International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhancing Routing Performance in Software-Defined Wireless Sensor Networks through Reinforcement Learning

Nilesh P. Sable[1], Vijay U. Rathod[2], Mangesh D. Salunke[3], Hemantkumar B Jadhav[4], Ravindra S. Tambe[5], Suhas R. Kothavle[6]

Submitted: 21/05/2023 **Revised**: 07/07/2023 **Accepted**: 26/07/2023

Abstract: Software-Defined Networking (SDN) has swiftly taken over networks in data centers, telecommunications companies, and organizations because to its programmable and adaptable control plane. Due to its adaptability, SDN is a new architecture that is employed in numerous applications. The necessity for routing optimization has increased as a consequence of the exponential growth in network traffic demands needing quality of services. In order to enable the Internet of Things (IoTs), it is considered to be vital. Modern developments in SDN technology has allowed for central control and management, and programmatic interfaces enable flexible customization of network service like switches. SDN for routing has been introduced in WSNs. The SDN controller uses a variety of different methods to establish the routing path, but none of them are sufficiently efficient to provide the ideal routing path. As a result, reinforcement learning (RL) is a practical method for figuring out the best routing path. In this study, we improve the SDWSN's RL-based routing path. It is recommended to use a reward system that contains the relevant network QoS and energy efficiency metrics. While the agent receives the award and chooses what to do next base on the reward received, the SDWSN controller improves the routing path based on prior information. However, the Web also allows for remote management of the entire network.

Keywords**:** *WSNs, SDWSN, routing, RL-based WSN, RL, IoTs, Energy optimization.*

1. Introduction

SDN has developed as one of the most intriguing networking recently developed technologies. The paradigm emphasized in emphasizes the separation of the control plane from the data plane and runs on high-performance commodity hardware with a logically centralized control plane. It has been widely embraced in actual wireless networking contexts, including datacentres, organizations that provide network infrastructure, and business networks. Additionally, since SDN has gained popularity, its security has drawn greater scrutiny. As a result, it is simple to locate numerous SDN security-related works that aim to protect SDN elements namely the switching, controllers, and SDN applications as shown in Figure 1.

[1]Department of Information Technology, Bansilal Ramnath Agarwal Charitable Trusts, Vishwakarma Institute of Information Technology, SPPU, Pune, India, drsablenilesh@gmail.com
[2]Department of Artificial Intelligence & Data Science, GH Raisoni College of Engineering & Management Wagholi, Pune, SPPU, Pune, India, vijay.rathod25bel@gmail.com
[3]Department of Computer Engineering, Marathwada Mitramandal's Institute of Technology, SPPU, Pune, India, salunkemangesh019@gmail.com
[4]Department Department of Computer Engineering, Adsul's technical campus, Ahmednagar, India, hem3577@gmail.com
[5]Department Department of Computer Engineering, Dr. Vithalrao Vikhe Patil College of Engineering, Ahmednagar, India, ravindra.tambe01@gmail.com
[6]Department of Computer Engineering, Marathwada Mitramandal's Institute of Technology, SPPU, Pune, India, suhas.kothavle15@gmail.com

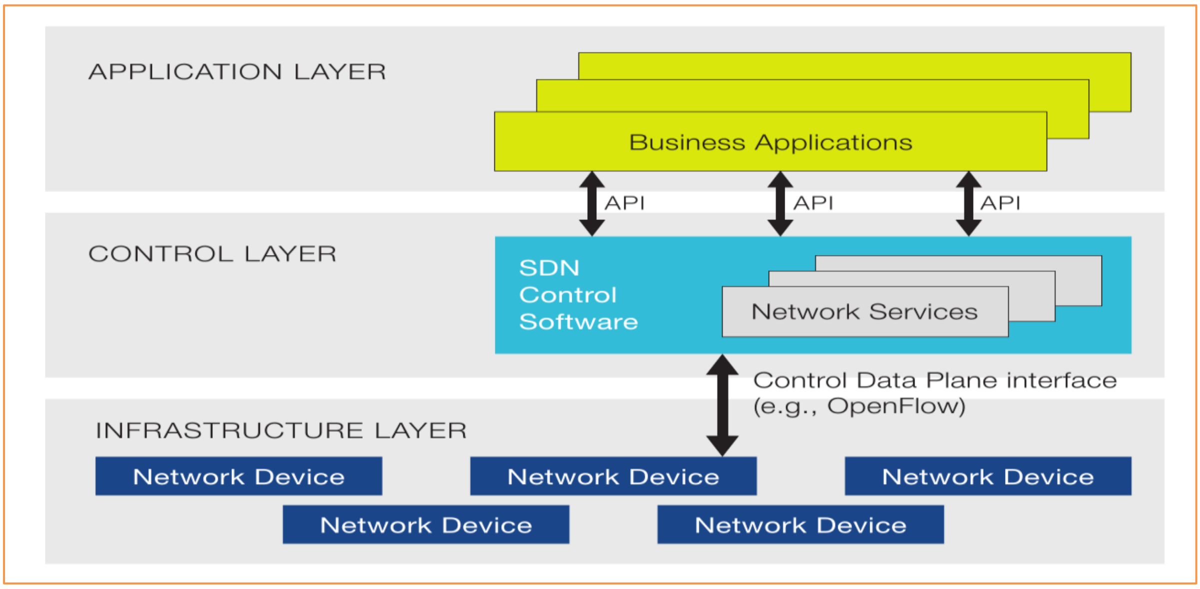

Fig 1. Overview of the SDN Architecture [1].

For instance, FortNox has already looked at the plausibility of attack scenarios particular to SDN. Many scholars have offered attack scenarios and alternative defenses [1].

Small sensor nodes, which might be mobile or permanent nodes deployed in a dynamic environment, are present in WSNs. One or more processing units, communication units, and tiny power sources make up each sensor node. The generalized WSN communication architecture is shown in Figure 2. A communications network with an emphasis on applications, the WSN is employed in numerous applications. Such as security, army, healthcare, and environmental sensing (including lighting, temperatures, moisture, and vibrations) [2].

Wireless network resources (i.e. bandwidths, CPU Time, etc.) optimization in WSN is one of the complicated issues. The performance of the network is enhanced. A recent networking innovation known as SDN can help with this. The idea behind SDN is to divide a network's control plane and forwarding plane in order to boost the performance of communication networks. Network programmed that control network traffic and seek to make the best use of available resources are referred to as traffic engineering (TE). Numerous capabilities of SDN are available to enable TE applications. SDWSN is a new networking domain created by combining SDN and WSN. Resource optimization is required in WSN due to the resource-constrained nodes it has [3].

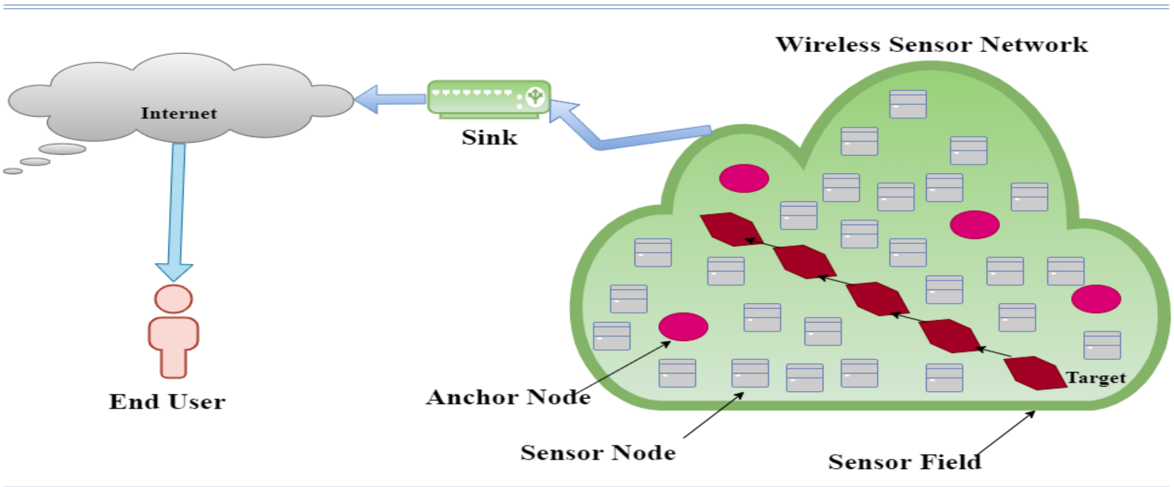

Fig 2. Communication Architecture for WSN

Recent work in the WSN field has largely focused on developing minimal-cost and limited-power networking systems to carry out collaborative and cooperative tasks under strict computational and energy limitations. As a result, the wireless sensor nodes can run for a long time in many applications without needing to be recharged. In the wireless network architecture, the energy consumption of the sensor nodes is therefore seen as being of utmost importance. Routing, which transfers sensed data from the source sensor node to the destination or sink sensor node, is the primary networking activity in WSNs. It has a substantial impact on the network's performance in terms of things like energy use, network latency, and packet delivery rate (PDR). Routing techniques like SDN and RL might be a great option to acquire an optimized routing path in WSNs and increase the efficiency of WSNs [2].

Network administrators have continued to use TE strategies to improve network resource management performance in order to handle the steady rise in network traffic volume. To enhance the overall QoS for network users, TE in communication networks integrates the packet forwarding pathways of various flows. In order to enhance network utility, one of the major issues in TE has been continually studying routing optimization [4].

In conventional routing techniques, each router independently decides how to forward packets without taking into account the choices of other routers. The fact that this distributed routing technique can be used regardless of the size of the network gives it scalability. It is challenging to approach network-wide routing optimization and manage resources effectively and flexibly. A better network management paradigm is more necessary as a result of these factors. It has been proposed that SDN, which separates the control and data planes of the network, is an effective method of managing the entire network. The SDN also distinguishes between control and data transport activities. SDN enhances network programmability for network operation and offers a comprehensive picture of the entire network. The SDN manages the network by logically dividing its control plane from its data plane. This SDN paradigm allows for effective network monitoring and flexible network policy deployment [4].

Although SDN makes it possible to centralize packet forwarding management with a network-wide perspective, designing the best routing strategy is not an easy task. A limited shortest path issue is how the routing challenge is formulated in many previous studies, although an optimal solution to these challenges is typically NP-hard. Additionally, even though a popular approach to the generic multi-commodity flow problem involves analyzing network operation as a fixed model with variable traffic, such models are unable to accurately depict good network operation under complicated and dynamic traffic [4].

An innovative architecture called SDN effectively governs the network. It has three kinds of planes: I. Data plane, II. Application plane and III. Control plane. It divides the control plane from the programmable data plane. Its purpose is to use and optimize network resources effectively. In SDN, a centralized controller manages the network with help from the data plane's packet forwarding devices and the control plane, which can keep an eye on the entire underlying network from a distance. The control plane is in charge of fault recovery, traffic management, and routing. The management of packet delivery is under the purview of the data plane. SDN makes use of the well-defined interfaces between each plane, such as the southbound interface between the control plane and the data plane. The interface between the control plane and the application plane lies in the north. Additionally, in order to interface with data plane devices like sensor, switch, and router, it makes use of the open flow communication(OFC) protocol utilized by the SDN controller [2].

Due to SDN's effective structure, which can effectively monitor and regulate the billions of network devices, its use in the IoT is growing day by day. Edge network, access connectivity, core connectivity, and datacentres connectivity are just a few of the networks that use software-defined IoT. IoT also includes numerous wireless devices controlled by SDN. In terms of security and scalability, the IoT network is still dealing with several issues and challenges [2].

WSN is made more resilient and organized by the usage of SDN, also known as an SDWSN. Finding the optimal routing path, for example, is one of SDWSN's remaining drawbacks. It can be resolved using the learning method known as RL. Lowering energy consumption and latency while boosting PDR is possible with real-time path optimization utilizing RL. In earlier work, we employed RL to optimize SDWSN's energy usage. Estimate Node Lifetime (ENLT) and Path Estimated Lifetime (PELT) were used to determine the reward. However, the QoS factors have not been taken into account, which could lower network performance. Using a different model, the energy consumption of SDWSN was determined. However, rather than comparing the most recent work, the suggested work was merely compared to basic procedures. The primary goal of was to reduce SDWSN's energy use. Figure 3 provides a comparison of our most recent and earlier findings.

Parameters	Previous Research Work	Current Research Work
QoS Parameters used in Reward Function	No	Yes
QoS based Algorithms for both Controller & Node Side	No	Yes
PDR Comparison	No	Yes
Result Comparison with Previous RL-SDWSN Techniques	No	Yes
IoT based Control Network	No	Yes

Fig 3. Comparisons between the Present and Past Work.

The network can be managed more effectively and performs better when SDWSN and RL are combined. Figure 4 depicts the RL-based SDWSN architecture. Some artificial intelligence and deep RL techniques are utilized in SDN to enable the network to effectively learn for itself and operate the network. The IoTs, on the other hand, can remotely control the WSN network. Any gadget can now connect with another device through the Internet thanks to a new paradigm. The IoT architecture comprises of nodes, which are low-processing-power devices. The regional controller is responsible for gathering node data. The general control mechanism of the system is provided by the cloud layer that remotely maintains and monitors the nodes. Figure 5 displays the WSN background for an IoT application. The SDN architecture and RL are used for local control and optimization of the routing path.

However, as indicated in the suggested portion, it is also possible to operate the sensor node globally over the internet. The contribution of this paper is enumerated in the following.

A. To maximize the routing that picks the optimal path from the routing list, it is advised to utilize a QoS-based reward function that takes into consideration a number of QoS criteria. The longevity and PDR of the network are both increased while the energy consumption is decreased.

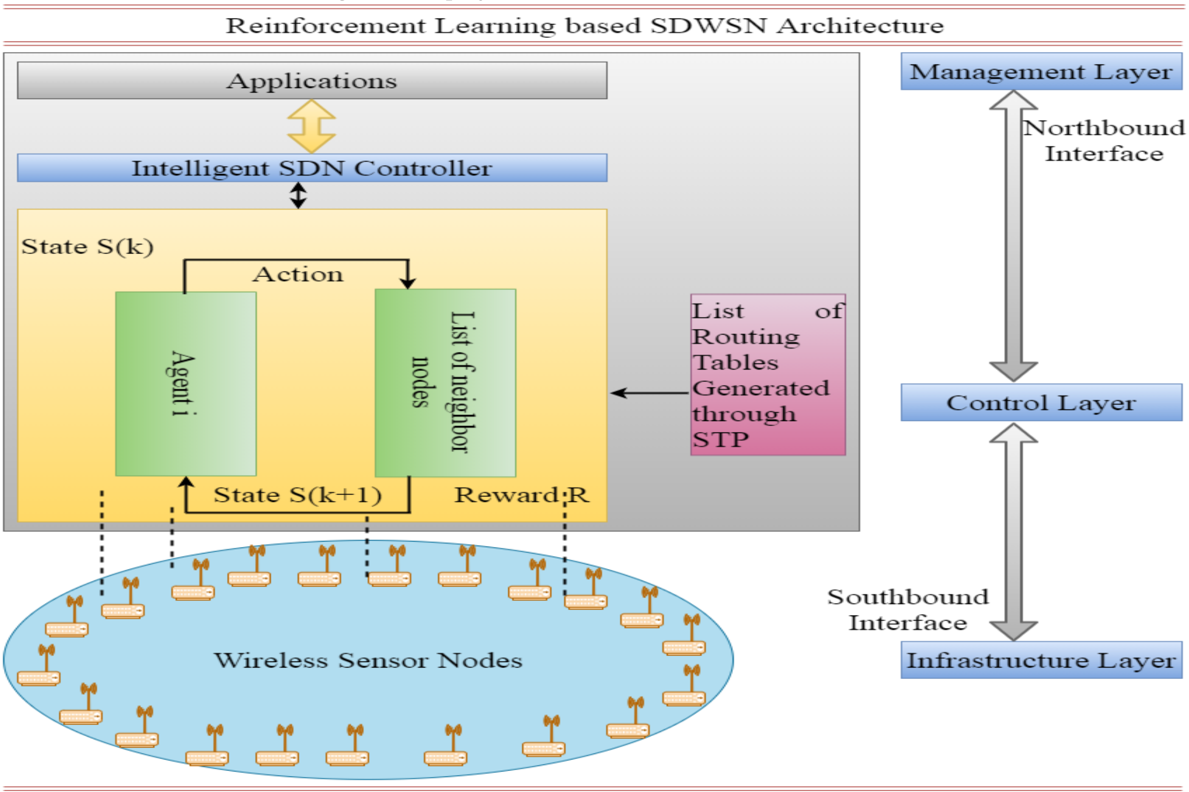

Fig 4: SDWSN architecture built on RL.

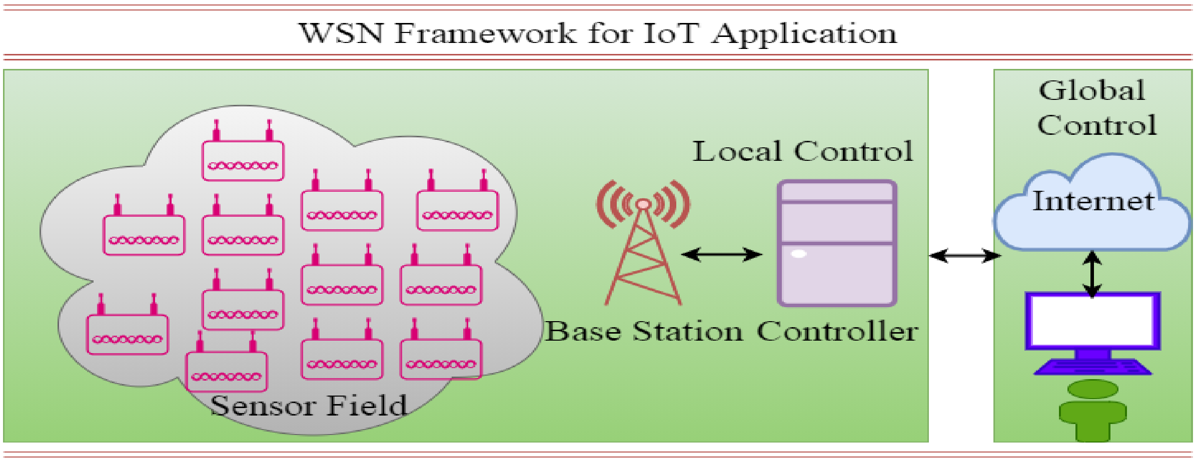

Fig 5. IoT application using the WSN framework.

B. We propose two algorithms to enhance network efficiency in terms of PDR, energy efficiency, and other aspects. To effectively manage the data plane devices, these algorithms are used to a clever SDN controller.

C. The spanning tree protocol (STP) is used to achieve loop free communication.

D. The data from the sensor nodes can also be remotely managed and analyzed using a web based interface.

2. Related Work

WSN performance, including energy economy and QoS factors, can be improved through routing. We use the SDN architecture for optimum routing and provide a several algorithms to improvement the efficiency of SDN-based networks. While the SDN-based network occasionally performs poorly in real-time. RL can play a crucial part in improving network performance to address this problem. The routing methods are divided into two groups in this section: 1) Routing techniques based on SDN and 2) SDN routing strategies based on RL.

A. Routing Techniques not based on RL

An improved link utilization-based technique for TE in SDWSN is described in this paper. In the suggested approach, the Optimized Link Allocation based TE (OLA-TE) algorithm for streamlining traffic is covered. OLA-TE is implemented by the controller node, improving overall network utilization. With the assistance of the controller node, the SDN sensor nodes identify the optimal traffic routing path [3].

The WSNs are made up of discrete nodes that communicate with the outside world by monitoring and regulating physical variables including moisture, pressure, and temperatures. Given the energy limitations of WSN nodes, significant work has been put into designing an efficient routing protocol to increase the lifespan of WSNs. In order to increase the network's lifespan, clustering is a useful technique. SDN has recently been recognized as an effective network model for WSNs. Authors provide an overview of WSNs and SDN in this work. Authors talk about some recent works that combine SDN with WSN. Next, researcher suggests the two energy-efficient routing strategies to lengthen the lifetime of WSNs. In this initial contribution, they combine a traditional methodology with the clustering method. In the second contribution, they suggest a novel strategy employing the SDN paradigm. In this instance, they are energy-doping a portion of the advanced node to perform the role of SDN controller [5].

An SDN-based management system for WSNs using IPv6 over low-power wireless LANs is built and evaluated by researchers in this study. The entire framework is explained, including several data, control, and application plane implementations. The framework uses a brand-new packet format and addressing scheme. A centralized routing protocol built on the shortest path algorithm is also housed in the SDN controller. The authors compare their approach to the distributed routing protocol employed by the low-power and lossy networks [6].

The granularity of monitoring by an SDN controller and the communication overhead of flow management are traded off in this study using a cluster-based approach to flow management. In a network, clusters form when there are few or no border nodes. The SDN controller merely handles incoming flows as opposed to managing each node's separate flows. Only the outbound traffic from clusters is controlled by the SDN controller using border nodes [7].

In this work, authors survey this area based on the clustering goals, such as load balancing and energy consumption reduction. The mobility and heterogeneity of the network, which are essential for efficient clustering in the IoT. They also look at the benefits and drawbacks of clustering when IoT is combined with cutting-edge computing and communication technologies like 5G, Blockchain, and fog or edge computing. This study adds important light on the topic of IoT clustering research, giving a better comprehension of its network design issues and illuminating its potential future applications in cutting-edge integrating IoT technology [8].

B. Routing Approaches based on RL for SDWSNs

Researchers describe a strategy for optimizing SDN routing based on deep reinforcement learning (DRL) in this research. Using the described technique, the DRL agent gains knowledge of the connection between network efficiency and traffic load on network switches. The proposed method chooses an optimal set of link weights to strike a compromise between the end-to-end delay and packet losses of the network. The SDN controller installs the flow-rules on the SDN-capable switches and chooses the routing paths based on the link weights. In an environment of topological change, to prevent a highly drawn-out DRL learning process. Researchers construct an M or M or 1 or K queue-based network model and employ the network model for the offline learning of DRL until it converges [4].

Unsupervised learning techniques, such as RL, are widely employed in real-time applications. A decision-making dilemma is the basis of RL. In RL, the mediator continuously engages with the atmosphere and decides what to do next based on feedback from prior actions regarding rewards. In this study, RL teaches the SDWSN controller to enhance the routing routes. The routing tables on the SDN controller are built using RL by researchers who combine RL and SDN [9].

The goal of this paper is to summarize the status of SDWSN proposal developments. Acquaint readers with the new area of research known as ML-SDWSNs, which combines the principles of machine learning (ML) with SDWSNs. To increase network performance, ML-SDWSN provides a centralized, resource-aware, intelligent architecture. The difficulties currently encountered in the practical deployment of SDWSNs can be resolved using ML-SDWSN techniques. The present state-of-the-art ML approaches are a major focus of this survey, which also raises questions for the technical and manufacturing societies, as well as specialized organizations concerned in SDWSN [10].

To build a scalable SDN control plane, many controllers are interconnected. Every switch is linked to a certain data plane controller. As the services offered by switches change, so does the pressure on controllers. To divide the load among SDN controllers, switches must be converted from master to slave controllers. This paper presents a dynamic switch migration strategy combining Q-learning to produce a scalable SDN control plane. The algorithm simulates the switch migration problem, modifies the Q-learning parameters in accordance with the switch migration model, and derives switch immigration decisions animatedly when the load on the controllers changes in SDN [11,13,14].

3. Q-Learning

In Q-learning, a learner is an agent that decides on an action based on the current state by interacting with the environment, receiving feedback that is either positive or negative based on the action known as reward, and computing the Q-value. The agent selects an action in the succeeding state S_{t+1}, based on the preceding reward. The agent receives training, and after a few rounds, converges to the ideal position. According to Figure 7, the Q-value in Q-learning is changed after each repetition.

It originates from the Q-table, which routes the data packets using the Q-learning technique. The initial Q-matrix of node *i* is formed in Q-routing. The Q-initialization matrixes may be arbitrary. Next, neighbor node *j* receives packet *p* from node *i*. The forwarder node *j* with the lowest Q-value is chosen by node *i* due to the close proximity to the destination *d*. Low distance results in minimal routing costs and delivery delays. Algorithm one gives specific the Q-routing algorithm, as demonstrated in Figure 8.

The delivery delay is determined by the formula $Q_i(d, j)$ from equation (2). The t_j is the projected amount of time left in the journey till node *i* returns to node *j* without delay, and it may be determined by

$$t_i = Min_{K \epsilon N_g} Q_j(d,k) \quad (1)$$

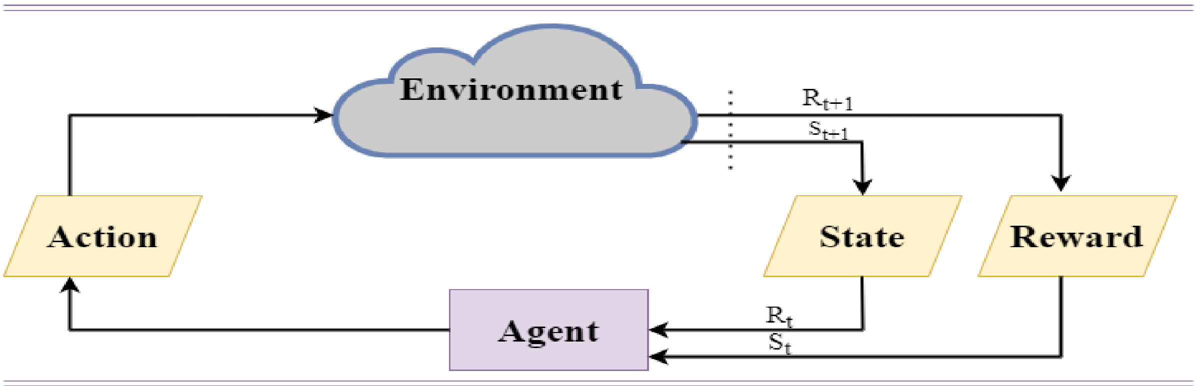

Fig 6. The RL Model's basic operating principle

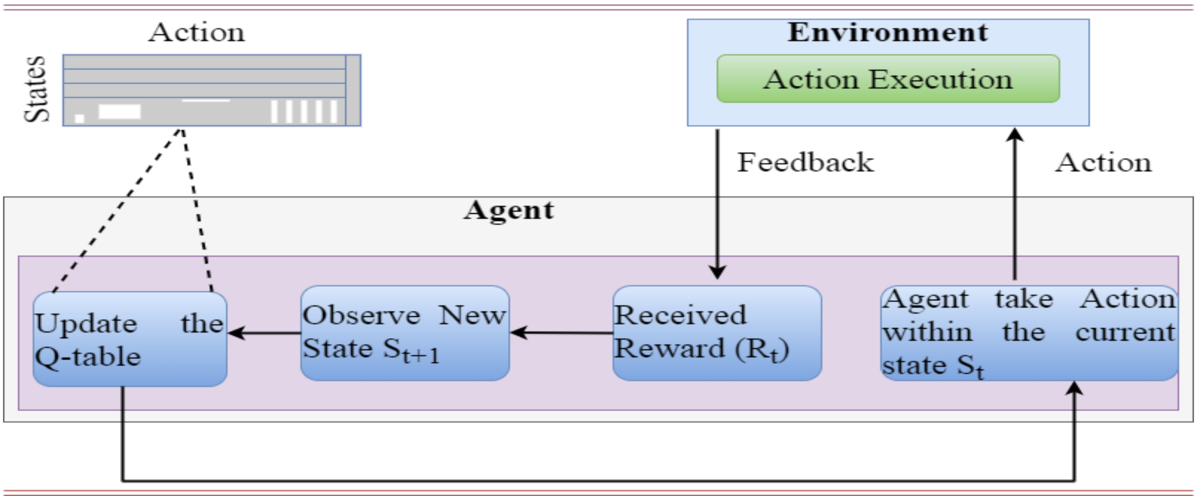

Fig 7. Block diagram for Q-learning

Using a distance, we can calculate $Q_i\ (d,\ j)$, which represents the expected time it will take node *i* to transmit the packet *p* to node *j*.

$$Q_i\,(d,j) = (1\text{-}\alpha) * Q_i\,(d,j) + \alpha * (q_i + T_x + t_j) \tag{2}$$

Here each symbols represents, T_x stands for the amount of time needed for communication among nodes *i* and *j*, whereas q_i stands for the amount of time to spend packet *p* in the queue of node *i*. Here α stands for the learning period.

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 (1)
        and (2).
        Update the node i delivery delay time.
    End
End
```

Fig 8. Algorithm 1: Pseudo Code for Q-Routing Algorithm

4. Proposed Method

According to the network architecture and transmission range, every sensor node in the WSN gathers information from the environment and transmits it to the sink through only one or multiple hops. In a conventional WSN, every node transmits a data packet to acquire data about its adjacent nodes [12]. Due to broadcast, the network consumes a significant amount of energy, and generalized algorithms are unable to optimize the path [13]. The control plane and data plane are separated by a new architecture known as SDN. The network is managed by a centralized controller with access to a global view of the system. However, the SDN controller algorithms do a poor job of handling real-time network path optimization. Real-time routing can be optimized with the use of RL, an effective learning method. In this situation, combining SDN with RL is the optimum strategy for optimizing the WSN routing path. In this study, the SDN controller's RL is used to select the routing list's ideal path and, if necessary, to alter the path.

In our energy optimization strategy, we suggest a reward function that takes into account all necessary network performance-related metrics. The distance to sink, the quantity of hops needed to sink, the amount of energy still in the system, and the packet success rate make up a reward function. This algorithm is split into two pieces to optimize the WSNs routing path: I. Intelligent SDN Controller and II. Sensor Nodes. The first phase of Figure 9 shows how the controller side locates adjacent nodes after the controller node startup, which compiles the status data for the entire network. The controller uses STP to construct every conceivable routing path. Figure 9's second phase illustrates how it utilizes Q-learning to pick one routing table from the list and send it to nearby nodes. The controller gathers the sensor nodes' status information after each epoch and determines the reward. The third phase of Figure 9 shows how the intelligent SDN controller modifies the routing path based on system feedback regarding rewards. The performance of the network will be decreased and the path will be altered if the reward is negative; if it is positive, the path is kept. The controller also regularly checks the amount of energy left in each node. A routing table is selected from the list and delivered to the neighboring node after the routing path list is recalculated using STP and any nodes with energy below the threshold are deleted from the list of nodes. Algorithm 1 displays the SDWSN controller algorithm. As demonstrated in Algorithm 3, the initial neighbor discovery phase begins on the node side, where each node broadcasts the Hello packet to detect the neighbors. This process keeps going until a certain period has passed and the number of acceptable neighbors has been attained. Each node informs the nodes nearby of its status after the neighbor discovery time. The status information from each node reaches the designated recipient or controller via multihop communication. The SDN controller provides a routing table to each node, and each node transmits a data packet in accordance with the routing table that the knowledgeable SDN controller has provided (i.e., Algorithm 2). Each round's energy usage is determined by the sensor node using a mathematical model and it notifies the controller in a data packet how much energy is still available. It initially assesses the remaining energy of the intermediary node and, if it exceeds the threshold, decides whether to accept. If not, it sends a low energy signal message to the controller and disconnects from the network.

Considering the computational complexity of the node side algorithms and the energy-efficient controller. Until the last node dies, the controller side keeps running n times. The controller side operation as a whole has a complexity $O(n)$. The first operation, however, is still applicable on the node side up to two threshold approaches, t_{nbr} and NBR_{max}. While the two internal operations that receive the status information from nearby nodes require n iterations to complete, this operation executes n times. Other actions determine a node's energy usage up to the "*Required Energy for Tx load*." The total number of iterations in the operation is $n(n + n)$. Therefore, $O(n2)$ is the entire node side complexity. The great majority of previous solutions depended on heuristics that were slower than the suggested techniques since global optimization is an NP-hard issue.

Fig 9. Set up the route in accordance with the flow table that you received from the controller example.

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 ⟵ $\{x_1, x_2, x_3, ..., x_n\}$.
SDNController ⟵ 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 1.
Calculate the Q-Value using equation 2.
Update the Q-value.
while (*Received node data upto last node die*) **do**
    if ($E^{residual}$ $_{node}$ < TH) **then**
        Exclude that node from the list.
        Recalculate the routing tables using STP.
        RT ⟵ $\{x_1, x_2, x_3, ..., x_m\}$.
        Select one routing path from the list and send it to the neighboring node.
    end
    else
        RT ⟵ $\{x_1, x_2, x_3, ..., x_n\}$.
        SDN Controller ⟵ RT.
        Estimate the PELT.
        Calculate the Reward by using equation 1.
        Calculate the Q−Value using quation 2.
        Select one routing table with highest Q−value.
        Update the Q−value.
    end
end

Fig 10. Algorithm 2: SDN Controller Algorithm

Algorithm 3: Algorithm for Sensor Nodes

Input: Routing paths that receive from controller.

Output: 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 = $\{S_1, S_2, . . . , S_n\}$.

Assign the IP to controller.

if (RE $>E_{Threshold}$) **then**
    S ⟵ 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 does **not** exist in "Nbr list"*) **then**
            Add into the neighboring list.
        end
    end
    Calculate the energy consumption of nodes after each T_x and R_x 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 does not exist in the "list"*) **then**
            Add the sender node address into list.
            ++ Status data response number.
        end
    end
    while (*RE <Required Energy for* T_x *load*) **do**
        T_x & R_x the both control and data traffic .
        Calculate the energy consumption of nodes after each T_x and R_x control and data traffic.
    end
    Send the notification of low energy to controller through neighboring node and disconnect.
end

Fig 11. Algorithm 3: Sensor Node Algorithm

5. Conclusion

WSNs are being used more frequently in our daily lives. Due to its critical importance across a range of application scenarios, this paper elucidates on the requirement for an effective IoT-based WSN infrastructure. Using effective routing to attain the necessary WSN performance is a daunting challenge. As a result, we combine RL with SDN for effective WSN routing, which results in better routing choices. The SDN controller applies RL to understand the routing path and bases its decision on the previous reward. From the routing list generated by STP, we used RL to choose the optimal routing path. Comparisons are made between the efficacy of RL-based SDWSN and other SDN-based strategies in use at the moment, such as outmoded SDN and energy efficient routing techniques.

Reference

[1] S. Lee *et al.*, "A Framework for Policy Inconsistency Detection in Software-Defined Networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 3, pp. 1410–1423, 2022, doi: 10.1109/TNET.2022.3140824.

[2] M. U. Younus, M. K. Khan, and A. R. Bhatti, "Improving the Software-Defined Wireless Sensor Networks Routing Performance Using Reinforcement Learning," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3495–3508, 2022, doi: 10.1109/JIOT.2021.3102130.

[3] R. Kumar, U. Venkanna, and V. Tiwari, "Optimized Link Utilization Based Approach for Traffic Engineering in SDWSN," *Int. Symp. Adv. Networks Telecommun. Syst. ANTS*, vol. 2019-Decem, pp. 1–6, 2019, doi: 10.1109/ANTS47819.2019.9118068.

[4] G. Kim, Y. Kim, and H. Lim, "Deep Reinforcement Learning-Based Routing on Software-Defined Networks," *IEEE Access*, vol. 10, pp. 18121–18133, 2022, doi: 10.1109/ACCESS.2022.3151081.

[5] M. S. Azizi and M. L. Hasnaoui, "Software defined networking for energy efficient wireless sensor network," *Proc. - 2019 Int. Conf. Adv. Commun. Technol. Networking, CommNet 2019*, pp. 1–7, 2019, doi: 10.1109/COMMNET.2019.8742362.

[6] F. F. Jurado-Lasso, K. Clarke, and A. Nirmalathas, "A Software-Defined Management System for IP-Enabled WSNs," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2335–2346, 2020, doi: 10.1109/JSYST.2019.2946781.

[7] Q. Liu, T. Ozcelebi, L. Cheng, F. Kuipers, and J. Lukkien, "CluFlow: Cluster-based Flow Management in Software-Defined Wireless Sensor Networks," *IEEE Wirel. Commun. Netw. Conf. WCNC*, vol. 2019-April, pp. 1–8, 2019, doi: 10.1109/WCNC.2019.8885485.

[8] A. Shahraki, A. Taherkordi, O. Haugen, and F. Eliassen, "A Survey and Future Directions on Clustering: From WSNs to IoT and Modern Networking Paradigms," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 2242–2274, 2021, doi: 10.1109/TNSM.2020.3035315.

[9] M. U. Younus, M. K. Khan, M. R. Anjum, S. Afridi, Z. A. Arain, and A. A. Jamali, "Optimizing the Lifetime of Software Defined Wireless Sensor Network via Reinforcement Learning," *IEEE Access*, vol. 9, pp. 259–272, 2021, doi: 10.1109/ACCESS.2020.3046693.

[10] F. F. Jurado-Lasso, L. Marchegiani, J. F. Jurado, A. M. Abu-Mahfouz, and X. Fafoutis, "A Survey on Machine Learning Software-Defined Wireless Sensor Networks (ML-SDWSNs): Current Status and Major Challenges," *IEEE Access*, vol. 10, pp. 23560–23592, 2022, doi: 10.1109/ACCESS.2022.3153521.

[11] Z. Min, Q. Hua, and Z. Jihong, "Dynamic switch migration algorithm with Q-learning towards scalable SDN control plane," *2017 9th Int. Conf. Wirel. Commun. Signal Process. WCSP 2017 - Proc.*, vol. 2017-Janua, pp. 1–4, 2017, doi: 10.1109/WCSP.2017.8171121.

[12] N. P. Sable, V. U. Rathod, P. N. Mahalle, and D. R. Birari, "A Multiple Stage Deep Learning Model for NID in MANETs," *2022 Int. Conf. Emerg. Smart Comput. Informatics, ESCI 2022*, pp. 1–6, 2022, doi: 10.1109/ESCI53509.2022.9758191.

[13] V. U. Rathod and S. V Gumaste, "Effect Of Deep Channel To Improve Performance On Mobile Ad - Hoc," vol. 41, no. 7, pp. 754–767, 2022.

[14] Poonam N. Railkar Nilesh P. Sable,Vijay U. Rathod,Parikshit N. Mahalle, "An Efficient and Reliable Data Transmission Service using Network Coding Algorithms in Peer-to-Peer Network", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 10, Issue- 1s (2022), 144 – 154

[15] Prof. Shweta Jain. (2017). Design and Analysis of Low Power Hybrid Braun Multiplier using Ladner Fischer Adder. International Journal of New Practices in Management and Engineering, 6(03), 07 - 12. https://doi.org/10.17762/ijnpme.v6i03.59

[16] Merwe, M. van der, Petrova, M., Jovanović, A., Santos, M., & Rodríguez, M. Text Summarization using Transformer-based Models. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/141

[17] Dhanikonda, S. R., Sowjanya, P., Ramanaiah, M. L., Joshi, R., Krishna Mohan, B. H., Dhabliya, D., & Raja, N. K. (2022). An efficient deep learning model with interrelated tagging prototype with segmentation for telugu optical character recognition. Scientific Programming, 2022doi:10.1155/2022/1059004