This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

Improving the software defined wireless sensor networks routing performance using reinforcement learning

Muhammad Usman Younus, *Member, IEEE,* Muhammad Khurram Khan, *Senior Member, IEEE,* Abdul Rauf Bhatti

***Abstract*—Software defined networking (SDN) is an emerging architecture used in many applications because of its flexible architecture. It is expected to become an essential enabler for the Internet of Things (IoTs). It decouples the control plane from the data plane, and the controller manages the whole underlying network. SDN has been used in wireless sensor networks (WSNs) for routing. The SDN controller uses some algorithms to calculate the routing path; however, none of these algorithms have enough ability to obtain the optimized routing path. Therefore, reinforcement learning (RL) is a helpful technique to select the best routing path. In this paper, we optimize the routing path of SDWSN through RL. A reward function is proposed that includes all required metrics regarding energy efficiency and network quality-of-service (QoS). The agent gets the reward and takes the next action based on the reward received, while the SDWSN controller improves the routing path based on the previous experience. However, the whole network is also controlled remotely through the Web. The performance of the RL-based SDWSN is compared with SDN-based techniques, including Traditional SDN and Energy-Aware Software Defined Networking (EASDN), QR-SDN, TIDE and non SDN-based techniques, such as Q-learning and RL-based Routing (RLBR). The proposed RL-based SDWSN outperforms in terms of lifetime from 8% to 33% and packet delivery ratio (PDR) from 2% to 24%. It is envisioned that this work will help the engineers for achieving the desired WSN performance through efficient routing.**

Keyword: Reinforcement Learning, Wireless Sensor Networks, Internet of Things, SDWSN, RL-based WSN, Energy Optimization, Routing.

I. INTRODUCTION

There are tiny sensor nodes in wireless sensor networks (WSNs) that may be stationary or mobile nodes deployed in a dynamic environment. Each sensor node consists of a small power source, transmission, and processing units [1]. The WSN is an application-oriented information-centric network used in many applications, including defense, environmental monitoring (i.e., light, temperature, humidity, and vibration), military, and health [2], [3].

Muhammad Usman Younus is with the Ecole Doctorale Math ematiques, Informatique, Telecommunications de Toulouse, UniversityPaul Sabatier,31330, Toulouse France (e-mail: usman1644@gmail.com).

M. K. Khan is with the Center of Excellence in Information Assurance, College of Computer and Information Sciences, King Saud University, Saudi Arabia (e-mail: mkhurram@ksu.edu.sa) .

Abdul Rauf Bhatti is with the Department of Electrical Engineering, Govt. College University ,Faisalabad Pakistan (e-mail: bhatti_abdulrauf@gcuf.edu.pk) .

Much of the recent research work in the WSN area focuses on low-cost and low-power networking solutions to execute the cooperative and collaborative tasks under rigorous computation and energy constraints. Therefore, the wireless sensor nodes operate for a long time without replacing their batteries in many applications [4]. Thus, the sensor nodes' energy consumption is considered extremely crucial in wireless network design. Routing is the core networking activity in WSNs that routes the sense data from source (sensor node) to destination (sink). It significantly impacts the network performance, such as energy consumption, delay, latency, and packet delivery ratio (PDR). In order to make WSNs more efficient, routing strategies, such as software defined networking (SDN) [5], [6] and Reinforcement Learning (RL) [7] , can be an excellent choice to get an optimized routing path in WSNs.

SDN is an emerging architecture that manages the network efficiently. It has three planes that include a data plane, a control plane, and an application plane. It decouples the control plane from the programmable data plane. The idea behind it is to manage and utilize network resources efficiently. In SDN, the network is controlled through a centralized controller (control plane) that can globally view the whole underlying network and packet forwarding devices on the data plane. The control plane is responsible for routing, traffic management, and fault recovery; however, the data plane manages the packet delivery. SDN uses the well-defined interfaces between each plane, such as the southbound interface between the data plane and the control plane, while the northbound interface between the control plane and application plane. It also uses the OpenFlow communication protocol used by the SDN controller to communicate with data plane devices i.e., sensors, switches, and routers. The use of SDN in the Internet of Things (IoTs) is increasing day-by-day because of its efficient structure that can efficiently manage and control the billion of network devices [8]. Software defined IoT uses in a different networs such as edge networking, access networking, core networking, and data center networking. IoT is also comprised of several wireless devices managed through SDN; however, the IoT network is still facing some issues and challenges related to security and scalability [9]. SDN is also used in WSN, known as a software defined wireless sensor network (SDWSN), that makes it robust and well organized. However, SDWSN still has some limitations, such as finding the best routing path. It can be solved through a learning technique called RL. RL is an efficient method for a real-time path

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

optimized path in reducing energy consumption and delay, and increasing PDR.

RL is a type of machine learning (ML), in which a learner is known as an agent who learns the optimal policy (π) based on rewards and experiences. It satisfies the Markov property, called the Markov Decision Process (MDP) [7], [10], which gives the concept of RL. RL agent in the environment can be explained as (S, A, P, R), where S is a set of states $\{s_1, s_2, s_3, \ldots s_m\}$, A is the set of actions $\{a_1, a_2, a_3, ..., a_m\}$ that agent goes form one state to another state, while P is the state transition probability. The agent A will get positive or negative feedback after each round, called reward $R(s'|s, a)$, and select the next action according to the reward received, as shown in Figure 3. The reward can be maximized by optimizing the policy $\pi : A \leftarrow S$.

The combination of both SDWSN and RL can efficiently manage the network and improves the network performance. The RL-based SDWSN architecture is shown in Figure 1. Some artificial intelligence (AI) techniques (Deep Reinforcement Learning) are used in SDN for the network self-learning that can control the network efficiently [11]. However, the WSN network can be controlled remotely by the IoTs [12], [13]. It is a new paradigm that enables any device to connect with another device through the Internet. The IoT architecture consists of low-processing devices called nodes, the local controller that collects the data from nodes, and the cloud layer that monitors and controls the nodes remotely known as the global control mechanism. The WSN framework for IoT application is shown in Figure 2. For the local controlling and optimization, we use the SDN architecture and RL for optimizing the routing path. However, we are also controlling the sensor node globally through the Web (Internet) as shown in Section VI. The contribution of this paper is summarized as follows:

- We propose a reward function to optimize the routing that selects the best path from the routing list. It leads to reduce the energy consumption of the network, improve the network lifetime, and packet delivery ratio (PDR).
- We also propose two algorithms to improve the network performance (i.e., energy efficiency, PDR, etc.). These algorithms are applied to an intelligent SDN controller, which can efficiently control the data plane devices.
- The loop-free communication is established through Spanning Tree Protocol (STP).
- A real-time experimental platform is developed using Raspberry Pi, where SDWSN routing based on RL experiment is carried out.
- A web-based dashboard is also developed to control and analyze the sensor nodes data remotely.
- The proposed RL-based SDWSN technique is compared with RL-based and non RL-based techniques/algorithms on a real-testbed.

The rest of this paper is organized as follows: Section II provides the literature review related to RL-based SDWSN and non RL-based routing techniques. In Section III, RL is explained. However, the reward function is also explained in detail in Section III. Section IV provides the detail of the energy consumption model used for experimental work. In Section V, the proposed methodology and algorithms for the SDN controller and sensor node are explained. Section VI provides the detail of the experimental testbed, and also compares the proposed RL-based SDWSN algorithm with different routing protocols. Finally, the paper is concluded in Section VII.

II. Related Work

Routing can optimize the performance of WSN, such as energy efficiency and Quality-of-Service (QoS) parameters. In optimal routing, we adopt SDN architecture and propose some algorithms to improve SDN-based network performance through routing. While, sometimes, the SDN-based network has poor real-time performance. To resolve this issue, RL can play an essential role in optimizing network performance. This section divides the routing approaches into two different groups: SDN-based routing approaches and RL-based SDN routing approaches.

A. Non RL-based routing techniques

In [14], the author proposed an energy-aware routing algorithm for the SDN-based WSN network. The author adopts the extreme assumption for implementation that includes the SDN controller knows the initial status data (i.e., position, energy, and neighboring nodes) of all sensor nodes. The controller also has direct access (e.g., the controller can send a routing table directly to each node) to all the nodes, which is practically impossible. A distance-based routing path is established by using a Dijkstra algorithm and changes it if any node runs out of energy. An energy-aware routing protocol is proposed for the SDN network know as EASDN [15]. This paper proposes three algorithms, including neighbor discovery algorithm, status data collection algorithm, and controller operation phase algorithm. In EASDN, the controller uses two parameters of distance and residual energy to calculate the routing path. The SDN controller sends a new routing table whenever he observes any node runs out of energy or is below the user's threshold. Raspberry pi is used to perform a real-time experiment as a sensor node. However, in [16], the author proposed a traffic control scheme that monitors the traffic behavior and prevents traffic congestion through Deep Packet Inspection (DPI). In another work [17] proposed a novel green datapath framework was proposed to reduce energy consumption by looking for energy-efficient routing paths for TCAM-based SDN. TCAM provides search operations for packet switching networks and is used by networking devices to speed-up the networking tasks, for example, packet classification. This study of green datapath focuses on TCAM usage in hybrid SDN networks by introducing dynamic voltage and frequency scaling (DVFS) power management technique. Hence, TCAM memory deploys the enriched routing functionality and also provides speed-up interaction between switches and controllers. However, in the [18], the energy-efficient architecture is proposed for SDWSN. It reduces data packet generation by content awareness and adaptive data broadcast of cached data. While, an SDN-based routing protocol is developed for the multihop wireless

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

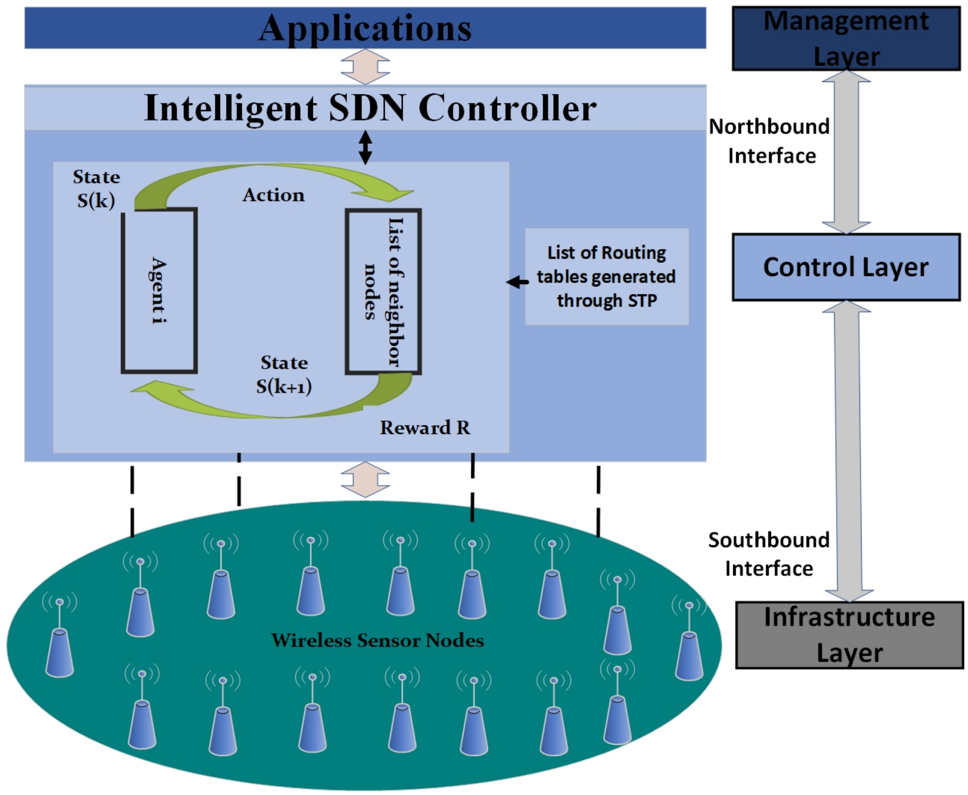


Fig. 1. RL-based Software Defined Wireless Sensor Network Architecture.

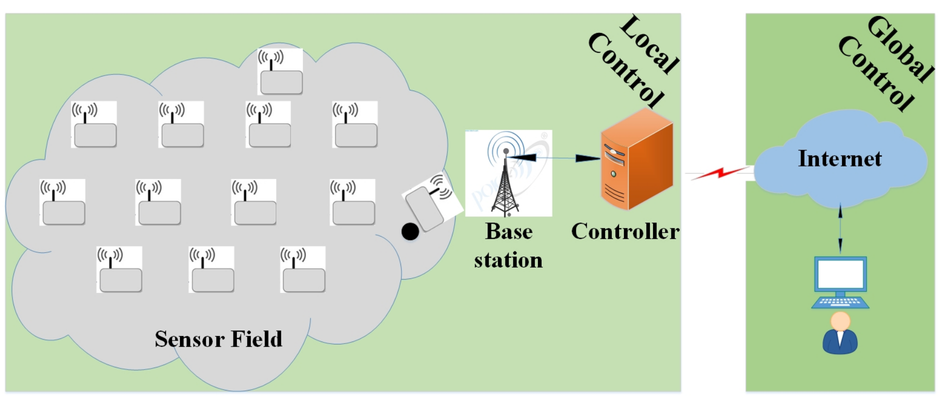


Fig. 2. Wireless Sensor network framework for IoT application.

network [19]. The routing path develops through residual energy and a minimum number of hop count for getting the shortest path. The model was developed using OPNET. The simulation results are compared with old routing protocols, such as Optimized Links State Routing Protocol (OLSR), and Ad hoc On-Demand Distance Vector (AODV). A software defined energy-aware routing (SD-EAR) is proposed in [20] to reduce the network's energy consumption. The network is divided into different zones or clusters, and each zone is controlled through the SDN controller. The SDN controller knows each zone topology, selects the energy-efficient path based on the global view, and knows each node's residual energy. While in [21], an energy-efficient routing algorithm is developed for SDWSNs. The proposed algorithm selects the control node to assign different tasks dynamically. The NP-hard definition is used to select control nodes. For tackling the NP-hard problem, an improved particle swarm optimization (PSO) algorithm is proposed. However, PSO often becomes more complicated in solving these types of issues.

SDN-based optical networks support huge IP traffic with great energy efficiency. Therefore, an SDN-based cross-network is proposed in [22] for joint energy efficiency designs from wired to wireless networks, potentially facilitating a large-scale deployment. This approach uses SDN to focus on designing the improved communication protocols, considering that the sensors are not constrained by the battery-based power. As energy efficiency has always been one of the important goals of cellular wireless communications. A great work made

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

by some authors in [23], focuses on the design of green cloud radio access network (C-RAN) through jointly assuming the power consumption of remote radio head (RRH) and transport network. In [24], the authors proposed an energy-efficient mechanism for transferring the power wirelessly to SDWSN. The proposed process specifies the minimum number of energy transmitters. The optimization problem is formulated to find the minimum number of energy transmitters that reduce the network's energy consumption. However, SDN can also improve WSN management. In [25], the authors addressed the many traditional and SDN-based protocol. This paper gives the inside detail of SDN-based management techniques for WSN. The benefits of SDN-based network management are discussed, some challenges of SDN faces are also explained. For Multi-Protocol Label Switching (MPLS) [26], the energy-aware routing and resource management model is developed using SDN. First, the SDN controller sets up multipath and then uses these pre-multipath paths (PMP) for routing. However, depending on traffic conditions, the SDN controller saves energy by turning the PMP on or off. Load balancing and route resizing are often used to utilize network resources and effectively minimize network energy consumption. However, global power management is achieved through SDN in [27]. In the proposed technique, the traffic is rerouted and adjust the network workload in different links. A network topology is constructed according to routers connection. The In teger linear programming model (0-1) minimizes the integrated chassis and line-cards power. Two algorithms (alternative greedy algorithm and global greedy algorithm) are proposed for efficient link utilization and packet delay reduction.

WSN network reliability and traffic load management are also a critical problem discussed in [28], [29]. A system called improved software defined wireless sensor network (Improved SD-WSN) is proposed to solve the WSN reliability problem. The proposed structure tackles heterogeneous network management and network coverage problem, which can increase network reliability. Where the traffic load minimization (TLM) problem in software defined wireless sensor Networking is resolved through the flow splitting optimization (FSO) algorithm. In [29], two methods are used to reduce the traffic load issue; the first one is used to select the optimal relay sensor nodes, while the second one is responsible for the transmission of optimal splitting flow.

A recent study in [30] confirms the great potential of information and communications technologies (ICTs) to achieve social and environmental advantages. Such systems make a significant contribution for maintaining user connectivity requirements, ensuring the desired user experience through the deployment of new frameworks. Despite the abundance of energy-saving technologies, ICTs systems are critical regarding the current and future energy consumption of telecommunication networks. Most of the evidence speaks against flattening or reducing ICT power. Great efforts have been made to improve one or more specific features of energy-efficient based systems. Energy efficiency has gained great importance in the green spectrum. Green topics are multidisciplinary in nature, encompassing energy-related issues and the broader context of the environmental impact of the IoTs. Green IoTs can be integrated into standalone lighting systems [31], from green energy harvesting to smart energy management, which means significant energy savings. In machine-to-machine (M2M) systems, some machine nodes are battery-powered and need to operate for long periods of time without battery replacement. Therefore, it is very important for such nodes to be energy efficient. To that end, researchers have come up with several techniques to make these nodes energy efficient. To reduce the energy consumption, a wake-up/paging strategy is proposed in [32] by activating only those nodes ready to transmit and receive. Using a more efficient XML interface, core links, and protocol buffers, a security technique is used to control sleeping devices from attackers.

B. RL-based routing techniques for software defined wireless sensor networks

RL-based routing protocols are used to improve network performance. A reinforcement learning-based routing protocol is defined in [33], called Reinforcement Learning-based Routing (RLBR). The authors suggested the reward function in RLBR, which uses neighbors' distance, residual energy, and hop count to sink to measure the reward. It optimizes WSN routing, which helps to increase the network lifetime. For the first time, Watkins proposed an estimation action function called Q-function in [34] to reduce the time delay in the network. After each iteration, the Q-function was modified and converged to an optimum point after some iterations. The combination of RL and SDN, however, provides a stronger solution. A prototype is proposed for improving the energy efficiency and adaptability of SDNWSN by using RL [35]. For monitoring the environmental application, the prototype considers the energy, computational capabilities, and radio resources to optimize the WSN performance. RL is not adequately used to optimize SDWSN performance. However, in the SDN-based network, the SDN switches face the service placement problem that increases end-users' service costs. An algorithm named Q-placement[36] is suggested to reduce the end-user expense through reinforcement learning. It guarantees the efficiency of the network and increases its convergence rate. It does not, however, concentrate on the issue of energy optimization for the network. For multimedia based SDNs, a reinforcement learning-based LearnQoS system is proposed in [37]. Services based on video are important because it has become an integral part of the end-user life. By employing policy-based network management (PBNM), the proposed LearnQoS framework increases the video QoS. A video streaming framework based on SDN is suggested that uses reinforcement learning to handle the flow of network traffic [38]. The proposed approach learns the optimized routing path and tries to minimize the controller cost, packet loss rate while maintaining the video quality. In [39], QoS aware adaptive routing is proposed to enhance the SDN-based network throughput and reduce the delay and packet loss ratio. The proposed routing protocol consists of multi-layers hierarchical architecture. There are three kinds of controllers to control the network: super controller, domain controller, and slave controller. The network's performance is enhanced

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

through reinforcement learning. The designed reward function optimizes the QoS and achieves some metrics, such as QoS-provisioning packet forwarding, time-efficiency, and fast convergence rate. Time-relevance deep reinforcement learning (TIDE) is another technique presented in [40] to improve the network QoS. An Artificial Intelligence (AI)-based layer is introduced to control the network associated with the SDN controller. In TIDE, only QoS parameters are considered to make routing decisions. However, in [41], the QR-SDN routing approach provides multiple routing paths while preserving flow integrity. An other algorithm [42], that is based on Q-learning, which sets up the Q-tables. The goal of the proposed algorithm is to find the best link for data forwarding. But the main concern is QoS. Reward value is fixed against the QoS performance of each link, so that if the QoS performance of the link is between 0% to 30% then the reward value will be 50, if the QoS performance of the link is between 31% to 60% then the reward value will be 100; otherwise it will be 150. In the future, deep learning will be combined with Q-learning, which is called deep Q-learning. In the modern era of information technology, most services mainly rely on cloud infrastructure. The network flow includes the mice and elephant flows in the cloud environment. For the management of these flows, SDN-based approaches are used to allocate the network resources efficiently. However, due to Ternary Content Addressable Memory (TCAM) 's limited capacity in OpenFlow enables switches, the management of flow is a big issue. Still, it is challenging to find the useful forwarding rule in the flow table, which rules need to be processed by the SDN controller. It is required to investigate the useful flow items in the flow table and exclude the remaining flow items. By doing this, control plane overhead can be reduced. For this reason, it is helpful to use Reinforcement learning in [43]. To minimize the overhead between the SDN controller and the switch, the author proposes an algorithm based on RL and an algorithm based on deep RL. Cybersecurity in SDN also becomes a critical issue. Different forms of causative attacks are studied using RL to learn how to react. RL-based algorithm is proposed in [44] to prevent the SDN network. However, the proposed algorithm's basic purpose is to make the SDN platform's autonomous defense system. As IoT is a new paradigm to interconnect the large-scale wireless devices and has a wide range of application prospects. In [45], AI-based solutions have been addressed related to spectrum access and random access. In addition, deep learning algorithms can make IoT smarter and friendly, as the development of AI is crucial for IoT, which can be strongly supported in different ways. It also facilitates AI's development direction by proposing a DQN for conducting efficient online training for RL. A learning-based algorithm protects WSN from any external attack in [44]. Each sensor node plays a critical role in WSN, and it should be protected from external attacks. The author of [44] proposed a self-protected learning algorithm (SPLA) that can resist external attacks. In the proposed algorithm, Sensing Graph (SG) is used to model the problem, determining the minimum number of the nodes that can be protected. Each node is equipped with a learning automation. SPLA tries to find the minimum number of nodes that can be activated to protect the network. However, in [46], a Decentralized Swift Vigilance (DeSVig) framework is proposed to identify adversarial attacks in an industrial AI system. Due to decentralization, DeSVig improves the effectiveness of recognizing abnormal inputs.

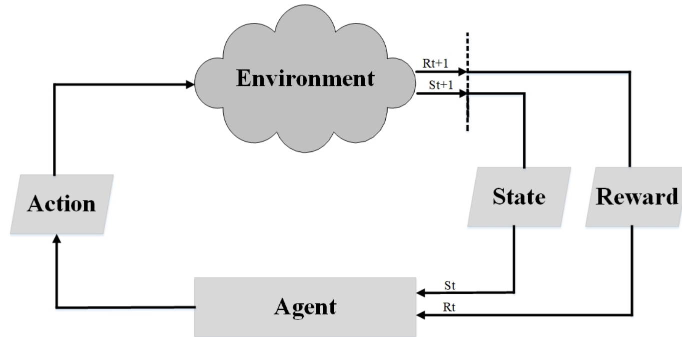


Fig. 3. RL Model.

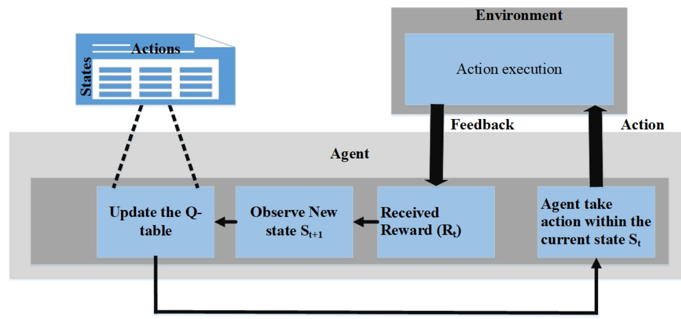


Fig. 4. Q-Learning block diagram.

III. Q-LEARNING

Q-learning is a model-free learning approach first proposed by Watkins [34]. It is used to estimate $Q'(s_t, a_t)$, called Q-function, and its learning technique is called Q-learning. In Q-leaning, a learner is known as an agent that selects an action according to the current state by interacting with the environment and getting either positive or native feedback based on the action called reward and calculates the Q-value. In the next state s_{t+1}, the agent selects an action based on the previous reward. After some rounds, the agent is trained and converges to the optimal point. In Q-learning, the Q-value is updated after each iteration, as shown in Figure 4. The update Q-value is defined in equation 1.

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[R_t + \gamma max Q'(s_{t+1}, a_{t+1})] \quad (1)$$

$Q(s_t, a_t)$ is the Q-value that the agent takes an action at in state st and gets an immediate reward R_t, where $maxQ'(s_{t+1}, a_{t+1})$ is the maximum value that can get in the next state s_{t+1}. α is the learning rate that determines the what extend the newly acquired information update to the old information and the range is $(0 < \alpha \leq 1)$. Where the γ is the discount factor that determines the importance of future reward and its range is $(0 < \gamma \leq 1)$.

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

A. Q-Routing Protocol

It comes from Q-table, which uses the Q-learning method to route the data packets [34]. In Q-routing, the first Q-matrix of node i is initialized. The initialization of the Q-matrix may be random. Then node i sends a packet P to neighbor node j. Node i selects the forwarder node j with the lowest Q-value because of low distance up to destination d. Low distance has a low delivery delay (end-to-end delay), which is routing cost. $Q_i(d, j)$ is the delivery delay calculated from equation 3 and t_j is the estimated remaining time in the trip that node i immediately gets back $j's$ and it can be found by equation 2.

$$t_j = \min_{k \in Ng(j)} Q_j(d, k) \tag{2}$$

Let $Q_i(d, j)$ is the estimated time that the node i takes to deliver the packet P up to node j using distance, and it can be calculated by equation 3.

$$Q_i(d, j) = (1 - \alpha) * Q_i(d, j) + \alpha * (q_i + Tx + t_j) \tag{3}$$

q_i is the time to spend packet P in the queue of node i, α denotes the learning period, and Tx represents the time used for transmission between node i to node j.

The Q-routing algorithm is given below in Algorithm 1.

Algorithm 1 Q-Routing Algorithm

```
begin Initialized the Q-matrix (Q(x,y)))

while (Until the terminal state is reached) do
    if Is data packet ready to send then
        Calculate the Q-Value.
        Select next hop j with lowest Q-value.
        Send the packet to selected forwader node j.
        Calculate the delivery delay time using Eqns (2) and (3).
        Update the node i delivery delay time.
    end
end
```

B. Q-learning based routing in SDWSN

Consider a network containing the n sensor nodes, and each sensor senses the environment and selects one neighboring node j to send the data packet up to the destination d. For selecting the best optimal path, a Q-learning uses the previous experience to select routing paths. In the next epoch, the sensor node i selects the best path to send the destination's data packet. We use the Q-learning for the route selection in SDWSN. In the SDN-based network, the controller controls the whole network. It selects the best path according to the implemented algorithm. We use the spanning tree protocol (STP) for getting all possible routing paths at the controller. SDN controller selects a routing table from the list of paths with Q-learning and sends the nearest neighboring node. Q-learning includes the state, action, reward, and Q-value, which are defined as follows:

1) States: Let the S is the set of states $S = \{s_i, s_j, s_k, ..., s_n\}$ which means that after each round, the controller selects an other routing table from routing table list.

2) Actions: An action is an act where any path from the set of routing paths $\{p_1, p_2, p_3, ..., p_n\}$ is selected by the agent and sends it to the neighboring node.

3) Reward Function: The reward R_t is the feedback after agent action a_i, which can be positive or negative. It has much importance in Q-learning because the next action will be based on Q-value, which varies according to the received reward. The optimal strategy will be obtained after each iteration. The reward function uses different metrics to calculate the reward. In [34], Watkins only uses packet delay in the reward function that is not enough for the energy-efficient network. This paper uses different reward function metrics, including distance to sink, the number of hops from neighboring nodes to destination/sink, node residual energy, and packet success ratio. Each metric's weight is considered in the proposed reward and took the sum of all nodes reward defined in equation 4.

$$R_t = \sum_{i=0}^{N} (w_1 {*} d(s_i) + w_2 {*} H(s_i) + w_3 {*} E(s_i) + w_4 {*} p(s_i)) \tag{4}$$

$$d(s_i) = \frac{Dist\_to\_Negb(s_i)}{Dist\_to\_Sink(s_i)} \tag{5}$$

Where the $Dist\_to\_Negb$ is the distance from neighboring node to destination/controller and $Dist\_to\_Sink(s_i)$ is the maximum distance to sink.

$$H(s_i) = \frac{H_j(s_i)}{H_{max}(s_i)} \tag{6}$$

$H(s_i)$ is the ratio of the number of hops from neighboring nodes to destination and maximum possible hop counts.

$$E(s_i) = \frac{E_R(s_i)}{E_{total}(s_i)} \tag{7}$$

$E(s_i)$ is the ratio of remaining energy $E_R(s_i)$ to the total energy $E_{total}(s_i)$.

$$p(s_i) = \frac{p_{ack}(s_i)}{p_{send}(s_i)} \tag{8}$$

$p(si)$ is the ratio of packet acknowledgment $p_{ack}(s_i)$ received by the neighboring node to the total packet $p_{send}(s_i)$ sent.

All these factors are used in the reward function to calculate the reward, which is given in equation (5) to (8).

The intelligent SDN controller calculates the reward after each round. After getting a reward, Q-value is updated. The new Q-value of the state-action (s_t, a_t) pair is determined by:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha[R_t + \gamma max Q(s_{t+1}, a_{t+1})] \tag{9}$$

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

IV. Energy consumption model

Each sensor node contains its small power supply that is required for transmission and reception. Each node consumes energy during transmission and reception. In order to calculate the energy consumption during data communication, we consider the first-order energy model [47]. In this model, two types of channels are used to calculate path loss: One is a free space model, and the second is multipath fading [15], [21]. The model selection is based on the distance between the transmitter and the receiver. If the transmitter-receiver distance is less than or equal to the threshold d_0, then the free space model is selected; otherwise, multipath is selected. The following equations measure the energy consumption due to the transmission of each packet:

$$E_{tx}(b_l, d) = \begin{cases} b_l * E_{fs} * d^2 + b_l * E_{elec} & d \leq d_0 \\ b_l * E_{mp} * d^4 + b_l * E_{elec} & d > d_0 \end{cases} \tag{10}$$

E_{tx} is the energy consumption due to transmission, b_l is the total packet length in bytes, and d is the distance between the sender and receiver. Since E_{fs} denotes the energy consumption due to free space and E_{mp} is the energy consumption of multiple paths. While E_{elec} refers to the energy consumption due to circuit processing before the transmission of a data packet and after the reception of a data packet. As d_0 is the distance threshold, which can be found by the following equation.

$$d_0 = \sqrt{(E_{fs}/E_{mp})} \tag{11}$$

The energy consumption due to the reception of a data packet is calculated by:

$$E_{rx}(b_l) = b_l * E_{elec} \tag{12}$$

E_{rx} is the energy consumption due to the receiving bytes.

V. Proposed method

In the WSN, each sensor node collects data from the environment according to the network design and communication range and sends it to the sink through a single/multihop. In traditional WSN, each node broadcasts a control packet for getting the neighboring node information. The network consumes a lot of energy because of broadcasting, and also generic algorithms cannot optimize the path. SDN is an emerging architecture in which the data plane is separated from the control plane. It manages the network through a centralized controller that can view the network globally. However, the SDN controller algorithms are inefficient in real-time routing path optimization. RL is an efficient technique for learning that can optimize the real-time routing route. The best choice for optimizing the WSN routing path is the combination of both SDN and RL. This paper uses RL on the SDN controller to choose the routing list's best routing path and change the routing path if it finds the path to be bad.

We propose a reward function in our energy optimization approach including all the required parameters related to network performance. A reward function consists of the distance to sink, the number of hops to sink, the residual energy, and the packet success rate. To optimize the WSNs routing path, this algorithm is divided into two parts: one for the intelligent SDN controller and the other for the sensor nodes. On the controller side, neighboring nodes are found after initializing the controller node that collects the entire network's status data, as shown in the first phase of Figure 5. The controller generates all possible routing paths through STP [48], [49]. It selects one routing table from the list with Q-learning and sends it to the neighboring nodes, as shown in the second phase of Figure 5. After each epoch, the controller gets the status data of the sensor nodes and calculates the reward. The intelligent SDN controller changes the routing path according to system feedback in terms of reward, as shown in the third phase of Figure 5. If the reward is negative, it will decrease the network's performance and change the path; otherwise, it maintains the same path. The controller also continuously monitors each node's remaining energy. If any node has energy less than the threshold, it is excluded from the node list, recalculates the routing path list using STP, selects a routing table from the list, and sends it to the neighboring node. The algorithm for the SDWSN controller is shown in Algorithm 1.

However, on the node side, the first neighbor discovery period starts, and each node broadcasts the Hello packet to find the neighbors, as shown in Algorithm 2. This process continues up to a specific time and a maximum number of an acceptable neighbor threshold. After the neighbor discovery period, each node shares the status data with its neighboring nodes. The status data of each node reaches its destination (controller) through multihop communication. Each node receives a routing table from the SDN controller and sends a data packet according to the routing table provided by the intelligent SDN controller. The sensor node calculates the energy consumption using a mathematical model at the end of each round; and sends the residual energy status to the controller through a data packet. In the case of the relay node, it first checks the node's remaining energy; if it is greater than the threshold, it accepts otherwise send a low energy message to the controller and disconnect from the network. While the computational complexity of the energy-efficient algorithm of both controller and node side is also examined, the controller side runs n times operation until the last node dies, and the complexity of the whole controller side operation is $O(n)$. While, on the node side, the first operation remains valid up to two threshold approaches t_{nbr} and NBR_{max}. This operation runs n times while the two inside operations take n times that receivers the status data of neighboring nodes, and other operations calculate the energy consumption of node up to "Required Energy for Tx load". The whole operation contains $n(n+n)$ iterations. So the overall node side complexity is $O(n^2)$. The global optimization problem is NP-hard, and the majority of previous solutions have been heuristic algorithms that are slower than the proposed algorithms.

VI. Experimental Results and discussion

This section will discuss the performance of the proposed routing technique RL-SDWSN in terms of lifetime and PDR. The performance of RL-SDWSN is compared with previous

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

protocols, such as Q-routing, Reinforcement Learning-Based Routing (RLBR), T_SDWSN, EASDN, QR-SDN, and TIDE implementation is given in the next subsection. Q learning and RLBR are based on RL, but T_SDWSN and EASDN depend on the SDN architecture. However, QR-SDN and TIDE use RL to control the flow in an SDN-based network. Q-routing, RLBR, T_SDWSN, and EASDN routing protocols are used to optimize the energy consumption of WSN; however, QR-SDN and TIDE are used to optimize the network QoS. In our experimental setup, we use raspberry pi for real-time experimental work and create a wireless ad hoc network. Each raspberry pi (node) contains a wireless card using the IEEE 802.11ac standard. The topology used for experimental work is shown in Figure 6. All nodes are placed in the same location, communicate using logical distance, and access the neighboring node based on distance threshold.

Algorithm 2 Algorithm for SDN Controller

Input Network status data including total number of edges, Vertex, G=(V, E), STP, Reward function, Learning rates.
Output Set of routing paths.
Initialize the controller.
Assign the IP to controller.
Controller discover the neighboring nodes.
Collect the status data from all sensor nodes within the threshold and N_{max}.
N_{max} is the maximum number of nodes that can be possible neighbors; however, the threshold is the time threshold.
Calculate the routing table using STP.
RT $\leftarrow \{x_1, x_2, x_3, ..., x_n\}$.
SDN Controller$\leftarrow RT$.
Initially, the Q-value is considered as the worst case where all the nodes die without sending any data.
Select one routing table randomly from the routing table list.
Calculate the Reward by using equation 4.
Calculate the Q-Value using equation 9.
Update the Q-value.
while *(Received node data upto last node die)* **do**
    if *($E_{node}^{residual} < TH$)* **then**
        Exclude that node from the list.
        Recalculate the routing tables using STP.
        $RT \leftarrow \{x_1, x_2, x_3, ..., x_m\}$.
        Select one routing path from the list and send it to the neighboring node.
    end
    else
        RT $\leftarrow \{x_1, x_2, x_3, ..., x_n\}$.
        SDN Controller$\leftarrow RT$.
        Estimate the PELT.
        Calculate the Reward by using equation 4.
        Calculate the Q-Value using equation 9.
        Select one routing table with highest Q-value.
        Update the Q-value.
        end
    end

TABLE I
EXPERIMENTAL PARAMETER TABLE.

Parameters	Value
Initial Energy	5J
E_{elc}	50 nJ/bit/m^2
E_{fs}	100 pJ/bit/m^2
E_{mp}	0.0013 pJ/bit/m^4
Data packet size	100 bytes
α	0.3,0.5
γ	0.5

Algorithm 3 Algorithm for Sensor Nodes

Input Routing paths that receive from controller.
Output Each node $\{S_1, S_2, ..., S_n\}$ sends the collected information to the controller related to energy and QoS.
Initialize the nodes $S = \{S1, S2, \ldots, Sn\}$.
Assign the IP to controller.

if *(RE $> E_{Threshold}$)* **then**
    S $\leftarrow$parameter setting from controller
    Search the neighboring node
    while *(time $< t_{nbr}$) & len (My Neighbor list) $< NBR_{max}$* **do**
        if $t_{tbt} < t_{max\_allowed}$ **then**
            Broadcast the Hello packet
        end
        On *(Reception of Hello packet for neighbor)*
        if *(Nbr node id **not** exist in "Nbr list")* **then**
            Add into the neighboring list.
        end
    end
    Calculate the energy consumption of nodes after each Tx and Rx control and data traffic.
    while *(t $< t_{threshold}$) and (response number $< N_{max}$)* **do**
        Send the status data to neighboring node.
        On Reception of Status data packet **do**
        if *(source address **not** exist in the "node list")* **then**
            Add the sender node address into list.
            ++ Status data response number.
        end
    end
    while *(RE <Required Energy for Tx load)* **do**
        Tx & Rx the both control and data traffic .
        Calculate the energy consumption of nodes after each Tx and Rx control and data traffic.
    end
    Send the notification of low energy to controller through neighboring node and disconnect.
end

In our experimental work, the communication between the nodes is real-time, but the energy consumption is calculated by simulation. The reason to use the simulation for the calculation of energy consumption is that raspberry pi could not directly calculate the energy consumption. It requires an external module to calculate energy consumption (i.e.,

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

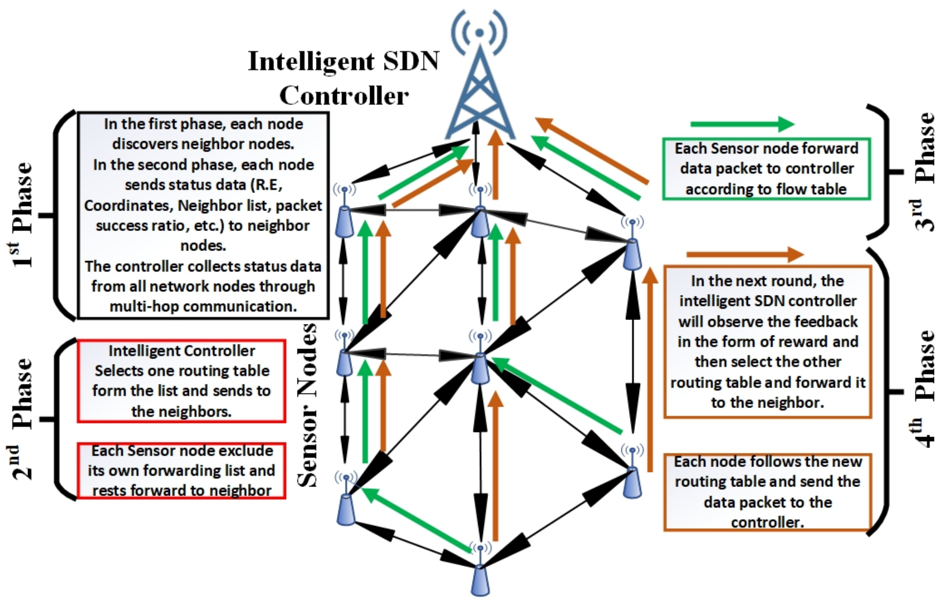


Fig. 5. Establish the route according to receive the flow table from the controller example.

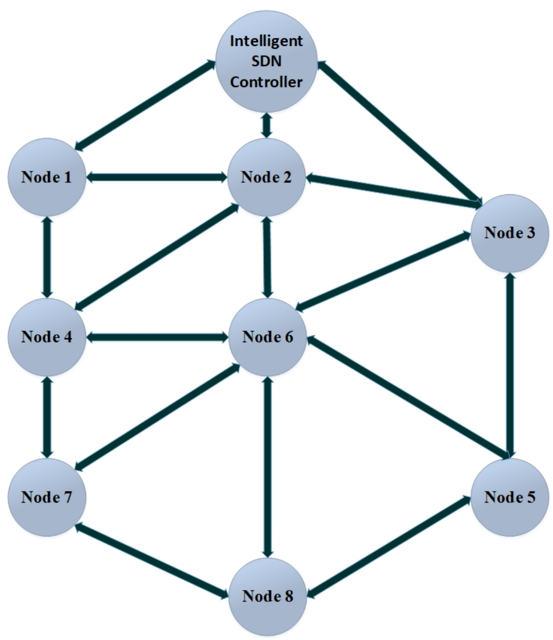


Fig. 6. Topology used for experimental work.

MoPi). However, the SDN controller energy consumption is not considered here, and it is assumed that the controller has infinite energy. The simulation parameters are given in Table I. In the parameter setting, we took two different α learning rates to observe the impact on network performance. In each α experiment, we consider three definitions of the lifetime (the first node dies, 50% node dies, and the last node dies) and two different deployment areas (100m*150m and 200m*300m). Experimental settings, platforms, and performance metrics are explained in the following subsections.

A. Experimental Platform

We did our experimental work on a real testbed using Raspberry Pi and developed the program using Python 3.0. Raspberry Pi is a low powered, low cost, and small size single-board computer. It is used as a sensor node to transmit data up to the controller/sink and as a controller to control the underlying network that collects data from the sensor nodes in the SDWSN network. Raspberry pi has become more attractive because of its small size and used in many real-time applications such as WSN, cloud computing applications, robotic projects, and so on. In our experiments, we used the Raspberry pi B+ model, which has a powerful 1.4GHz x Cortex-A53 CPU and ARMv8 microcontroller with 1Gb RAM. On the software side, it supports a variety of operating systems, including Debian Linux-based operating system recommended by the Raspberry pi foundation, and also optimizes the Raspberry pi hardware. Compared with other models, the specifications of Pi B+ are marvelous. It also contains the wireless LAN card having IEEE 802.11ac for wireless communications. In our experimental work, we use Raspberry pi WLAN to create the wireless ad hoc connection. Another advantage of Raspberry pi is that it has a secondary memory, storing large amounts of data on a micro SD card. It has been deployed as an intelligent sensor node, but in our experimental work, we use it as an intelligent SDN controller, and the rest of the nodes are considered as normal nodes that collect the data from the environment and send it to the SDN controller.

B. Performance metrics

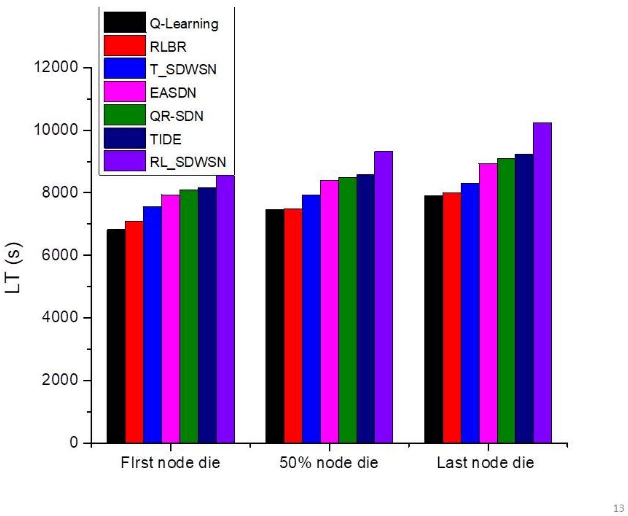
In this paper, we consider two metrics for results evaluation, as describe below:

1) Network Lifetime (LT):

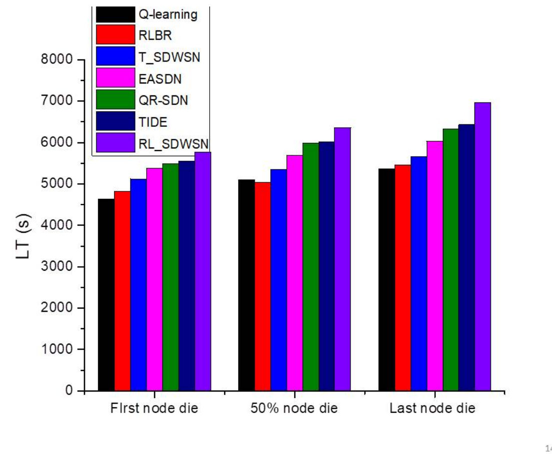
- The time until the first node runs out of energy.
- The time until the X% nodes runs out of energy.
- The time until the last node run out of energy.

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal



(a) Area 100m*150m.



(b) Area 200m*300m.

Fig. 7. Network Lifetime.

2) Packet delivery Ratio (PDR): The percentage of packet successfully deliver up to a destination is called PDR. It can be represented as:

$$PDR = [\frac{P_{TDP}}{P_{TSP}}] * 100 \quad (13)$$

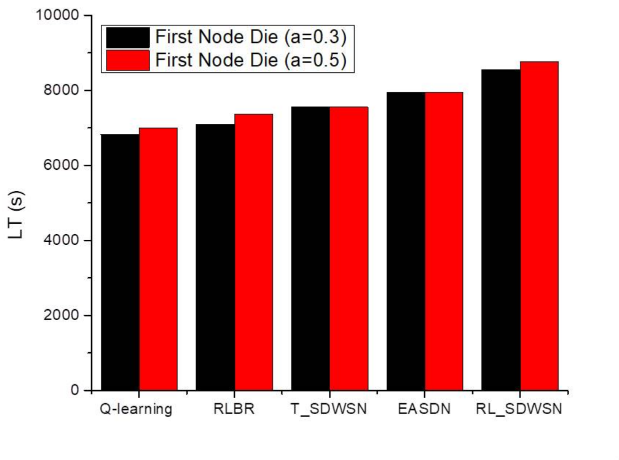
P_{TDP} is the total delivered packet, and the P_{TSP} is the total send packets from source to destination.

3) Lifetime: We performed our experimental work in two different scenarios. In the first scenario, the deployment area is 100m*150m, and in the second scenario, it is 200m*300m. We consider three different lifetime definitions: the first node dies, 50% node dies, and the last node dies. The experimental work for a lifetime is also performed with two different parameters (e.g., α=0.3 and 0.5). In the first part of the experimental work, we compare the lifetime of five different prior methods with three different lifetime definitions by considering two different areas (100m*150m and 200m*300m). The proposed technique RL-based SDWSN is compared with four previous techniques. In the second scenario of lifetime description, we compare all previous techniques with our proposed technique with different learning rates (α).

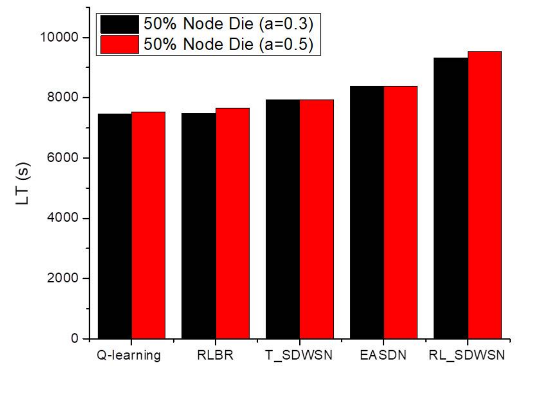
First, we use two dimensions to calculate the network's lifetime, as shown in Figure 7. In Figure 7(a), all three-lifetime definitions are taken into account, and the area of 100m*150m is taken for the experiment. The proposed RL-based SDWSN technique provides a better lifetime in all definitions. In the first node dies definition, the proposed technique RL-based SDWSN has a higher lifetime than Q-learning, RLBR, traditional SDWSN, EASDN, QR-SDN and TIDE. RL-based SDWSN is better than Q-learning because it considers all parameters (distance from the source to destination, number of hopes up to the destination, remaining energy, and QoS) related to energy optimization. But in Q-learning, it only considers the delay parameter and wastes a lot of network energy during path finding. Q-learning also uses the control packet to obtain new information about neighboring nodes. Usually, the control packet frequency is greater than the data packet frequency, causing the network lifetime to be shortened. However, in RLBR, it also has a shorter lifetime than RL-based SDWSN. In RLBR, a large number of control packets update the neighboring table. It is also not considered all the required parameters to find the optimized path in energy optimization. Both traditional SDN and EASDN give less lifetime than RL-based SDWSN because of non-efficient SDN controller algorithms. In traditional SDN, only distance is considered for the calculating the routing path and also considered some assumption (i.e., the controller can access to all nodes within one hop, and also knows the initial status data of all nodes, such as node energy, position, and neighboring node list) in a real-time network, that is not possible. SDN controller cannot find the optimized path based on distance only; for these reasons, the traditional SDN has a shorter lifetime than RL-based SDWSN. In EASDN, it takes into account two parameters (i.e., distance and residual energy) to calculate the routing path. Network quality is also not taken into account when calculating routing paths. However, both QR-SDN and TIDE networks are based on SDN architecture that is controlled through RL. These techniques do not focus on network energy consumption issues, and no energy measurement parameter is considered in the reward function. In contrast, RL-based SDWSN considers all required parameters in its reward function to find the best optimal path and change the path whenever the SDN controller observes another best path. Therefore, SDWSN based on RL has a higher lifetime than Q-learning, RLBR, traditional SDN, EASDN, QR-SDN and TIDE which are approximately 25%, 20%, 13%, and 8%,6% ,and 5.5%, respectively. In the second lifetime definition (50% node die), the proposed technique RL-based SDWSN is also outperforming as compared to SDN and RL-based techniques, as described earlier. RL-based SDWSN gives a higher lifetime than Q-learning, RLBR, traditional SDWSN, and EASDN, which is approximately 25%, 24%,17%, and 11%, respectively, as shown in Figure 7(a) However, in the third-lifetime definition, the proposed RL-based SDNWSN

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

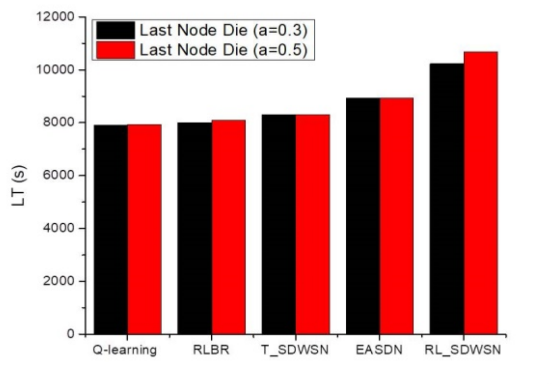
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal



(a) First node die.



(b) 50% nodes die.



(c) Last node die.

Fig. 8. Network Lifetime (Area 100m*150m) with different learning rates.

technique also has a higher lifetime as compared to SDN and RL-based techniques. RL-based SDWSN gives a higher lifetime than Q-learning, RLBR, traditional SDWSN, EASDN, QR-SDN and TIDE, which is approximately 33%, 28%,23%, and 14%, 12% and 11%, respectively.

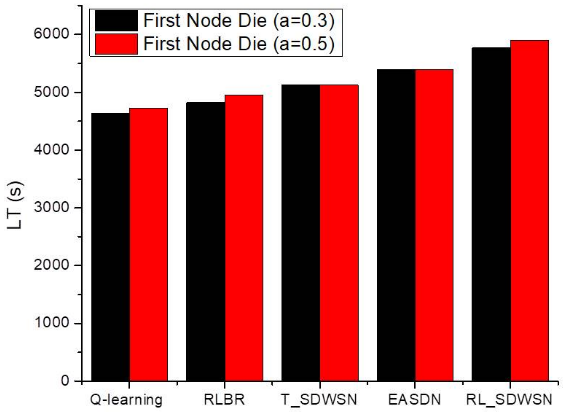
In the second scenario, we enhanced the deployment area 100m*150m to 200m*300m and observed the lifetime performance of the network. Lifetime decreases in this scenario compared to the small area scenario because the increased distance between nodes results in more energy consumption. The energy consumption of nodes depends on the length of the data packets and the distance between the nodes, as discussed in the energy consumption model section. However, the proposed technique improves the lifetime as compared to SDN and RL-based techniques, as shown in Figure. 7(b). With the exception of RLBR in the second life cycle definition (half node die), all techniques behave similarly to the first scenario. RL-based SDWSN has a higher lifetime than Q-learning, RLBR, traditional SDWSN, EASDN, QR-SDN, and TIDE approximately 8% to 33% in all definitions.

4) Comparison of Lifetime with difference learning rate: In the RL-based network, the learning rate has a significant effect on network performance. This section compares the two learning rates and looks at their impact on the network's lifetime. For this experimental work, we also consider three definitions of lifetime and two different areas. In the first scenario, the area of 100m*150m is selected. As can be seen from Figure 8, the learning rate is influencing the lifetime of the network. It increases the network lifetime of the RL-based network, while SDN-based network remains unchanged due to the lack of learning. All RL-based techniques (i.e., Q-learning, RLBR, and RL-SDWSN) improve the lifetime as the learning rate increases. From Figure 8(a),8(b), & 8(c) show that $\alpha = 0.5$ gives a higher lifetime because it quickly learns the network and can quickly optimize the routing path. In all definitions of a lifetime, when α changes from 0.3 to 0.5, the life cycle increases by 3% to 5%.

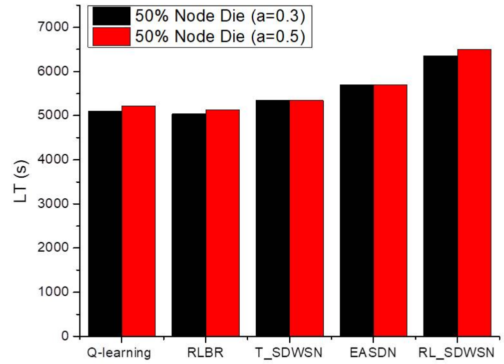
In the second scenario, we change the deployment area from 100m*150m to 200m*300m to see its impact on lifetime. As can be seen from Figure 9(a),9(b), & 9(c) , as the area increases, the lifetime of the network also decreases because

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

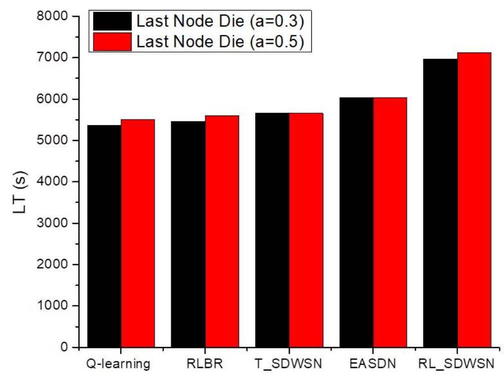
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal



(a) First node die.



(b) 50% nodes die.



(c) Last node die.

Fig. 9. Network Lifetime (Area 200m*300m) with different learning rates.

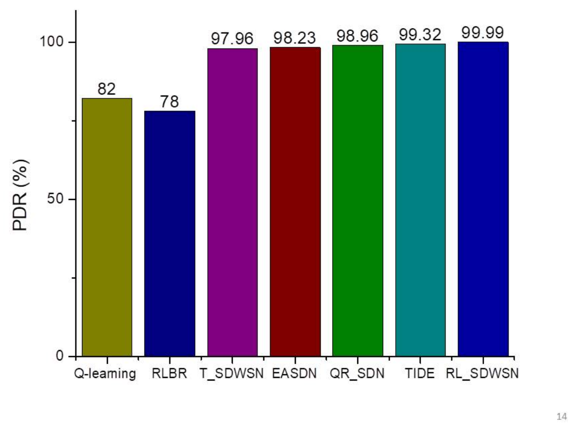
of the distance between nodes increases, which increases the energy consumption of the network. However, we can also see from these results that the learning rate also affects the network lifetime. It increases the lifetime of RL-based techniques by 3% when α increases from 0.3 to 0.5. We also observed that the proposed technique RL-based SDWSN has a higher lifetime than SDN and RL-based networks.

5) Packet Delivery Ratio (PDR): This section discusses the packet success rate, how many packets reach their destination, and how much are lost in transmission. To test PDR, we also use two different areas. In Figure 10(a), the area of 100m*150m is used. It can be seen that RL-SDWSN has a higher PDR than Q-learning, RLBR, traditional SDN, EASDN, QR-SND, and TIDE. In Q-learning, each node tries to learn the best path by itself. At the beginning of the communication, many packets are lost because the network needs time to establish a path from source to destination. During this period, each node broadcasts control packets to get neighbor information to estimate the best possible path. However, in this case, the network becomes congested, and a large number of packets are lost due to congestion.

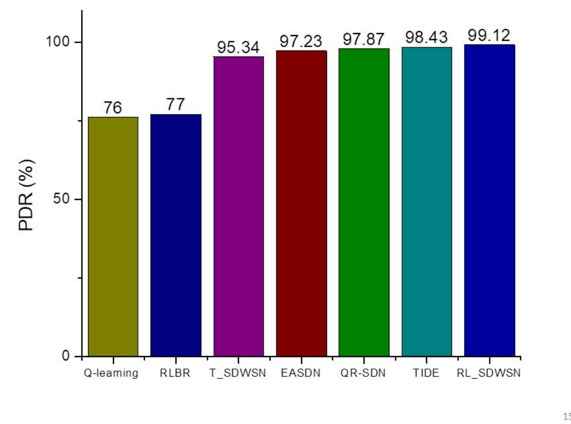
Similarly, in RLBR, the network is decentralized, and each node tries to learn the optimal path, but in order to learn, each node broadcasts control packets to obtain the state information of neighboring nodes. In the RL-based network, it is necessary to obtain the latest status data of neighboring nodes. The network became congested due to the large number of control packets being exchanged. As a result, a large number of packets are lost. However, in the SDN-based network, it has better PDR as compared to RL-based network, but the RL-based SDWSN performance is better than that of traditional SDWSN and EASDN. In the SDN-based network, the controller can view the underlying network globally and control the network. However, both QR-SND and TIDE are also SDN-based networks. In QR-SDN, the design of the reward function is not too much affecting the network data flow control that results in losing more packets during communication and giving less PDR as compared to the proposed RL-based SDWSN. In contrast, TIDE performance is approximately approaching RL-based SDN performance because it is taken into account

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

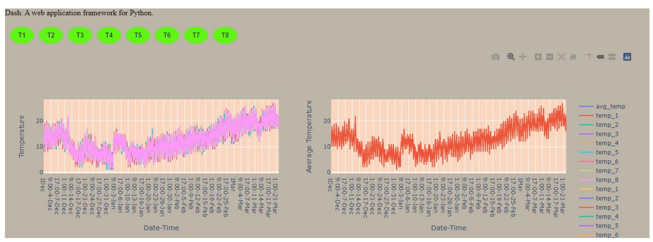


(a) Packet delivery ratio of 100m*150m area.

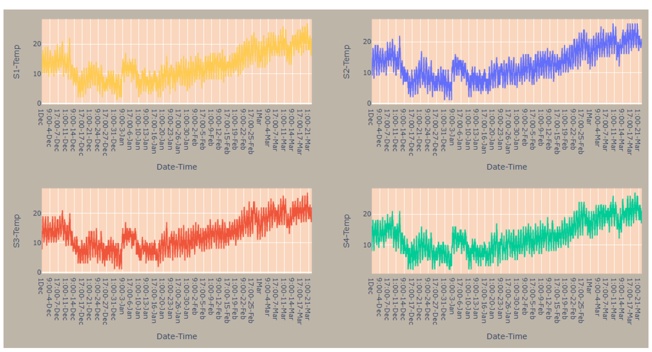


(b) Packet delivery ratio of 200m*300m area.

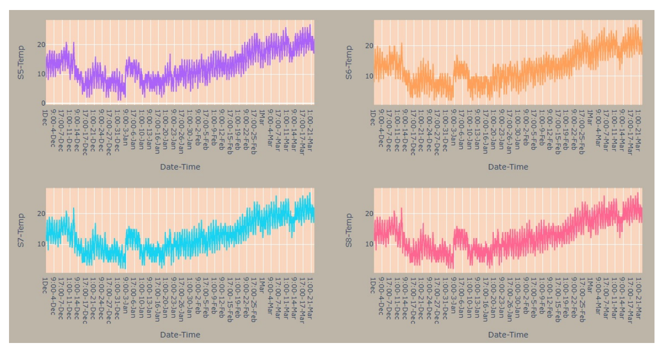
Fig. 10. Packet Delivery Ratio (PDR).



(a) Single dashboard for all sensor nodes.



(b) Sensor node 1,2,3 and 4 dashboard.



(c) Sensor node 5,6,7 and 8 dashboard.

Fig. 11. Web based sensor node monitoring dashboard 1.

in reward function that improves the network performance in terms of PDR.

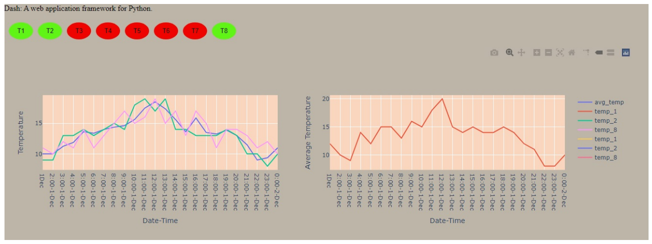
The SDN controller manages network traffic to avoid congestion; that is why the SDN-based network gives better PDR than the RL-based network. However, the SDN controller algorithm sometimes does not work well. The controller did not consider all the necessary factors to avoid congestion. In RL-based SDWSN, the agent selects a routing path after considering all factors (i.e., packet loss rate between two nodes, number of hops up to the destination, and distance up to sink). These factors affect network performance in terms of the packet success rate. In RL-based SDWSN, the agent has the ability to select the best routing path to achieve low congestion and high PRR. In the second scenario, the node deployment area is 200m*300m. In Figure 10(b), it can be seen that the RL-based SDWSN also has better PDR compared with all other techniques, and its behavior is also similar. However, the performance of RLBR is a little low compared with Q-learning. In terms of delivery ratio, all methods are giving low PDR in this scenario (200m*300m) than the first scenario (100m*150m) because the distance between two nodes increases, and the loss rate also increases with distance. From Figure 10(b), we can see that the RL-based SDWSN has little impact compared with other technologies. However, RL-based techniques are more effective in terms of PDR.

C. Web based sensor node monitoring System

IoT-based networks control the network devices (i.e., sensor nodes) remotely through the Web. We also developed a web-based dashboard for controlling the sensor nodes. In our

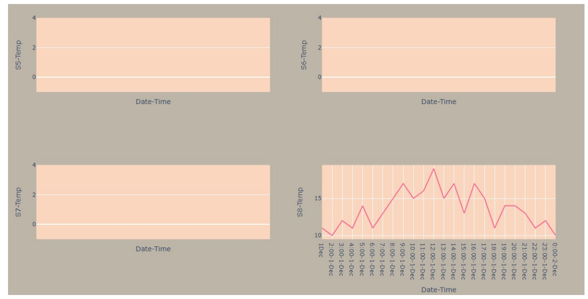
2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal



(a) Single dashboard for all sensor nodes.

(b) Sensor node 1,2,3 and 4 dashboard.



(c) Sensor node 5,6,7 and 8 dashboard.

Fig. 12. Web based sensor node monitoring dashboard 2.

network, eight sensor nodes have been deployed that measure the temperature. The dashboard is shown in Figure 11 & 12. In Figure 11, four months temperature is monitored through web. In Figure11(a), two different windows are shown. The first window shows the temperature of all deployed sensor nodes, while the second window shows the average temperature of all sensor nodes. However, in Figure 11(b), shows the individual temperature of sensor node 1 to 4 and 11(c), shows the temperature from sensor node 5 to 8.

In Figure 12, some sensor nodes are off (i.e., 3,4,5,& 6) that is shown in red colour; however, sensor node 1, 2, and 8 are on and the data (temperature) of these sensor nodes are shown in Figure 12(a). However, Figure 12(b) and 12(c) show the individual data of all sensor nodes. In Figure 12(b), we can see that only sensor nodes 1 and 2 are active and data is also showing on the dashboard, and similarly sensor node 5,6, and 7 are inactive, and the windows of these sensor nodes are empty. As we can see from both Figure 11 and Figure 12, network devices can be easily managed, observed and controlled remotely through the web.

VII. Conclusion

Wireless sensor networks (WSNs) is increasingly used in our daily life. This paper elaborates the need for an efficient IoT-based WSN framework because of its great importance in various application environments. It is an intimidating task to achieve the desired WSN performance through efficient routing. Therefore, we use RL and SDN combinations for efficient WSN routing, leading to improved routing decisions. SDN controller learns the routing path through RL and takes the next action according to the previously received reward. We used RL to select the best routing path from the routing list obtained by STP. The performance of RL-based SDWSN is compared with existing SDN-based techniques, including the traditional SDN and energy-efficient technique used for routing. The proposed RL-based SDWSN technique provides a better lifetime of 8-33% on average in all scenarios. We also compared the results of two different learning rates. Our proposed RL-based SDWSN method also increased the average network lifetime from 8 to 12% in all scenarios. Furthermore, we compare the performance of PDR that gives a higher delivery ratio as compared to SDN and RL-based techniques. Some other metrics, such as latency, delay, and throughput, also greatly impact network performance. In the future, we plan to optimize the parameters that are mentioned above by proposing some new algorithms and implementing the large-scale network. Also, we intend to compare the performance of the real-time network with the simulation work. However, we have also plan to detect the abnormal traffic of the network by using RL.

Acknowledgment

We would like to thanks Dr. S. Sabahat H. Bukhari for reviewing and giving valuable comments to improve the manuscript's quality.

References

[1] G. Wang, J.-K. Zhang, M. G. Amin, and K. M. Wong, "Nested cooperative encoding protocol for wireless networks with high energy efficiency," *IEEE transactions on wireless communications*, vol. 7, no. 2, pp. 521–531, 2008.

[2] A. De La Piedra, F. Benitez-Capistros, F. Dominguez, and A. Touhafi, "Wireless sensor networks for environmental research: A survey on limitations and challenges," in *Eurocon 2013*. IEEE, 2013, pp. 267–274.

[3] R. Jafari, A. Encarnacao, A. Zahoory, F. Dabiri, H. Noshadi, and M. Sarrafzadeh, "Wireless sensor networks for health monitoring," in *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*. IEEE, 2005, pp. 479–481.

[4] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless charging technologies: Fundamentals, standards, and network applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1413–1452, 2015.

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

[5] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[6] M. U. Younus, S. ul Islam, I. Ali, S. Khan, and M. K. Khan, "A survey on software defined networking enabled smart buildings: Architecture, challenges and use cases," *Journal of Network and Computer Applications*, vol. 137, pp. 62–77, 2019.

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[8] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.

[9] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for internet-of-things: A review," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453–463, 2015.

[10] T. Hu and Y. Fei, "Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796–809, 2010.

[11] H. Yao, T. Mai, X. Xu, P. Zhang, M. Li, and Y. Liu, "Networkai: An intelligent network architecture for self-learning control strategies in software defined networks," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4319–4327, 2018.

[12] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "Iot gateway: Bridging wireless sensor networks into internet of things," in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. Ieee, 2010, pp. 347–352.

[13] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless sensor networks and the internet of things: Do we need a complete integration?" in *1st International Workshop on the Security of the Internet of Things (SecIoT'10)*, 2010.

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

[15] M. U. Younus, S. W. Kim, *et al.*, "Proposition and real-time implementation of an energy-aware routing protocol for a software defined wireless sensor network," *Sensors*, vol. 19, no. 12, p. 2739, 2019.

[16] G. Li, M. Dong, K. Ota, J. Wu, J. Li, and T. Ye, "Deep packet inspection based application-aware traffic control for software defined networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–6.

[17] H. Huang, S. Guo, J. Wu, and J. Li, "Green datapath for tcam-based software-defined networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 194–201, 2016.

[18] 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, p. 4779, 2020.

[19] J. Wang, Y. Miao, P. Zhou, M. S. Hossain, and S. M. M. Rahman, "A software defined network routing in wireless multihop network," *Journal of Network and Computer Applications*, vol. 85, pp. 76–83, 2017.

[20] A. Banerjee and D. Hussain, "Sd-ear: Energy aware routing in software defined wireless sensor networks," *Applied Sciences*, vol. 8, no. 7, p. 1013, 2018.

[21] W. Xiang, N. Wang, and Y. Zhou, "An energy-efficient routing algorithm for software-defined wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7393–7400, 2016.

[22] S. Fu, J. Wu, H. Wen, Y. Cai, and B. Wu, "Software defined wireline-wireless cross-networks: Framework, challenges, and prospects," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 145–151, 2018.

[23] J. Wu, "Green wireless communications: from concept to reality [industry perspectives]," *IEEE Wireless Communications*, vol. 19, no. 4, pp. 4–5, 2012.

[24] W. Ejaz, M. Naeem, M. Basharat, A. Anpalagan, and S. Kandeepan, "Efficient wireless power transfer in software-defined wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7409–7420, 2016.

[25] M. Ndiaye, G. P. Hancke, and A. M. Abu-Mahfouz, "Software defined networking for improved wireless sensor network management: A survey," *Sensors*, vol. 17, no. 5, p. 1031, 2017.

[26] M. R. Celenlioglu, S. B. Goger, and H. A. Mantar, "An sdn-based energy-aware routing model for intra-domain networks," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2014, pp. 61–66.

[27] R. Wang, Z. Jiang, S. Gao, W. Yang, Y. Xia, and M. Zhu, "Energy-aware routing algorithms in software-defined networks," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. IEEE, 2014, pp. 1–6.

[28] Y. Duan, W. Li, X. Fu, Y. Luo, and L. Yang, "A methodology for reliability of wsn based on software defined network in adaptive industrial environment," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 74–82, 2017.

[29] G. Li, S. Guo, Y. Yang, and Y. Yang, "Traffic load minimization in software defined wireless sensor networks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1370–1378, 2018.

[30] J. Lorincz, A. Capone, and J. Wu, "Greener, energy-efficient and sustainable networks: State-of-the-art and new trends," 2019.

[31] C. Estevez and J. Wu, "Recent advances in green internet of things," in *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2015, pp. 1–5.

[32] N. Gligorić, S. Krčo, D. Drajić, S. Jokić, and B. Jakovljević, "M2m device management in lte networks," in *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*. IEEE, 2011, pp. 414–417.

[33] 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*, vol. 15, no. 2, p. 1550147719833541, 2019.

[34] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[35] R. Huang, X. Chu, J. Zhang, and Y. H. Hu, "Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 360428, 2015.

[36] Z. Zhang, L. Ma, K. K. Leung, L. Tassiulas, and J. Tucker, "Q-placement: Reinforcement-learning-based service placement in software-defined networks," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1527–1532.

[37] A. Al-Jawad, P. Shah, O. Gemikonakli, and R. Trestian, "Learnqos: a learning approach for optimizing qos over multimedia-based sdns," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2018, pp. 1–6.

[38] T. Uzakgider, C. Cetinkaya, and M. Sayit, "Learning-based approach for layered adaptive video streaming over sdn," *Computer Networks*, vol. 92, pp. 357–368, 2015.

[39] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "Qos-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 25–33.

[40] P. Sun, Y. Hu, J. Lan, L. Tian, and M. Chen, "Tide: Time-relevant deep reinforcement learning for routing optimization," *Future Generation Computer Systems*, vol. 99, pp. 401–409, 2019.

[41] 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. 174 773–174 791, 2020.

[42] C. Fang, C. Cheng, Z. Tang, and C. Li, "Research on routing algorithm based on reinforcement learning in sdn," in *Journal of Physics: Conference Series*, vol. 1284, no. 1. IOP Publishing, 2019, p. 012053.

[43] T.-Y. Mu, A. Al-Fuqaha, K. Shuaib, F. M. Sallabi, and J. Qadir, "Sdn flow entry management using reinforcement learning," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 13, no. 2, pp. 1–23, 2018.

[44] H. Mostafaei and M. S. Obaidat, "Learning automaton-based self-protection algorithm for wireless sensor networks," *IET Networks*, vol. 7, no. 5, pp. 353–361, 2018.

[45] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled internet of things: Network architecture and spectrum access," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 44–51, 2020.

[46] G. Li, K. Ota, M. Dong, J. Wu, and J. Li, "Desvig: Decentralized swift vigilance against adversarial attacks in industrial artificial intelligence systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3267–3277, 2019.

[47] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*. IEEE, 2000, pp. 10–pp.

[48] P. Winter, "An algorithm for the enumeration of spanning trees," *BIT Numerical Mathematics*, vol. 26, no. 1, pp. 44–62, 1986.

[49] M. Chakraborty, S. Chowdhury, J. Chakraborty, R. Mehera, and R. K. Pal, "Algorithms for generating all possible spanning trees of a simple undirected connected graph: an extensive review," *Complex & Intelligent Systems*, vol. 5, no. 3, pp. 265–281, 2019.

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3102130, IEEE Internet of Things Journal

Muhammad Usman Younus received his Doctorate and master's degree in engineering from the University of Toulouse (III) Paul Sabatier France and University of Engineering and Technology Lahore Pakistan 2020 2014, respectively. His research interests include Wireless Sensor Network, Software Defined Networking, Energy Optimization, Wireless Communication, and Machine Learning. He has more than twenty journal and conference publications. He is a member of PEC, IEEE, etc. and reviewer of more than ten journals and conferences.

Muhammad Khurram Khan (M'07-SM'12) is currently working as a full professor at the Center of Excellence in Information Assurance, King Saud University, Saudi Arabia. He has edited seven books and proceedings published by Springer-Verlag and IEEE. He has published more than 370 papers in international journals and conferences and he is an inventor of several U.S./PCT patents. Dr. Khan is the Editor-in-Chief of a well-reputed journal 'Telecommunication Systems' (Springer). He is a full-time Editor/Associate Editor of several international journals/magazines, including IEEE Communications Surveys Tutorials, IEEE Communications Magazine, IEEE Internet of Things Journal, IEEE Transactions on Consumer Electronics, Journal of Network Computer Applications (Elsevier), IEEE Access, Security Communication Networks, IEEE Consumer Electronics Magazine. His current research interests include Cybersecurity, biometrics, multimedia security, and digital authentication. He is a Fellow of the IET (UK), Fellow of the BCS (UK), Fellow of the FTRA (Korea), senior member of the IEEE (USA), a member of the IEEE Technical Committee on Security Privacy, and a member of the IEEE Cybersecurity community

Abdul Rauf Bhatti received the B.Sc. and M.Sc. degrees in electrical engineering from University of Engineering and Technology (UET), Lahore, Pakistan, in 2004 and 2008, respectively, and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia in 2017. He is currently working with Government College University Faisalabad (GCUF), Faisalabad, Pakistan as an Associate Professor and Chairman of Electrical Engineering and Technology Department. He is also performing the duties of Director ORIC (Office of Research, Innovation and Commercialization) on attachment basis in GCUF. He started BS, MS and Ph.D. electrical engineering programs and established various engineering labs in GCUF. He is serving the GCUF since 2006 at various posts like Lecturer and Assistant Professor. Before that he was with Government College of Technology (GCT), Faisalabad, Pakistan and Shakarganj Mills Ltd. (SML), Jhang, Pakistan in 2006 and 2005 respectively. He also served UET Faisalabad Campus as visiting Lecturer from 2006 to 2007. Dr. Bhatti is a life time Professional Engineer with Pakistan Engineering Council and member IEEEP. He received Best Teacher and Outstanding Researcher Awards in GCUF, Most downloaded Paper in Wiley for 2018-2019 and Best Paper Award in IEEE 6th International Conference on Power and Energy, Malaysia. He is the reviewer of various national and international research journals in the field of electrical engineering. He has 35+ research articles on his credit. His research interests are in electrical energy management, electric vehicle charging, solar photovoltaic, energy forecasting, artificial intelligence, deterministic algorithms and renewable energy.

2327-4662 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Glasgow. Downloaded on August 20,2021 at 14:05:01 UTC from IEEE Xplore. Restrictions apply.