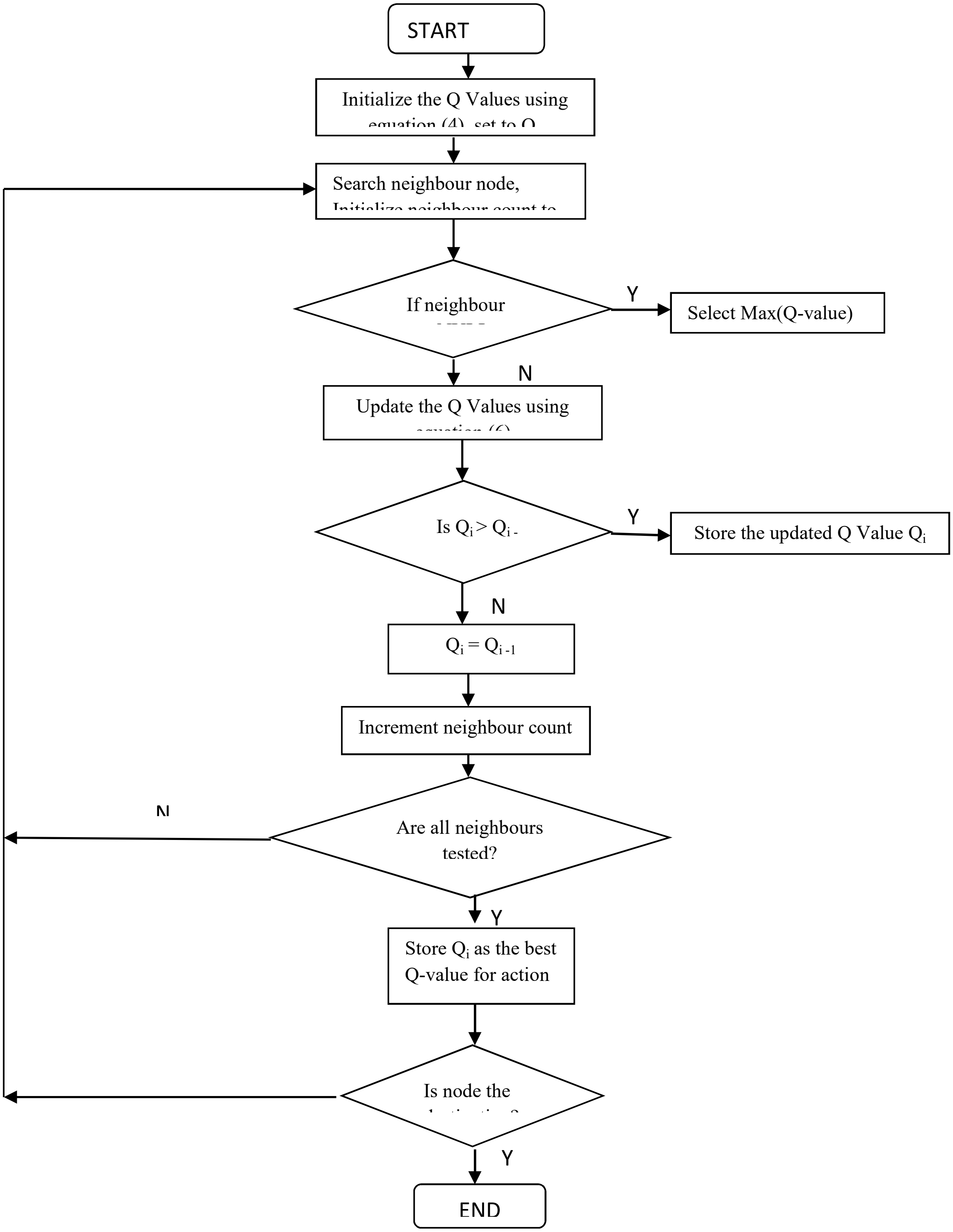
START
Initialize the Q Values using equation (4), set to 0
Search neighbour node, Initialize neighbour count to
If neighbour
Y
Select Max(Q-value)
N
Update the Q Values using equation (6)
Is $Q_i > Q_{i -}$
Y
Store the updated Q Value Q_i
N
$Q_i = Q_{i-1}$
Increment neighbour count
N
Are all neighbours tested?
Y
Store Q_i as the best Q-value for action
Is node the destination?
Y
END

Figure 2 Flowchart Q-Value update procedure

3.3 WSN IoT Clustering Protocol

The sequence of the clustering protocol is as follows: At the start of the experiment the control packet from the sink (sink announcement) will propagate the number of hops to the sink from a particular WMN. This is achieved as follows:

(i) The sink broadcasts the sink announcement to all the WMN. It is assumed here that the sink is not power constrained so that its transmission radius covers the entire WMNs making up the network.

(ii) The WMNs closest to the sink (neighbour) replies with hop count of 1

(iii) The set of WMNs closest to the sink will further propagate the control packet to its neighbours, this will increment the hop count to 2

(iv) This procedure will continue iteratively until the control packet gets to the source.

It should be noted here that the hop count increment iteratively until the control packet gets to the source. The beauty of this protocol is that a WMN need to only have the routing information of its neighbours. This reduces the amount of data stored in the routing table as information is not required from all multicast WMNs in the coverage area of a given node. The Q-value used at the commencement of algorithm represents the upper bound value as we select the WMN with the lowest energy in the network, this procedure also applies to all the five parameters used or the Q-value computation. As a result, during learning, Q-Values is bound to decrease and the best actions will be denoted with small Q-Values. The flowchart for the broadcast of sink announcement procedure is shown in figure 3.

The following parameters were used to compute the energy consumption during simulation:

P_{mod}: This defines the energy for transmitting one bit of data

P_{mul}: This defines the energy to transmit 1 bit of data over a radius of 1m.

P_{comp}: This defines the full energy of the embedded sensor module in the WMN.

P_{trans}: This defines full energy required for transmitting data.

P_{PRO}: This defines the full energy of the WMN.

$P_{PRO\ dat}$: This defines the full energy required by the WMN for data processing.

$P_{PRO\ sig}$: This defines full energy required by the WMN for signal processing.

P_{mrd}: This defines full energy required by the WMN for accessing data from its memory.

P_{mwr}: This defines full energy required by the WMN for data storage.

P_{rab}: This defines the full energy for data transmission.

P_{csen}: This defines the minimum energy required for sensing data.

B: This defines separately the data and signal packet size.

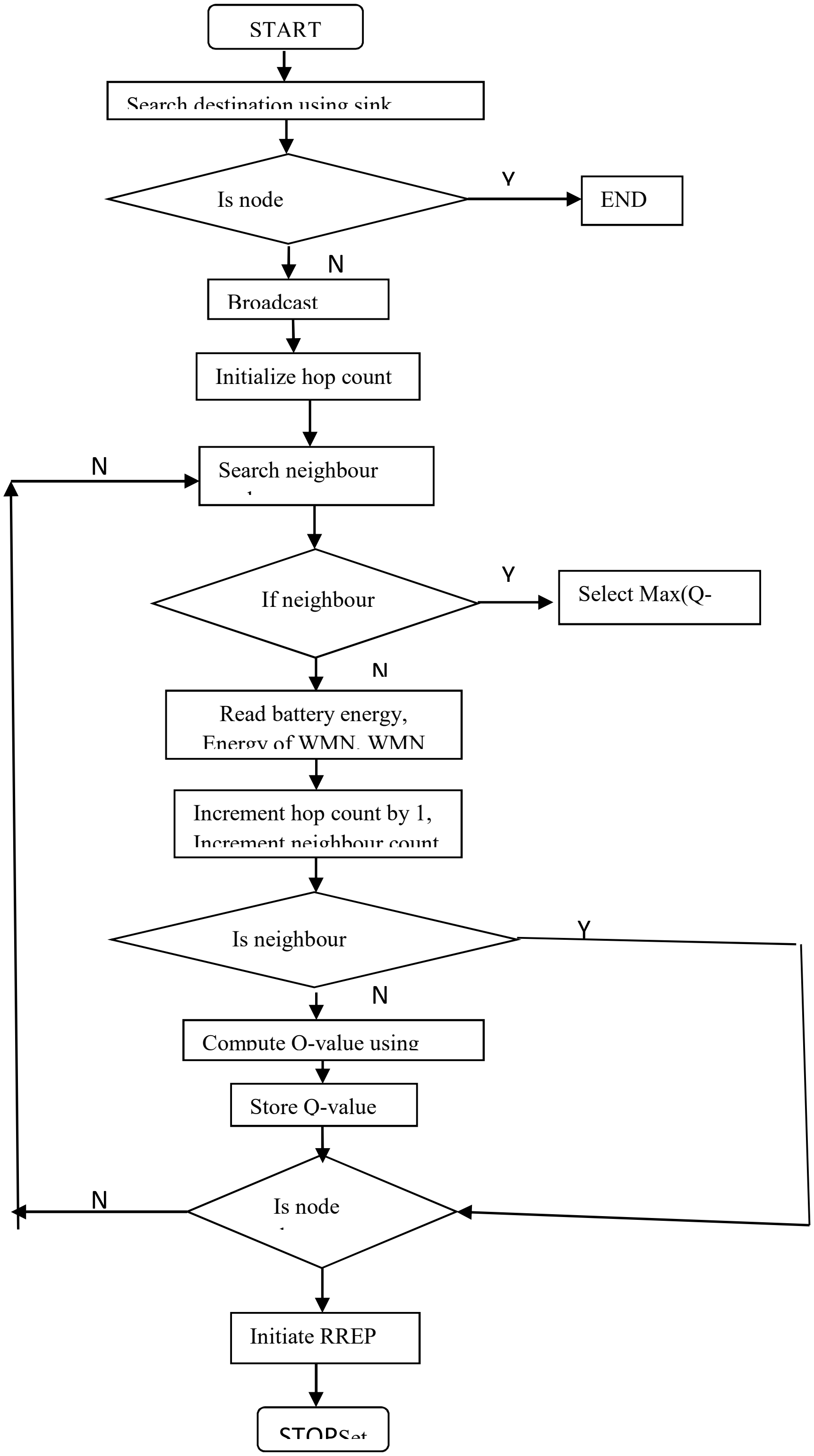

Figure 3 Flowchart for sink announcement

3.4 Theoretical Energy Calculation Used for Simulation

This is the definition of values used for the protocol parameters or the computation o the initial Q-value.

P_{mod} = 145 nJ/ bit (10)
P_{mul} = 300 pJ/ bit/m^2 (11)
Size of data packet: = 1024bits (12)
Size of signal packet = 128 bits = 16 bytes (13)
Rate of data transmission = 1024 bits /sec (14)
Ideal radius 'r' <= 100 m (15)
$P_{comp} = P_{trans} + P_{PRO} + P_{sen}$ (16)
$P_{trans} = TRX_{info} + TRX_{info} + TRX_{sig}$ (17)
$TRX_{data} = E_{com} * B + E_{mul} * B * r^2$ (18)
$TRX_{sig} = P_{com} * B + E_{mul} * B * r^2$ (19)
$TRX_{info} = P_{com} * B$ (20)
$TRX_{sig} = P_{com} * B$ (21)
$P_{PRO} = P_{PROinfo} + P_{PRO\ sig} + P_{mrd} + P_{wr}$ (22)
$P_{PROinfo} = P_{com} * B$ (23)
$P_{PRO\ sig} = E_{Pom} * B$ (24)
$P_{mrd} = 4 * P_{com} * B$ (25)
$P_{wr} = 0.75 * P_{com} * B$ (26)
$P_{rab} = P_{com}$ * size of data packet (27)
$P_{csen} = P_{csen} * 3/4$ (28)
Full energy of 100 nodes $= P_{mod} * 100$ (29)
Full energy of network for 3600 s = Full energy for 100 WMNs
within the same time interval (30)

Parameter values for TRX data, TRX sig, TER_{info} and $TER_{sig}$is computed using Equation (18) to Equation (21) respectively. This is shown below.

$TRX_{info} = 40 * 10^{-8} *1024 + 0.15*10^{-8} *1024*45^2 = 350$μJ/message
$TRX_{sig} = 40 * 10^{-8} * 128 + 0.15 * 10^{-98} * 228*45^2 = 30.5$μJ/message
$TER_{info} = 45 * 10^{-8} *1024 = 45$ μJ/message = 0.08 μJ/bit
$TER_{sig} = 45 *10^{-8}* 128 = 4$μJ/message
Using Equation (17), the energy of the transceiver is computed as:
$P_{cen} = 350+30.5+45 + 4 = 429.5$ μJ/message
The values of the parameters such as $P_{PRO\ info}$, $P_{PRO\ sig}$,P_{mrd} and P_{wr} can be computed using Equation (22) to Equation (25). This is shown as follows:
$P_{PRO\ info}$ = 1024 bits/message * 45 nJ /bit
$P_{PRO\ sig}$ = 128 bits/message * 45 nJ /bit = 4 μJ/ message
P_{mrd}= 3 * 45 nJ /bit * 16 bits = 2 μJ
P_{wr}= 0.75 * 45 nJ /bit 16 bits = 0.5 μJ
The full energy used up by the WMN is computed using Equation (22) as follows:
$P_{PRO} = 45 +4 +1.2 + 0.8 = 51$ μJ

In this paper, the network of WMN is modelled as a graph $G = (V, E)$ where each WMN is a vertex v_i and each edge e_{ij} is a bidirectional wireless communication channel between a pair of WMNs v_i and v_j. Each source node $s \varepsilon V$ and the sink is designated S. Optimal routing to the sink is defined as the minimum cost path starting at the source vertex s, to the sink S. This cost path is a function of the five parameters defined earlier This path is actually a spanning tree $T = (V_T, E_T)$ whose vertexes include the different sources and the destination sink S. The cost of a tree T is defined as a function over its nodes and links $C(T)$. This is defined using the Q-value model in equation 4. Figure 4 depicts the network configuration.

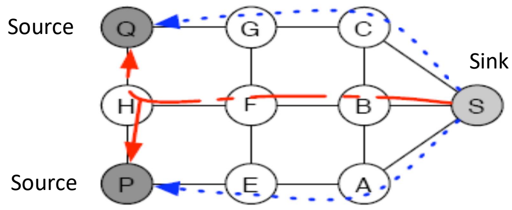

Figure 4a Network configuration

Routing table: Sink S

Neighb A	Source P	3 hops
	Source Q	5 hops
Neighb B	Source P	4 hops
	Source Q	4 hops
Neighb C	Source P	5 hops
	Source Q	3 hops

Routing table: WMN A

Neighb S	Source P	4 hops
	Source Q	4 hops
Neighb B	Source P	4 hops
	Source Q	4 hops
Neighb E	Source P	2 hops
	Source Q	4 hops

Fig 4b Sample routing table for WMN A and Sink S

3.3 WCRL Protocol Design

The protocol used in this paper WCRL (**W**MN **C**lustering using **R**einforcement **L**earning) is built on the modified Q learning model as presented in Section 3.2. The modification arose as a result of a minimized function used in our model as opposed to the maximization used in the original Q-learning model. The protocol is divided into four main steps. Its full pseudo code is shown in figure 5. The first step (lines 2 – 6) in the protocol is the sink announcement phase, here the sinks sends a control packet to all the WMNs in the network. The purpose of this action is to compute the Q-value from any given WMN to the sink using equation 4 , this computation includes the five parameters of (i) number of hop count required to route from the sink to all the nodes in the network, This is achieved by the sinks sending a control packet through its immediate neighbours to all the nodes in the network (ii) the residual energy of the WMN, (iii) the number of neighbour node(s) of the WMN (iv) the transmission energy of the WMN, and (v) the mobility of the WMNs .

At the initial stage the hop count is initialized to 0 meaning the hop count of a sink to itself is 0. Subsequently the sink then propagates the control data packet through its immediate neighbours. It has been stated earlier that the upper bound value of the number of neighbour to a WMN is three (3), this is aimed at reducing the state action space in the routing table so as to enable the protocol converge its Q-value in finite time. Each step of propagating through successive immediate neighbour of a WMN is followed by the hop count being incremented by one, until the control packet reaches its desired (destination) WMN. The hop count of the nodes computed in this way is used to initialize the routing table. It should be noted however that the hop count from the sink to any node can only converge to its optimal value after an exhaustive transfer of control packets through a sequence of all available neighbours up to when it reaches the desired node. It is assumed in this paper that the sinks have good reputation i.e. it can't corrupt the network,. The first stage of the protocol design ends with the full routing table built from each of the sinks. The second stage is the secured routing, here each node that need to send data to the sinks will route through its immediate neighbour. However it is not as simple as that, before the routing is done, the integrity procedure $(f_{ei}:f_{di};f_{rbi})$ is activated. This sub-function was used to identify all trusted neighbour nodes using a POMDP model. This part of the protocol is beyond the scope off this paper, however it was inbuilt in the protocol. Next is the optimal routing from a source to the sink using the modified Q-learning model described in section 3.2 and thereafter applying the route pruning heuristics to limit the number of routes transversed in the protocol i.e. limiting the maximum number of neighbour nodes to three. This route pruning heuristics is aimed at reducing the number of actions steps (routes) stored in the protocol. This is shown on lines 11 – 26 in figure 5. The last phase is the creation of clusters shown on lines 28 – 33.

```
1: start:
2: start_cost_procedure();
3: send (DATA_REQ):
4  do: initnext_hop = 0
5: find next_hop (loc.sinkNM,loc.nextNM,loc.fit(fhop$_i$, fRe$_i$, fT$_i$,fRt$_i$);
6: incr_next_hop(loc.sinkNM,loc.nextNM,loc.hiv(hco,loc. fit(fhop$_i$, fRe$_i$, fT$_i$,fRt$_i$);
7  next_hop = next_hop + 1
8   while next_hop != NULL
9: ACK(data_packet p):
10: // exhaustive search for neighbour node for routing
11: do: send_Control_Packet(compute(h.sinksNM,h.nextNM);
12  iffhop$_i$, fRe$_i$, fT$_i$,fRt$_i$= “good”
13: incr_feedback(fhop$_i$, fRe$_i$, fT$_i$,fRt$_i$);
14: // forward control packet to subsequent neighbour
15  if (h.neighbour.contains(ID))
16: paths = find_available_paths (h.the_sink,hiv);
17: path = randnei.pick_path(paths);
18 if randnei>3 goto end
18: h.path = path;
19: h.ACK = optimal_path(h.fhop$_i$, fRe$_i$, fT$_i$,fRt$_i$);
20: propagate(c);
21   if d.sink != “NULL”
22   goto 11
23   end if
24   else goto 11
25   end if
26  while d.sink.Stamps < threshold
27  // formation of cluster head
28  if (d.nexthop > d.self
29  clusterhead = self
30  clustermember = nexthop
31  else
32  clustermember = self
33: end if
```

Figure 5 Full WCRL Pseudocode

3.5 Control Packet Structure in WCRL

As stated earlier, before the routing table is computed, the sink must send control packets to all the nodes in the network. This is achieved through transmitting control packet from the sink through successive sequence of neighbour nodes until the control packet reaches the desired node. It should be noted that a route request is first sent from the sink, while in return a route reply is sent back from the source by indicating the Q-value using equation 4.

The WCRL protocol used in this paper consists of six control packets: (i) W_REQ, (ii) W_REP, (iii) W_HP, (iv) W_Ne, (v) W_Tr and (vi) W_NACK. The W_REQ control packet is used by the sink to transmit sink announcement to all WMNs in the network, W_REP is used to send an acknowledgement control packet from the WMN to the sink indicating its Q-value using equation 4. W_HP contains the number of hops from the WMN to the sink, which is sent as feedback after sink announcement, W_Ne is the number of neighbour node to the WMN, W_Tr is the transmission energy of the WMN and W_NACK is an error control packet signifying

WMN link failure or node failure (i.e. non acknowledgement) These components of these six packets are depicted in figure 6.

IP address of source

IP address of sink

Number of neighbor

Figure 6a W_REQ

IP address of source

IP address of sink

Number of neighbor

Number of hop count

Transmission energy

Remaining battery energy

Figure 6b W_REP

IP address of sink

IP address of source

Figure 6c W_HP

IP address of source

IP address of sink

Transmission energy

Figure 6d W_Tr

IP address of source

Number of neighbor

Number of hop count

Figure 6e W_Ne

IP address of source

IP address of unreached sink

Figure 6f W_NACK

Figure 6 WCRL Packet Structure

Table 1 SFROMS Data Packet Structure

CH ID	**IP Address of Source**	**IP Address of Sink**	**Neighbour ID**	**CM ID**	**Encryption Key**
1 byte	4 byte	4 byte	4 byte	8 byte	2 byte
Acknowledge-ment Packet	**NACK ID**	**Q-value update**	**Sink Announcement Packet**	**Transmission Energy of WMN**	**Distance to Sink**
12 bytes	12 bytes	4 bytes	4 byte	4 bytes	12 bytes

where
CH ID: This refers to the cluster head Identification
Encryption key: This refers to the encryption key used for secured routing (Outside the scope of this paper)
CM ID: This is the cluster member Identification
Neighbour ID: This is the neighbour nodes identification of the WMN
Distance to sink. This stores the number of hops of the WMN to sink
NACK ID : This flag is raised when there is link or node failure. i.e. no acknowledgment
Q-value update: It is the update value of a node's distance from the sink
IP of Source: This flags the address of the source WMN
IP of Sink: This flags the address of the sink
Acknowledgement packet: This stores the size of acknowledged packet
Sink Announcement packet: This stores the sink announcement packet

Transmission Energy of WMN: This stores the energy of the WMN.

It should be noted that anytime route request is to be sent from the sink to a particular node, the sink will send control packet through any of its neighbour nodes towards the node. Initially the number of hop count is set to 0, indicating that the hop count of the source (sink) to itself is 0. Successive routing (forwarding) through a sequence of neighbour nodes from the sink increments the hop count by 1. The other four parameters involved n identifying the neighbour nodes are also updated accordingly using equation 4. This process is called updating Q-value. This was explained in the Q-learning model. The process of computing the routing table involves routing from the source through available fit neighbour nodes to the appropriate sink. This is performed using the feedback procedure whose transmission is in opposite direction to the sink announcement procedure. It should be however noted that this Q-value update process can only converge for a particular node after the sink route through all its available fit neighbour to the node.

4.1 Simulation Experiment

This section describes the simulation experiments used for the WCRL protocol proposed in this paper. The simulation was done with MATLAB simulator 2020. The simulation parameters is shown in Table 2

Table 2 Simulation Parameters

Parameters	**Values**	**Parameters**	**Values**
Width of 3D beam	360^0	Time for simulation	I hour
Direction of focus	90^0	Simulation area	1200m X 1200m
Bandwidth of link	12Mbps	Range of number of nodes	50-600
No of Frequency channel	4	Packet size of Data	1250 bytes
Frequency of Transmission	2500MHz	Speed of Mobility	0 – 180km/h
Power of Transmission	20dbm	Standard deviation of shadowing	5.0
Threshold carrier sense	60dbm	Energy loss in System	1.2
Gain of Antenna	1.2dbi	Height of Antenna	2.0m
Average height of building	12m	Ricean factor	12.0db
Width of street	40m	Maximum length of queue	60packets
Path loss	2.5	Threshold of RTs	3600 bytes
Broadcast range of Antenna	100 – 350m		

Each of the WMN used contains memory that stores the information contained in the routing tables which is used to compute Q-values necessary for the selection of routes to the sink. Source of traffic was generated using a continuous bit rate generator. The range of the bit generation was set to 0 – 30 packets/s

Simulation experiment was performed in MATLAB 2020 by comparing the performance WCRL protocol proposed in this paper to that of meme-WSN (Masood et al 2021) and PBC-CP (Pathak 2020). The performance measures used for the simulation were (i) control overhead (ii) lifetime of cluster and (iii) the rate of cluster head re-computation. The efficiency of these metrics were measured using the following (i) the number of clusters (NC) which denotes the total number of clusters formed using the entire WMNs in the network area. It should be noted here that the smaller the number of cluster formed, the greater will be the stability of the network because fewer packet transmission will be involved in routing data to the sinks especially for those WMNs farthest from the sink. Similarly the rate of frequency reuse will increase when the number of clusters are increased. This will lead to higher probability of packet collision and more energy drain. (ii) Rate of re-affiliation (RoR). This represents the time it takes the network to adjust its cluster formation in an event of node failure or node moving out of the coverage area of its cluster head. In such scenario a new cluster formation must be formed to take cognizance of the new status of the WMNs. (iii) Control Overhead: This represents the number of control packets incurred in the event of node failure or node moving out of the coverage area of its cluster head. This may also arise when the topology of the network change as a result of increase or decrease in the number of WMNs.

4.2 Results and Analysis of Simulation Experiment

Figure 7 shows the results obtained during the simulation experiment to determine the number of clusters formed with variation in the number of WMNs in the network. The number of WMNs was varied from 50 – 300 nodes. It can be noticed that generally with all the protocols, the number of clusters formed increases with an increase in the number of WMNs in the network. However the number of clusters formed per WMNs is least with the WCRL protocol. This can be attributed to the learning paradigm of the Q-learning protocol as it was able to have up to date information about the different WMNs in the network. The computation for the hop size from each node to the sink can be quickly computed due to the heuristics employed in the protocol in reducing the state-actions pairs necessary for the protocol to converge. This is opposed to the memeWSN protocol where the routes through all the neighbour nodes to every WMN have to be utilized before deciding on WMNs that can be in the same cluster. The PBC-CP protocol also suffer from the same shortcoming stated for the memeWSN protocol but its performance is worse than for the memeWSN because of the increase in computation requirements resulting from increased number of ants involved in its computation. From the graph it can be noticed that the WCRL protocol outperforms the memeWSN by 17% and the PBC-CP by 29%.

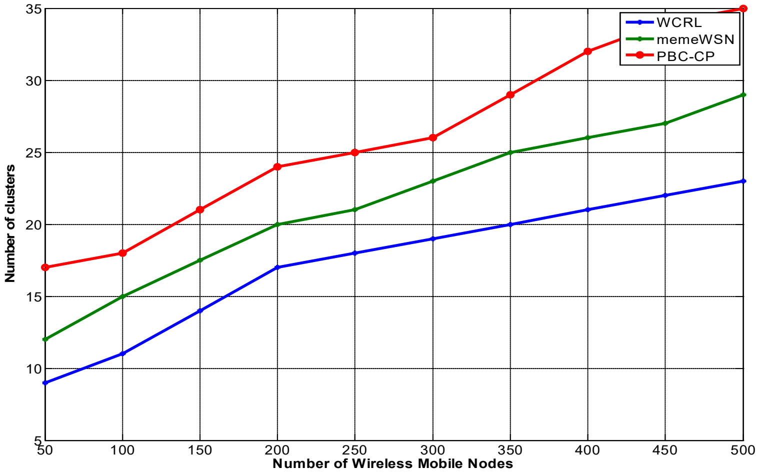

Figure 7 Number of Clusters with variation in WSN - IoT size WMN Broadcast range = 50m

In Figure 8 the simulation was modified by increasing the broadcast range of the WMNs from 50 m used in figure 7 to 200m used in Figure 8. From figure 8, it can be noticed that the number of clusters formed now is lower compared to when the coverage radius was limited to 50m This is because with an increased coverage radius, the WMNs can cover more distance and hence the reason for the reduced number of clusters. However the WCRL was able to further reduce the number of cluster formed due to the reasons adduced earlier for figure 7. In figure 8 The WCRL outperforms the memeWSN by 12% and the PBC-CP by 22%.

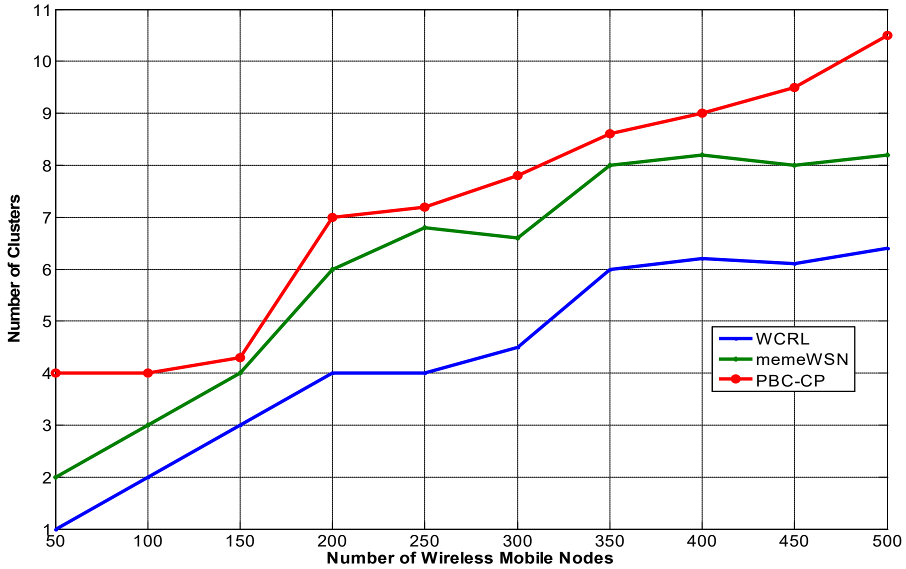

Figure 8 Number of Clusters with variation in WSN - IoT size WMN Broadcast range = 200m

4.3 Results and Analysis of Mobility of WMNs on Lifetime of Cluster

Figure 9 shows the relation between the mobility of the WMNs to the cluster lifetime. The number of nodes used in the simulation was 100 while the coverage radius of each WMN was set at 200m. The mobility range of the WMN is between 0 – 180km/h. From the figure it can be noticed that the cluster lifetime decreases as the speed of the WMNs increases. However the life expectancy rate in the WCRL protocol was higher than that of both memeWSN and PBC-CP. This is because the learning paradigm of the Q-learning protocol enables it to quickly determine the status of all WMNs, therefore the fact that a WMN moves out of the coverage area of a cluster head does not necessarily connote the activation of the redistribution procedure. It should be noted here that the WCRL protocol proposed in this paper has the innate ability of knowing the status of all WMNS in the network through its Q-value update of each WMN status. This was explained in section 3.2. This gives it an

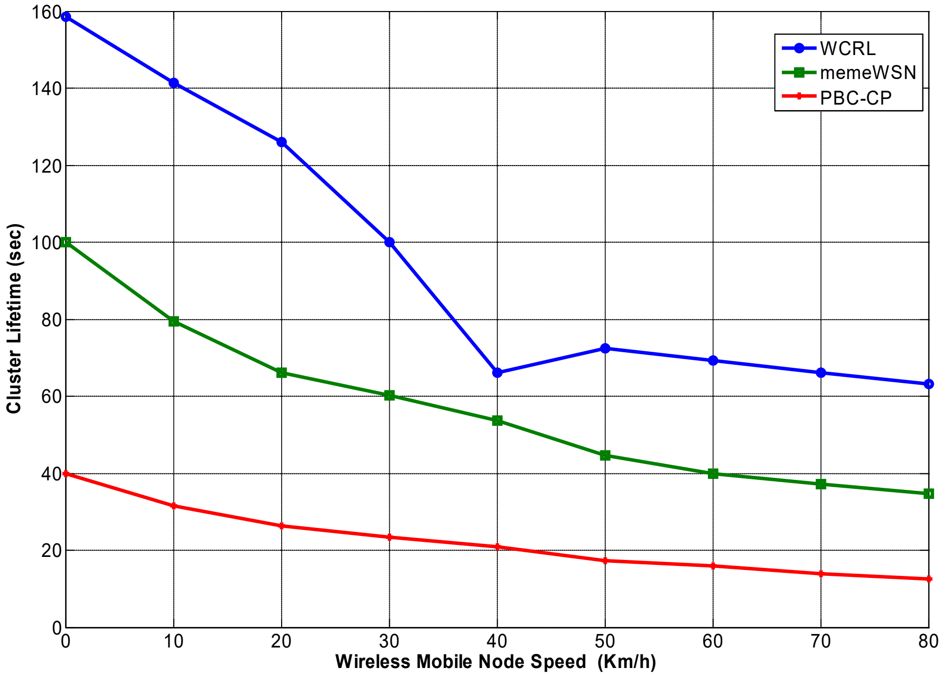

Figure 9 Cluster Lifetime to variation in the WMN speed WMN coverage radius = 50m

edge over the memeWSN protocol where clusters members were formed based of their relative speed. The shortcoming of this approach is that the mobility characteristics of WMNs are mostly random and hence cannot be predicted as assumed in the protocol. In the case of PBC-CP protocol, the mobility of the WMNs was not included in its design, so it performs poorly in the simulation. From Figure 9 it can be seen that the WCRL protocol outperform the memeWSN protocol by 16% and the PBC-CP protocol by 34%. The memeWSN protocol's performance declines when the speed of the WMN is less than 40km/h, however it performs favourably with the WCRL protocol when the speed of the WMNs is higher than 40km/h This can be adduced to the fact that as the speed of all WMNs increase, the protocol is able to adapt than with lower speed. The performance of PBC-CP protocol is worse compared to the others due to the fact that mobility of the nodes was not considered in its design.

In figure 10, the coverage radius of the WMNs was increased to 300 meters. From the graph, it can be seen that the lifetime of the clusters was increased across board for all the protocols; this is due to the fact that the increased coverage makes more WMNs to be contained in a single cluster. This leads to lesser number of clusters in the network which invariably leads to lesser need for the activation of redistribution procedure thereby leading to longer life expectancy for the clusters. From the figure, the WCRL protocol outperforms the memeWSN protocol by 11% and the PBC-CP protocol by 24%

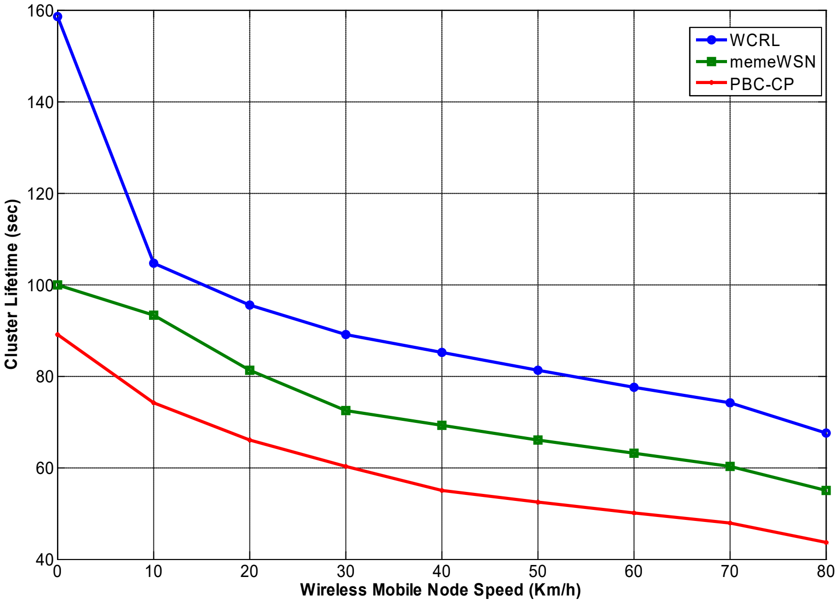

Figure 10 Cluster Lifetime to variation in the WMN speed WMN coverage radius = 300m

Rate of Re-affiliation: (RoR) In this simulation, two experiments were conducted to examine the stability of cluster head to variation in the speed of WMNs. As explained earlier, re-affiliation is caused by change in Q-value, mobility or link failure of one or more nodes within a cluster. This leads to cluster members leaving its cluster head to join other clusters. In this simulation the coverage radius of the WMNs were fixed to 200m, while the number of WMNs used was 100, this was randomly distributed in a network of size 1000 m by 1000m. The speed of the WMNs were varied between 0 – 180km/h, the average of 50 simulations was used for the experiment. From figure 11, it can be seen that WCRL protocol has the lowest re-affiliation rate due to its stable formation of clusters. This is because the WCRL protocol is always equipped with the current status of all WMNs through the regular computation of the Q-values. Hence a change involving mobility, link or node failure of WMN is taken care of intelligently through the appropriate update of the Q-value and necessary adjustments to the cluster in question without necessarily affecting other clusters. Each WMN has embedded in it an intelligent agent which enables it join the appropriate cluster head. It is only when the speed of the affected WMN is greater than that of the cluster head above a threshold value would the redistribution procedure be activated. This gives the protocol an edge over the memeWSN protocol where any change in mobility, link or node failure leads to the activation of the redistribution procedure. From figure 11 WCRL protocol outperforms the memeWSN protocol by 15% and the PBC-CP protocol by 26% .

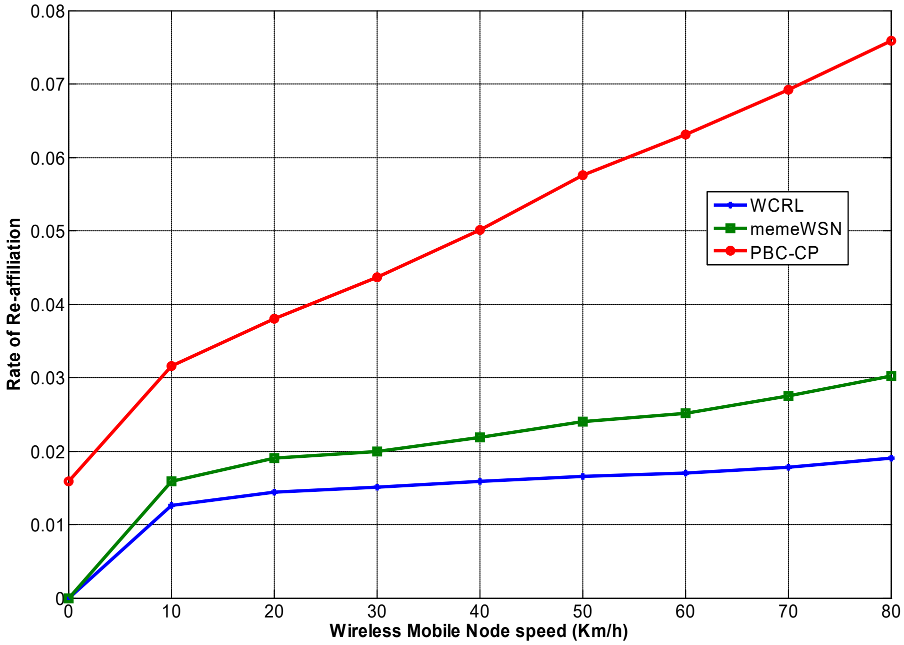

Figure 11 Re-affiliation rate to variation in WMN speed, WMN coverage radius = 50m

The second part of the simulation see the coverage radius of the WMNs increased to 300m. From figure 12, it can be seen that the increase in the coverage radius leads to a more stable cluster formation resulting from lesser number of clusters being formed in the network. This invariably leads to a longer lifetime for the WSN IoT devices, due to reduced need for re-affiliation; however WCRL protocol has the highest lifetime expectancy. The WCRL protocol outperforms the memeWSN protocol by 7% and the PBC-CP protocol by 26%. As in previous simulation, the performance of PBC-CP protocol is worst because the mobility of WMNs was not included in its design.

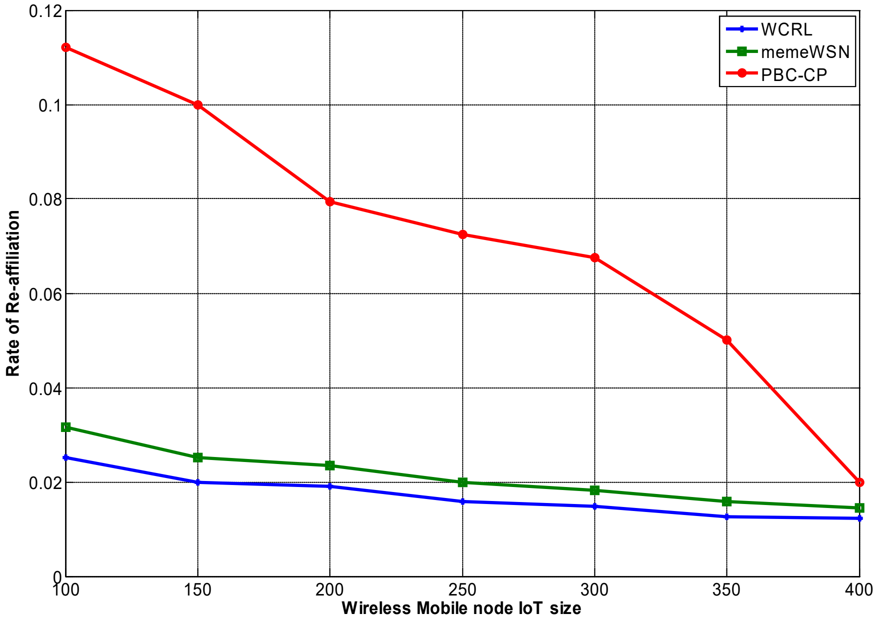

Figure 12 Re-affiliation rate to variation in WMN speed, WMN coverage radius = 300m

Figure 12 shows the experiment on re-affiliation rate performed with variation in the number of WMNs. The network size was incremented iteratively in steps of 50 nodes from 100 WMNs to 400 WMNs. From the figure, it can be seen that the progressive increase in coverage area leads to a corresponding decrease in the rate of re-affiliation. This is due to reduced number of clusters been formed and hence less need for the activation of the redistribution procedure. In figure 13, WCRL outperforms memeWSN protocol by 13% and RBC-CP protocol by 24%. The design of the WCRL protocol does not lead to frequent change in the WMNs in each cluster this is because the change here is due to each WMN adapting intelligently to different cluster heads as a result of the computation of Q-values to different cluster heads. However the memeWSN protocol forms clusters using WMNs with same relative velocity to each other. This may sometimes lead to frequent re-affiliation because the absolute speed of the WMNs cannot be predicted.

4.4 Results and Analysis of Control Message Overhead on Lifetime of Cluster

Control Message Overhead (CMO) : Control messages are packets that are not part of the data packets transmitted during network communication. However these packets are used to set the tone for the network. Control packets include but are not limited to start and end of data frame, data overflow, status request and acknowledgement of different WMNs in the network. This section will demonstrate the analysis and results of simulation done in order to determine the number of control messages (packets) involved in the election of cluster heads and cluster members. As was the case with previous simulation, the performance of WCRL protocol proposed in this paper was compared to PBC-PC and memeWSN protocols

The simulation network consists of 50 WMNs randomly deployed in a simulation size of 1000m by 1000m. The mobility model employed was the Random way point. The coverage radius of the WMNs were set to 200m and the speed (mobility) range of the WMNs were between 0 – 80 km/h, here 0 – 5km/h represents walking speed, 5 – 20km/h represents running speed and 20 – 80km/h represents car speed. The result of the simulation is presented in figure 13. Another simulation for the Control Message Overhead was performed by varying the coverage radius of the WMNs from 200m to 300m, the results of this simulation is shown in figure 14.

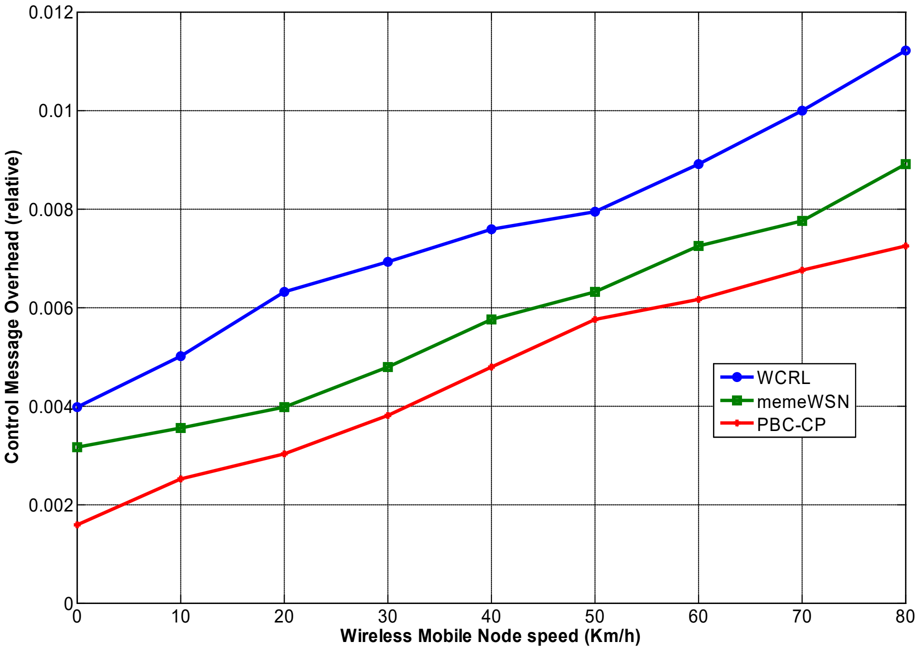

Figure 13 Control Message Overhead to variation in WMN speed, coverage radius of WMN = 200m

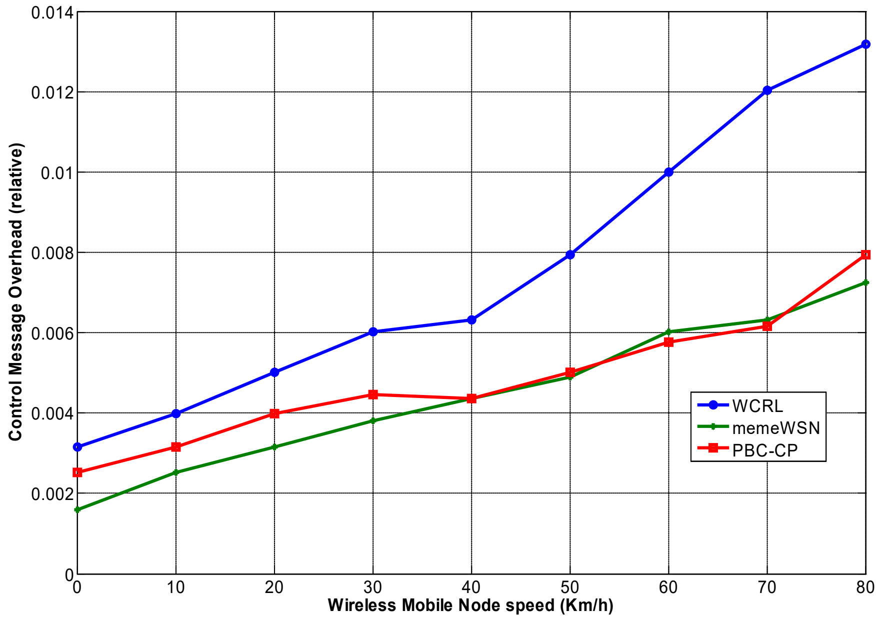

Figure 14 Control Message Overhead to variation in WMN speed, coverage radius of WMN = 300m

As can be seen from figures 13 and figure 14, the Control message overhead was least for the WCRL protocol, this is due to the fact that the state action space involved in the routing for the election of cluster heads was cleverly pruned as discussed in section 3.2. This resulted in lesser nodes involved in routing and a quick convergence rate. The second reason can be adduced to the lower re-affiliation rate of the WCRL protocol caused by the learning paradigm which does not favour the redistribution of networks in the event of node mobility or failure, but rather makes a WMN attach itself to a cluster head or become a cluster head using the intelligent agent embedded in it. The PBC-CP protocol performs worst mainly because mobility of WMNs was not included in its design. It can also be observed that an increase in the coverage radius of the WMN led to a reduction in the control messages overhead across all protocols. This is because an increase in coverage radius of the WMN reduces the chances of cluster head redistribution procedure. From figures 13 and 14, it can be seen that the WCRL protocol outperforms the memeWSN protocol in terms of Control messages overhead by 15% and the PBC-CP protocol by 24%.

6. Conclusion and Future Work

This paper has been able to demonstrate the effectiveness of the application of reinforcement learning and in particular Q- learning in the clustering algorithm for WSN - IoT. The advantage of the protocol is mainly due to its low memory consideration especially as it involve novel techniques that that be used to prune the number of state action processes required in electing of cluster heads. Also the protocol does not require an explicit model of the network environment, this is a shortcoming of the compared protocols. Q-learning is based on the continuous interaction of an intelligent agent with its environment. This makes it get the actual condition of the nodes and not a probabilistic or random model of the environment as used in the genetic algorithm and memeWSN protocol respectively. The simulation experiments conducted in this paper has been able to justify the advantage of the Q-learning technique over other state of the art protocols and more importantly justify the application of an energy efficient clustering algorithm for WSN-IoT.

The future drive of this paper will include higher mobility of WMNs especially nodes in fast moving cars and trains.

.

References

Ali H., Shahzad W., and Khan F. A., (2018) Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. *In Journal of Applied Soft Computing* (Vol. 12(7 pp. 1913–1928).

Azni A. H., Ahmad R., Seman K, .Alwi N. H., and Noh Z. A, (2021) Correlated topology control algorithm for survival network in MANETS. *In Advanced Computer and Communication Engineering Technology* (Vol 4(2) pp 214 -228)

Basagni S. and Chlamtac I.,(2007) "A generalized clustering algorithm for peer-to-peer networks,*" In Workshop on Algorithmic Aspects of Communication,* Bologna, Italy.

Behera T. M., Mohapatra S. K., Samal U. C., Khan M. S., Daneshmand M., and A. H. Gandomi A. H., (2019) Residual energy based cluster-head selection in WSNs for IoT application. *IEEE Internet of Things Journal*. (Vol. 6, no. 3, pp. 5132–5139).

Belding-Royer E. M., (2012) Hierarchical routing in ad-hoc mobile networks *Journal of Wireless Communications and Mobile Computing,* vol. 2, no. 5, pp. 532- 546.

Cheng H., Yang S., and Cao J., (2020) Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks. *In Journal of Expert Systems with Applications* (Vol. 40 pp. 1381–1392).

Deb K., Sindhya K., and Hakanen J, (2020) Multi-objective optimization. *In Decision Sciences: Theory and Practice*, pp. 145–184, CRC Press.Gupta P and Kumar P. R. (2010), The capacity of wireless networks, *IEEE Transactions on Information Theory*, Vol 46(2) pp. 388–404,

Fagbohunmi, G.S and Eneh I. I. (2015) Improving the scalabilty of wireless sensor networks by reducing sink node isolation, *International Journal of Applied Information systems* (Vol 8 No 7 pp 25–32)

Fagbohunmi, G. S. and Eneh I. I (2019) A secured routing protocol for wireless sensor networks using Q-learning, *In International Journal of Information, Technology & Innovation in Africa* (Vol 11 Number 4) [3]

Kannan G. and Sree Renga Raja T., (2019) Energy efficient distributed cluster head scheduling scheme for two tiered wireless sensor network. *Egyptian Informatics Journal*, (Vol. 16, no. 2, pp. 167–174).

Konstantopoulos, Gavalas C. D and Pantziou G., (2017) Clustering in mobile ad hoc networks through neighborhood stability based mobility prediction. *In Journal of Computer Networks*. (Vol. 52, pp. 1797–1824).

Masood A., Gohar A., Babar S., Abdul H., Abrar U., Fernando M. and Omar A. (2021) "Optimal Clustering in Wireless Sensor Networks for the Internet of Things Based on Memetic Algorithm: memeWSN" Hindawi Wireless Communications and Mobile Computing, Volume 1 no. 3 Article ID 8875950.

Pathak A (2020) A proficient bee colony-clustering protocol to prolong lifetime of wireless sensor networks *Journal of Computer Networks and Communications*, (Vol. 20(4) pp 85-92). [17]

Perkins C. E, (2011) *Ad-hoc Networking*, Addison-Wesley Publisher .

Shah N., S. Abid A., Qian D., and Mehmood W., (2020) A survey of P2P content sharing in MANETs. *Journal of Computers & Electrical Engineering*, vol. 57, pp. 55–68.

Venkanna U. and Leela Velusamy R., (2020) Distributed cluster head election in MANET by using AHP," In *Hournal of Peer-to-Peer Networking and Applications*, (Vol. 9, no. 1, pp. 159–170).