A Lifetime-Aware Centralized Routing Protocol for Wireless Sensor Networks using Reinforcement Learning

Elvis Obi
Computer Science Research Institute
Paul Sabatier University
Toulouse, France
elvis.obi@irit.fr

Zoubir Mammeri
Computer Science Research Institute
Paul Sabatier University
Toulouse, France
zoubir.mammeri@irit.fr

Okechukwu E. Ochia
Department of Electrical Engineering
University of Calgary
Calgary, Canada
okechukwu.ochia1@ucalgary.ca

***Abstract*—This paper presents the design of a Lifetime-Aware Centralized Q-routing Protocol (LACQRP) for Wireless Sensor Network (WSN) to maximize the network lifetime. This is achieved by implementing Q-learning on the sink of the WSN, which also acts as a controller that has global knowledge of the network topology as enabled by Software-Defined WSN (SDWSN). The controller generates all possible distance-based minimum spanning trees (MSTs), which form the set of routing tables (RTs). The maximization of the network lifetime is achieved by the controller learning the routing table that minimizes the maximum of the sensor nodes' consumption energies using Reinforcement Learning (RL). The simulation results show that the LACQRP learns the best RT that maximizes the network lifetime and has a better network lifetime performance when compared with recent distributed RL routing protocols for lifetime optimization, which are Reinforcement Learning-Based Routing (RLBR) and Reinforcement Learning for Lifetime Optimization (R2LTO).**

***Index Terms*—reinforcement learning, routing, wireless sensor network, software-defined wireless sensor network, network lifetime, path optimization**

I. INTRODUCTION

A wireless sensor network (WSN) is a group of densely deployed, spatial wireless nodes that carry out application-specific tasks such as tracking, data logging, monitoring, etc. in a given environment with the aim of aggregating and sending the recorded data to a central location [1]. The wireless nodes are limited in memory, battery capacity, and computational power. This makes the problem of optimizing the lifetime of WSN by reducing the energy consumption of the wireless nodes a challenging task [2]. To tackle this problem, the software-defined WSN (SDWSN) paradigm has been proposed in recent works [3]. SDWSN involves the integration of WSN and software-defined networking (SDN) into a unified framework. SDN is a technology that separates the data plane from the control plane in network devices, allowing network control functions to be centralized [4]. SDN enables adaptive configuration of network devices and simplifies the complexity of management of these devices by the use of a central controller that has global knowledge about the entire network [5]. Therefore, in the SDWSN paradigm, a central controller performs the control and computational functionalities of the WSN, allowing the wireless nodes to perform the task of data forwarding [6]. This makes the SDWSN concept to be suitable in reducing the energy consumption of sensor nodes during routing of data packets to the sink and hence improves the network lifetime [7]. The SDN controller because of its global information of the network can build several minimum spanning trees (MSTs) to be used as possible routing tables (RTs) by the sensor nodes to send data packets to the sink. Consequently, due to the dynamics of the SDWSN such as topology and traffic changes, Reinforcement Learning (RL) can be used to learn the best MST to optimize the network lifetime. RL is a field of machine learning that enables an agent to learn the dynamic behavior of its environment by taking an action based on its current state which improves learning with time (maximizing the concept of cumulative reward) using trial and error interaction on the environment [8]. For example, a central controller interacts with all the nodes in the network to make routing decisions for the nodes. In this case, the agent is the controller and the environment is the controller's neighborhood, the state is the current tree the nodes are using to send data packets, and the action is the selection of the next tree to be used by all the nodes to send packets. Existing protocols provided on unicast routing for WSN with RL for network lifetime optimization to the best of the authors' knowledge are implemented in a distributed manner. These distributed RL- based unicast routing protocols are constrained by the learning agent having localized information of the entire network. This causes delay in learning the optimal routing path as a result of the exchange of the network information between the network nodes. This will affect the network lifetime and simulation run time. These drawbacks are alleviated by the proposed centralized unicast RL-based routing protocol called Lifetime-Aware Centralized Q-routing Protocol (LACQRP).

This paper is organized as follows: A review of similar works is provided in section II and the methodology of the proposed LACQRP is presented in section III. The simulation and results are discussed in section IV and section V concludes the paper.

978-1-6654-2854-5/21/$31.00 ©2021 IEEE

II. REVIEW OF SIMILAR WORKS

The first hop-by-hop routing protocol to utilize RL is called Q-routing proposed by Boyan and Littman, (1994) [9]. Q-routing minimizes the packet delivery delay. However, Q-routing suffers from Q-value freshness, slow convergence, and it is very sensitive to parameter setting. Different works have considered the design of routing protocols to optimize the lifetime of WSN using RL. The sequel gives the presentation of these works. Wang and Wang, (2006) proposed a routing algorithm called Adaptive Routing for WSNs using RL (AdaR) to maximize the network lifetime [10]. The protocol uses the multiple factors of hop count, residual energy, link reliability, and the number of routing paths crossing a node to determine the optimal routing path. AdaR converges faster than Q-routing to the optimal solution and does not suffer from the problem of initial parameter setting. Dong et al., (2007) proposed for ultra-wideband sensor networks a Reinforcement Learning Based Geographical Routing Protocol (RLGR) [11]. The protocol seeks to improve the network lifetime by reducing packet delivery delay and distributing energy consumption among nodes uniformly. RLGR considers hop counts to the sink, residual energy of nodes in choosing the next forwarder. RLGR improved the network lifetime by at least 75 percent when compared with Greedy Perimeter Stateless Routing (GPSR) [12] by simulation. Yang et al., (2013) proposed a reinforcement learning-based routing protocol between sensor nodes and mobile sinks, which are vehicles [13]. The protocol enable the direct interaction between the sensor nodes and the mobile sinks taking multiple metrics such as residual energy, hop count in learning the routing paths. Renold Chandrakala, (2017) proposed for WSNs a routing protocol called Multi-agent Reinforcement Learning-based Self-Configuration and Self-Optimization (MRL-SCSO) [14]. In this protocol, the reward function is defined using the buffer length and the node residual energy. The next forwarder selected is the neighbor with the maximum reward value. The protocol also incorporates the sleeping scheduling scheme to decrease the energy consumption of nodes. The network lifetime of MRL-SCSO is higher than that of Collect Tree Protocol (CPT) [15] when compared by simulation. Geo et al., (2019) proposed for WSN a Q-learning routing protocol called a Reinforcement Learning-Based Routing (RLBR) to optimize the network lifetime [16]. RLBR search for optimal paths for transmitting packets from each node to the sink taking into consideration of hop count, link distance, residual energy in its reward function. RLBR utilizes transmit power adjusting and data packet carrying feedback scheme to increase packet delivery, balances the energy consumption, and reduces the overall energy consumption. RLBR performs better than Q-Routing, MRL-SCSO in terms of network lifetime and energy efficiency. Bouzid et al., (2020) proposed a routing protocol for WSN known as Reinforcement Learning for Lifetime Optimization (R2LTO) to optimize lifetime and energy consumption [17]. R2LTO learns the optimal paths to the sink by considering the hop count, residual energy, and transmission energy (distance) between nodes. R2LTO consists of two processes, which are the discovery process to know the network topology and the continuous learning routing process. The effectiveness of R2LTO is carried out by comparison with Q-routing and RLBR by simulation, and the results show that R2LTO performs better in terms of network lifetime and energy efficiency. The RL-based routing protocols to optimize network lifetime reviewed so far are distributed in nature. These distributed protocols are constrained by having local information available at each sensor node regarding the present network connectivity. This results in a slow calculation and learning of the optimal routing path because of the time required to exchange routing information among neighboring nodes. This results in degradation in the network lifetime. Because of the drawbacks associated with the distributed RL-based unicast routing in WSNs, this paper aims to maximize the network lifetime in WSN with centralized routing protocol using RL.

III. METHODOLOGY

The topology of the WSN is modeled as a weighted graph, $G = (V, E)$. V is the set of network nodes (vertices) and E is the set of network links (edges). The connection between two nodes in the network is represented by a distance edge weight. In the proposed routing protocol, each WSN node broadcasts *Hello* packets after the initialization of the network. Based on the received *Hello* packets, the sink/controller builds the network graph and computes a list of all routing tables (RTs) based on distance-based minimum spanning trees (MSTs) [18]. The choice of the distance-based MST is because the transmission energy that the sensor nodes use to send packets is a function of the distance between nodes.

A. All MSTs Algorithm

The algorithm and the complexity of generating all MSTs are explained in the sequel.

The node set and edge set of the network graph are $V = \{v_1, ..., v_n\}$ and $E = \{e_1, ..., e_m\} \subseteq V \times V$, respectively. An integer weight $w(e) > 0$ is associated with each edge $e \in E$. The sum of the weights of constituent edges for an MST is depicted as weight $w(T)$. The list of all MSTs of the network graph is obtained by using a set of fixed edges $F = \{e_1, ..., e_k\}$ and a set of restricted edges $R \subseteq E$ in G that is disjoint with F, where k is the number of elements in F. An MST is said to be (F, R)-admissible if it contains all edges of F, but does not contain those of R. An MST of G, obtained by any standard MST algorithm [19] is used to divide the problem P of finding an MST into a set of mutually disjoint sub-problems $P(\{e_1, ..., e_{i-1}\}, \{e_i\})$, where $i = 1, ..., n-1$. This implies that the sub-problems $i = 1, ..., n-1$ list all the MSTs that contain $e_1, ..., e_{i-1}$, but do not contain e_i. Therefore the problem P becomes $P(F, R)$: List all the MSTs, which are (F, R)-admissible. an MST that is (F, R)-admissible is denoted by $T(F, R) = F \cup \{e_{k+1}, ..., e_{n-1}\}$. For $i = k+1, ..., n-1$, F_i and R_i are defined as $F_i = F \cup \{e_{k+1}, ..., e_{n-1}\}$ and $R_i = R \cup \{e_i\}$, respectively. Moreover, let e be an arbitrary edge of an (F, R)-admissible MST, T of G. Deleting e from T divides it into two non-connected components V_1 and V_2. $Cut(e)$ is the set of edges that can substitute e and reconnect V_1 and V_2, and is defined as $Cut(\hat{e}) = \{e^* \in E | e^* \in (V_1 \times V_2) \cup (V_2 \times V_1)\}$. From the cut-set optimality condition for MST, for a pair of edges $e \in T$ and $e^* \in Cut(e)\backslash\{e\}$, $T \cup \{e^*\}\backslash\{e\}$ defines an MST. Renumbering the vertices of the (F, R)-admissible MST at each sub-problem $P(F, R)$ in a post-order fashion as T is transverse from an arbitrary root as $\{v_i | i = 1, ..., n\}$, this makes T to be rooted at v_n, thereby making e_i to be an edge connecting v_i to be its parent vertex in T. An interval $[\sigma_i, \psi_i]$ is associated with v_i and represents the set of descendants of it. This implies that $j \in [\sigma_i, \psi_i] \iff v_j$ is a descendant of v_i in T rooted at v_n. Denoting $E_i = \{(v_i, v_j) \in E | (v_i, v_j) \notin T\}$ as a set that are not tree edges incident on vertex v_i and the set of quasi-cuts, Q to be a set of elements of the form $(w, v, v^*) \in Q$. This implies $e = (v, v^*) \in E$ has weight $w(e) = w$, and is a candidate of a cut-set edge. Q is then use to find the substitute e_i^* of $e_i \in T\backslash F$ which enables getting $T \cup \{e_i^*\}\backslash\{e_i\}$ as a new MST and updating it for the next i^{th} sub-problem. The All MSTs algorithm runs in $O(Nm \log n)$ time and $O(m)$ space. Where n, m, and N are the number of nodes, edges, and MSTs of the network graph, respectively. This is because Q includes a maximum of m elements, and for every non-tree edges, a maximum of two $Inserts$ and two $Deletes$ are performed. Also, for every tree edge, a maximum of one $Find$ is performed. Because Q is an order set, every $Inserts$, $Deletes$, and $Find$ is executed in $O(\log m)$ time. Therefore the total substitutes is performed in $O(m \log m) = O(m \log n)$ time. The other computation like traversing G along T, finding intervals $[\sigma_i, \psi_i]$, and renumbering V in post-order manner is done in $O(m)$ time. The algorithm for generating all the possible MSTs of a network graph using this description is given in **Algorithm 1**.

B. Lifetime-Aware Centralized Q-Routing Protocol

In this work, the lifetime of the network is considered as the time required for the first sensor node to die. Therefore, a centralized RL-based unicast routing protocol is designed for a WSN to maximize the minimum Estimated Node Residual Energy ($ENRE$) of the sensor nodes in the network. Therefore, the optimization problem is to find the RT of the WSN such that:

$$Minimum\, ENRE_n \; (for\; all\; n)\; is\; Maximized \qquad (1)$$

This is because, despite using the MSTs as the RTs to minimize the energy consumption of sensor nodes, the number of paths crossing each sensor node in the different RTs differs. This parameter makes the energy consumption of each sensor node to be different when using the different RTs for routing. The RT that has the minimum number of paths crossing a particular sensor node will drain less energy from the sensor node. Therefore, to prolong the time taken for the first sensor node to die, the proposed routing protocol tends to find the RT that has the least number of paths crossing a particular sensor node to be used for routing packets to the sink. The learning agent is located at the controller that also acts as the sink. The sink collects all the data sent by the sensor nodes in the network. The controller builds all the possible RTs. Therefore, the state space S and action space A of the learning agent are the lists of all RTs. The state of the learning agent is the RT that the sink is using to receive packets from the sensor node at the current learning round and the action is to choose the next RT that will optimize the network lifetime. After, each round of data transmission by the sensor nodes to the sink, each sensor node sends its residual energy to the sink. Based on the residual energy of each sensor node, the sink estimates the energy consumption of each sensor node in the previous

Algorithm 1 All MSTs Algorithm

```
Input: $F$, $R$, $T$
Output: All MSTs
 1: $Q = \{\}$
 2: for $i = 1$ to $n - 1$ do
 3:     for $e = (v^i, v^j) \in E^i \backslash R$ do
 4:         if $j < \sigma_i$ then
 5:             Reverse the direction
 6:             if $(w(e), j, i) \in Q$ then
 7:                 Delete it from $Q$
 8:                 Insert $(w(e), i, j)$ into $Q$
 9:             end if
10:         end if
11:         if $j \in [\sigma_i, \psi_i]$ then
12:             if $(w(e), j, i) \in Q$ then
13:                 Delete it from $Q$
14:             end if
15:         end if
16:         if $j > \sigma_i$ then
17:             Insert $(w(e), i, j)$ into $Q$
18:         end if
19:     end for
20:     if $e_i \notin F$ then
21:         Find $(w, i^*, j^*) \ni w = w(e_i)$ and $j^* \in [\sigma_i, \psi_i]$
22:         if such an $(w, i^*, j^*)$ is found with $j^* \in [\sigma_i, \psi_i]$ then
23:             Delete $(w, i^*, j^*)$ from $Q$, and go to line 21
24:         end if
25:         if such an $(w, i^*, j^*)$ is found with $j^* \notin [\sigma_i, \psi_i]$ then
26:             Set $e_i^* = (v^{i^*}, v^{j^*})$. {Subsitute for $e_i$ found.}
27:         end if
28:     end if
29: end for
30: for $i = k + 1$ to $n - 1$ do
31:     if $e_i^*$ exists, then
32:         Set $T_i = T \cup \{e_i^*\}\backslash\{e_i\}$
33:         Output $T_i$ {Comment: A new MST is found}
34:         Set $F_i = F \cup \{e_{k+1}, ..., e_{i-1}\}$ and $R_i = R \cup \{e_i\}$
35:         Call $All\ MST(F_i, R_i, T_i)$ recursively
36:     end if
37: end for
```

round of data transmission. To make the learning meaningful, the Q-value in equation (2) is made to denote the value of the maximum energy consumption of the sensor nodes in the network when using a particular RT in sending data packets to the sink.

$$Q_t(s_t, a_t) = (1-\alpha)Q_{t-1}(s_t, a_t) + \alpha\left[R_t + \gamma * \max_{a \in A}\{Q(s_{t+1}, a)\}\right] \tag{2}$$

where α is the learning rate and γ is the discount factor. The achievable reward R_t in each learning round is modeled as the maximum of the energy consumption by the sensor nodes in the network when a particular RT is used and is given as:

$$R_t = \max_{n \in V}\{EC_n\} \tag{3}$$

where EC_n is the energy consumption of the n^{th} sensor node and V is the set of sensor nodes in the WSN. This implies that the sink evaluates the effectiveness of the RT based on the reward function after a round of data transmission. To maximize the minimum estimated node residual energy of the sensor nodes and thereby maximizing the lifetime of the WSN, exploration of the solution search space is ensured by choosing the RT that has a minimum Q-value using the epsilon-greedy strategy [8]. That is given a probability value of epsilon, $\epsilon \in [0, 1]$ and a random number, $r \in (0, 1)$ generated in each learning round, the action a_t is selected as:

$$a_t = \begin{cases} Random\ action,\ if\ r \geqslant 1 - \epsilon \\ \underset{a \in A}{\operatorname{argmin}}\{Q_t(s, a)\},\ otherwise. \end{cases} \tag{4}$$

The proposed LACQRP for finding the optimal RT for maximizing the network lifetime of the WSN is given in **Algorithm 2**.

Algorithm 2 LACQRP

```
Input: Learning rate, Discount factor, Epsilon, Number of
    learning rounds, List of RTs
Output: Optimal RT
Initialize the Q-value for each state-action pair as zero.
Initialize a random RT as the current state of the controller.
for $i = 1$ to Number of learning rounds do
    The controller chooses an RT using equation (4) and
    broadcast it to each sensor node.
    Each sensor node sends its data to the sink using the
    RT.
    The controller evaluates the effectiveness of the RT
    using equation (3).
    The controller updates it Q-value using equation (2).
    The controller updates its state as the current RT.
    if Any sensor node depletes it energy source, then
        $break$
    end if
end for
return The optimal RT as the RT with the highest
    percentage utilization.
```

Since **Algorithm 2** depends on **Algorithm 1** to generate all MSTs, which are used as the RTs, the asymptotic time complexity of **Algorithm 2** is the same as that of **Algorithm 1**. Also, **Algorithm 2** requires the initialization of a Q-matrix which depend on the size of the list of RTs. Therefore the asymptotic space complexity of **Algorithm 2** is $O(N)$. Where the upper bound of N is $n^{(n-2)}$. The convergence of the LACQRP to the optimal RT will be demonstrated by simulation. The optimal RT is the RT that has the highest percentage utilization. The percentage utilization of an RT is the ratio between the time the RT is used and the network lifetime. That is, the percentage utilization of an RT is given as:

$$U_{RT} = \frac{T_{RT}}{LT} \tag{5}$$

Where U_{RT} is the percentage utilization of an RT, T_{RT} is the time the RT is used, and LT is the network lifetime.

IV. SIMULATION AND RESULTS DISCUSSIONS

The performance of the LACQRP is achieved by simulations using the performance metric of network lifetime. The network lifetime of the LACQRP is compared with recent distributed RL protocols for network lifetime maximization for WSN which are RLBR [16] and R2LTO [17]. The network lifetime is computed as the time taken for the first sensor node to deplete its energy source. The energy consumption of the n^{th} sensor node in each round is the difference between its previous estimated node residual energy, $ENRE_n^{Previous}$ and its current estimated node residual energy, $ENRE_n^{Current}$ after the end of a round. Therefore, the energy consumption of the n^{th} sensor node after a learning round is given as:

$$EC_n = ENRE_n^{Previous} - ENRE_n^{Current} \tag{6}$$

Where EC_n is the energy consumption of the n^{th} sensor node after a round. The energy consumed in sending and receiving data is given by equation (7) and equation (8), respectively [20].

$$E_{tx}(b, d) = \begin{cases} E_{elec}b\ell + e_{fs}bd^2\ \ if\ d \leqslant d_o \\ E_{elec}b\ell + e_{mp}bd^4\ \ if\ d > d_o \end{cases} \tag{7}$$

$$E_{rx}(b) = E_{elec}b\tau \tag{8}$$

where b is the number of bits per packet, d is the distance between the sender and the receiver, ℓ is the number of packets sent by a sensor node per round, τ is the number of packet received by a sensor node per round, $E_{tx}(b, d)$ is the transmit energy, $E_{rx}(b)$ is the received energy, E_{elec} is the electronic energy to transmit or receive unit data of the packet. e_{fs}, e_{mp} are the transmit amplifier efficiency and depend on the transmitter amplifier model (free space model is used when $d \leqslant d_o$, otherwise the multipath model is used). d_o is the reference distance and is obtained by equating the two expressions at $d = d_o$ and is given as:

$$d_o = \sqrt{\frac{e_{fs}}{e_{mp}}} \tag{9}$$

In equation (7), energy consumption by a sensor node in sending packet(s) is a function of a continuous distance, but in practice the transmission power of a sensor node at less than the transmission range is used. This is implemented by normalizing the continuous distance with the transmission range and applying the ceil function on the normalized distance. The LACQRP, RLBR and R2LTO are implemented with python 3.8 under the "PyCharm" development environment. The python networkx module is used to implement the graphical structure of the WSN [21]. The code is executed on a 10^{th} generation Intel core i7-10510U laptop with a 4.9 GHz processor and 16 GB of RAM. The simulation parameters used to implement the network and the randomly generated connected WSN are as shown in Fig. 1. and Table 1, respectively.

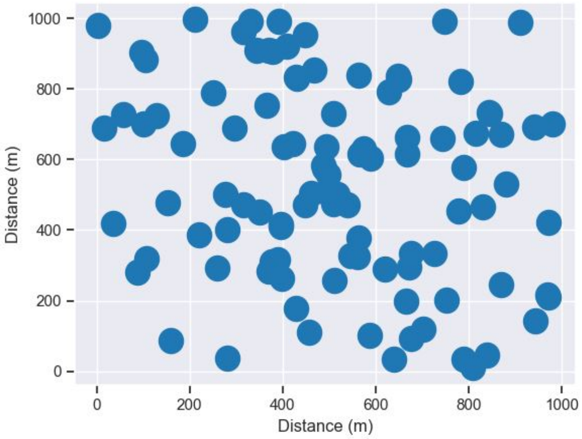

Fig. 1. The Deployed WSN.

TABLE I
SIMULATION PARAMETERS

	Values
Number of sensor nodes	100
Number of sink node	1
Deployment area	$1000\ m \times 1000\ m$
Sensor node deployment	Random
Sink position	$(500, 500)$
Transmission range	$50\ m$
Data packet size	512 bits
Packet generation rate	$1\ /s$ to $10\ /s$
Initial energy of sensor nodes	$100\ J$ to $1000\ J$
E_{elec}	$50\ nJ/bit$
e_{fs}	$10\ pJ/bit/m^2$
e_{mp}	$0.0013\ pJ/bit/m^4$
Learning rate	0.7
Discount factor	0.0
Epsilon	0.1

Fig. 1 shows the graphical representation of the randomly deployed WSN. The deployed WSN has a maximum number of 216 RTs when the transmission range of the sensor nodes is set as 50 m. The LACQRP converges to the optimal RT with the highest percentage utilization that maximizes the network lifetime. The network lifetime of LACQRP is compared with RLBR and R2LTO for increasing initial energy and packet generation rate of the sensor node as shown in Fig. 2 and Fig. 3, respectively.

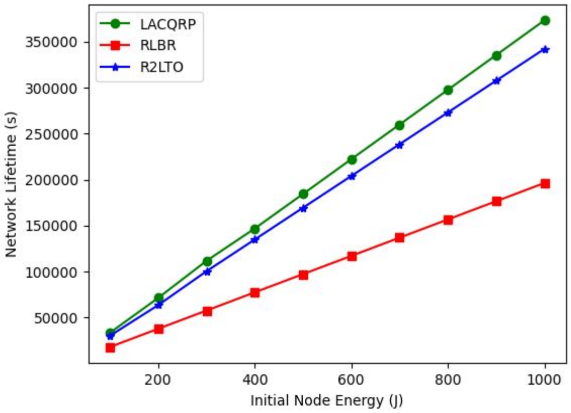

Fig. 2. Network Lifetime with increasing Initial Node Energy

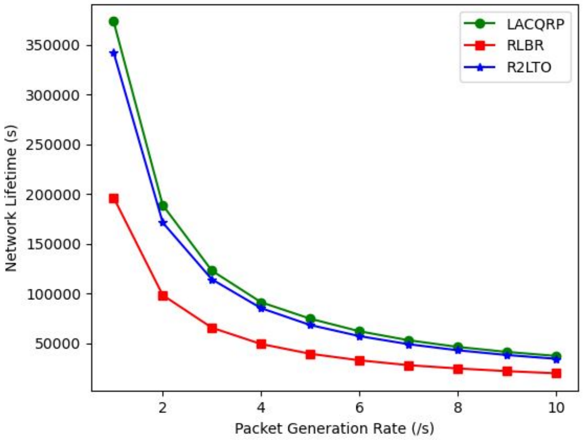

Fig. 3. Network Lifetime with increasing Packet Generation Rate

Fig. 2 shows the network lifetime of both protocols when the packet generation rate is set as one packet per second and the initial sensor node's energy is varied with the same amount. The network lifetime increases linearly with the initial sensor node's residual energy. This is because the lifetime of a sensor node is proportional to its residual energy. The linear relationship is also attributed to the same packet generation rate at the sensor nodes during the rounds of data transmission. The LACQRP has a better network lifetime performance of 90.14% and 9.26% when compared to RLBR and R2LTO, respectively with increasing sensor node residual energy. This is because the LACQRP agent has global information of the network topology and can quickly learn the best RT from the list of all RTs that maximizes the network lifetime. LACQRP is different from the RLBR and R2LTO that are distributed in nature and are constraint by the learning agent having local information for the entire network. This results in a delay in learning the optimal routing path and therefore degrades the network lifetime. Both RLBR and R2LTO consider the energy of sensor nodes, their corresponding distances, and hop counts to the sink in choosing the next forwarder. Subsequently, RLBR does not select a next forwarder that has a greater distance or hops count to the sink when compared to the current node. This makes the network lifetime of R2LTO to be higher than that of the RLBR. The network lifetime of LACQRP is also compared with RLBR and R2LTO when the initial sensor node energy is set as $1000\ J$ for increasing packet generation rate as shown in Fig. 3. The network lifetime of both protocols decreases as the packet generation rate increases. This is because the energy consumption of the sensor nodes increases as the packet generation rate increases. This will subsequently lead to the first sensor node to deplete its energy source on time. The LACQRP has a better network lifetime performance of 89.57% and 8.72% when compared to RLBR and R2LTO, respectively with increasing packet generation rates at the sensor nodes. This is because the LACQRP convergences quickly to the optimal RT that maximizes the minimum of the sensor nodes' residual energies. This is against the distributed routing protocols of RLBR and R2LTO that require more time to converge to the optimal routing path.

V. CONCLUSION

This paper presents the design of a lifetime-aware centralized Q-routing protocol for WSN to maximize the network lifetime. The sink of the WSN, which also acts as the controller as enabled by the SDWSN paradigm has global knowledge of the network information and enables the generation of all possible distance-based MSTs which are used as RTs. Q-learning is deployed at the controller to learn the RT that maximizes the lifetime for the first sensor node to deplete its energy source. The proposed protocol learns the best RT that optimizes the network lifetime for the scenario where all sensor nodes send equal packets periodically in each round to the sink. The proposed protocol has a better network lifetime when compared with the distributed RL routing protocols of RLBR and R2LTO. The limitation of the proposed protocol is that it depends on an algorithm that generates all MSTs of a graph. The problem of generating all MSTs of a graph is NP-hard (computational complexity is exponential). To be able to implement the proposed protocol in practice, future work will consider a sub-optimal solution (a solution that does not guarantee that all MST are found, in a reasonable time). When all or a subset of MSTs are found the controller will use them to learn which ones are optimal in terms of network lifetime.

ACKNOWLEDGMENT

Elvis Obi was sponsored by the Nigerian Petroleum Technology Trust Fund (PTDF) Overseas Scholarship Scheme.

REFERENCES

[1] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," IEEE Access, vol. 7, pp. 55916–55950, April 2019.

[2] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues," The Journal of Supercomputing, vol. 76, pp. 1–41, January 2020.

[3] J. Rischke, P. Sossalla, H. Salah, F.H. Fitzek, and M. Reisslein, "QR-SDN: Towards Reinforcement Learning States, Actions, and Rewards for Direct Flow Routing in Software-Defined Networks," IEEE Access, Vol. 8, pp. 174773–174791, September 2020.

[4] S. Buzura, B. Iancu, V. Dadarlat, A. Peculea, and E. Cebuc, "Optimizations for Energy Efficiency in Software-Defined Wireless Sensor Networks," Sensors, vol. 20, no. 17, pp. 1–23, August 2020.

[5] Y. Han, B. I. Rubinstein, T. Abraham, T. Alpcan, O. De Vel, S., Erfani, D.Hubczenko, C. Leckie, and P. Montague, "Reinforcement learning for autonomous defence in software-defined networking," In International Conference on Decision and Game Theory for Security, Springer, Cham., October 2018, pp. 145–165.

[6] P. M. Egidius, A. M. Abu-Mahfouz, and G. P. Hancke, "A comparison of data aggregation techniques in software-defined wireless sensor network," In 28th IEEE International Symposium on Industrial Electronics (ISIE), June 2019, pp. 1551–1555.

[7] F. Junli, W. Yawen, and S. Haibin, "An improved energy-efficient routing algorithm in software define wireless sensor network," In IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), October 2017, pp. 1–5.

[8] R. S. Sutton, and A. G. Barto, "Reinforcement learning: An introduction," 2nd ed. Cambridge, MA, USA: MIT press, 2018.

[9] J. A. Boyan, and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," In Advances in neural information processing systems, 1994, pp. 671–678.

[10] P. Wang, and T. Wang, "Adaptive routing for sensor networks using reinforcement learning," In 6th IEEE International Conference on Computer and Information Technology (CIT'06), September 2006, pp. 219–219.

[11] S. Dong, P. Agrawal, and K. Sivalingam, "Reinforcement learning based geographic routing protocol for UWB wireless sensor network," In IEEE Global Telecommunications Conference (GLOBECOM), November 2007, pp. 652–656.

[12] B. Karp, and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," In Proceedings of the 6th annual international conference on Mobile computing and networking, August 2000, pp. 243–254.

[13] J. Yang, H. Zhang, C. Pan, and W. Sun, "Learning-based routing approach for direct interactions between wireless sensor network and moving vehicles," In 16th IEEE International Conference on Intelligent Transportation Systems (ITSC), October 2013, pp. 1971–1976.

[14] A. P. Renold, and S. Chandrakala, "MRL-SCSO: multi-agent reinforcement learning-based self-configuration and self-optimization protocol for unattended wireless sensor networks," Wireless Personal Communications, vol. 96, no. 4, October 2017, pp. 5061–5079.

[15] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," In Proceedings of the 7th ACM conference on embedded networked sensor systems, November 2009, pp. 1–14.

[16] W. Guo, C. Yan, and T. Lu, "Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing," International Journal of Distributed Sensor Networks, February 2019, vol. 15, no. 2, pp. 1–20.

[17] S. E. Bouzid, Y. Serrestou, K. Raoof, and M. N. Omri, "Efficient Routing Protocol for Wireless Sensor Network based on Reinforcement Learning," In 5th IEEE International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), September 2020, pp. 1–5.

[18] T. Yamada, S. Kataoka, and K. Watanabe, "Listing all the minimum spanning trees in an undirected graph," International Journal of Computer Mathematics, vol. 87 no. 14, November 2010, pp. 3175-3185.

[19] J. Eisner, "State-of-the-art algorithms for minimum spanning trees-a tutorial discussion," 1997.

[20] G. Oddi, A. Pietrabissa and F. Liberati, "Energy balancing in multi-hop Wireless Sensor Networks: an approach based on reinforcement learning," 2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), July 2014, pp. 262–269.

[21] A. Hagberg, P. Swart, D.S. Chult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), January 2008.