IRLR: AN IMPROVED REINFORCEMENT LEARNING-BASED ROUTING ALGORITHM FOR WIRELESS MESH NETWORKS

Le Huu Binh[1], Tu T. Vo[1] and Le Duc Huy[2]

[1]Faculty of Information Technology, University of Sciences, Hue University, Hue City, Vietnam
[2]Faculty of Information Technology, Ha Noi University of Business and Technology, Vietnam

ABSTRACT

Reinforcement learning-based routing (RLR) in wireless mesh networks has recently attracted the attention of several research groups. Several recent studies have demonstrated that RLR provides higher network performs better than traditional routing protocols. In most RLR protocols, nodes use an ε-greedy policy to select data transmission routes and update their Q-value tables. With this policy, the best route is chosen with a high probability, corresponding to the exploitation phase. The remaining routes are chosen with low probability, corresponding to the exploration phase. A challenge with the ε-greedy policy in RLR protocols is that data packets transmitted in the exploration phase have a high dropped probability or a large end-to-end delay because they traverse long routes. In this paper, we propose an improved RLR for wireless mesh networks to further improve its performance. Our approach is to improve the ε-greedy policy in RLR by generating additional control packets for transmission in the exploration phase. All data packets are transmitted during the exploitation phase. Simulation results using OMNeT++ showed that he posed algorithm increases packet delivery ratio by an average value from 0.2 to 0.6%, and uces latency with an average value from 0.20 to 0.23 ms compared to the basic reinforcement learning-based routing algorithm.

KEYWORDS

Wireless mesh network, reinforcement learning-based routing, Q-learning

1. INTRODUCTION

The demand for wireless network traffic is increasing, especially in the process of comprehensive digital transformation in government agencies, businesses, and schools. To best meet this requirement, wireless mesh network (WMN) technology is a promising solution that is prioritized for use in wireless local area networks (WLAN) by network administrators because it has many advantages compared to wireless networks using traditional access points, typically reducing congestion due to the ability to balance load and convenience in deploying infrastructure because there is no need to connect wired links to all wireless routers. Consider the example shown in Figure 1, where a WMN consists of one gateway router, six mesh wireless routers (WR), and ten clients. For each pair of WRs, if they are within range of each other, a wireless link is formed between them. The set of all WRs and wireless connections forms a mesh topology. We observe that only WR1 and WR2 connect directly to the gateway router. All remaining WRs connect the gateway router through WR1 and WR2.

DOI: 10.5121/ijcnc.2024.16408

To respond well to the current explosion in traffic demand in wireless networks, it is necessary to develop solutions to improve network performance. This motivation has motivated many research groups to focus on WMN recently. Some typical topics covered most recently include mesh router node placement [2, 3, 3, 5], optimal routing protocols [6, 7], access point selection [8, 9], and network topology control [10, 11]. Each topic has its strengths in improving network performance. For example, topology control techniques are often highly efficient in terms of energy use and optimal router node placement is highly efficient in terms of network connectivity. For optimized routing protocols, many performance metrics such as throughput, end-to-end delay, quality of service (QoS), and quality of transmission (QoT) can be improved.

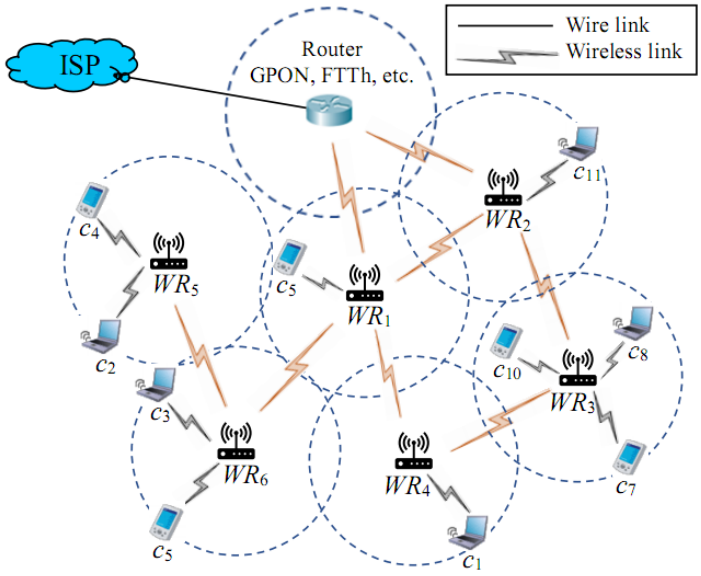

Figure 1. An example of a WMN using one gateway router, six mesh wireless routers, and ten clients [1]

WMN often uses popular routing protocols of ad hoc wireless networks, such as ad hoc on-demand distance vector (AODV) [12], destination-sequenced distance-vector routing (DSDV), optimized link state routing protocol (OLSR), and dynamic source routing (DSR) [13]. These are the basic routing protocols that use hop metrics. Therefore, routes with the fewest hops are usually prioritized. Hop-count-based routing is often ineffective in the case of heavy traffic loads and strict QoS and QoT requirements. In this case, it is necessary to use routing protocols that are capable of high-performance computing and that can quickly adapt to network state changes. Reinforcement-based routing is a suitable solution for these requirements. This solution has recently attracted the attention of many research groups. In [1], the authors investigated methods to apply RL to the routing problem in WMNs, focusing on two methods: learning through hello packets (hello-based-RL) and learning through acknowledgment packets (ACK-based-RL). The performance of these methods was evaluated by simulations using OMNET++. The simulation results demonstrate that the Hello-based-RL method outperforms the ACK-based RL method in terms of the network throughput and end-to-end delay. In [14], the authors proposed an improved AODV protocol for 5G-based mobile ad hoc networks (MANET) using reinforcement learning. Each node maintains a state information database (SIDB) that includes two metrics: traffic load and signal-to-noise ratio (SNR). The SIDB was updated regularly using the Q-learning algorithm. The new route discovery mechanism of the AODV protocol is improved by considering the constraints of traffic load and SNR in the SIDB every time a node broadcasts an RREQ packet. The simulation results using OMNET++ demonstrated that their proposed protocol outperformed the original AODV protocol in terms of throughput, end-to-end delay, and SNR. The authors of

[15] proposed a routing algorithm namely RL-based Best Path Routing (RLBPR) for WMN with the objective of choosing the best route to the gateway router. Using a simulation method with NS-2, the authors have shown that the RLBPR algorithm outperforms other algorithms in terms of end-to-end delay and throughput. In [16], the authors proposed a Q-learning-based energy-balanced routing protocol (QEBR) for WMN. The QEBR uses the principle of distributed routing. The concept of neighbor energy sorting was proposed for the reward of the Q-learning algorithm. The simulation results obtained using Python demonstrated that QEBR outperformed the conventional method. Another study presented two RL-based route choice algorithms to increase the performance of a multi-hop cognitive radio network [17], called traditional RL and RL-based with an average Q-value. Both approaches exploited the available channel time at the bottleneck link as a reward for the Q-learning algorithm. This metric is used to choose a route between two source and destination nodes. In addition, using the RL method, the authors of [18] have proposed a QoS-guaranteed intelligent routing algorithm for WMN with heavy traffic loads. They built a reward function for the Q-learning algorithm to select a route such that the packet delivery ratio was the highest. Concurrently, the learning rate coefficient is flexibly changed to determine end-to-end delay constraints. The simulation results showed that the proposed algorithm significantly improved network performance compared with other well-known routing algorithms.

The results of the abovementioned studies show that applying RL to routing control in WMN networks is a highly effective solution. In this paper, we propose a new method for applying RL to routing in a WMN to further improve network performance. The new contributions of this study are summarized as follows.

- We propose a new method to apply RL to routing in a WMN by modifying the way the agent takes action to update the Q-value table. The exploitation policy is implemented using data packets, whereas the exploration policy is implemented using a newly created control packet. This principle minimizes the situation of data packets traveling over long routes, reducing end-to-end delay, and increasing network throughput.
- We implement reinforcement-learning-based routing protocols using the OMNeT++ and INET frameworks to compare and evaluate their performance.

The remainder of this paper is organized as follows. Section 2 presents the basic methods for applying reinforcement learning to routing in a WMN. The proposed method is described in Section 3. The simulation results are presented in detail in section 4. Finally, the conclusions and suggestions for further development are presented in detail in Section 5.

2. RL-BASED ROUTING IN WMN

RL is a form of machine learning that does not require training data and operates based on the principle of trialanderror. Figure 2 illustrates the basic principles of RL. The main components of an RL system are entities that perform a learning task called agents, which perform learning by interacting with the environment through actions to change the environment and obtain a reward. In the next learning time, based on the rewards obtained in the previous learning times, the agent chooses the action that gives the best reward. Let $Q(s_t, a_t)$ be the total reward received when the agent actsa_t in state s_t. By applying the Q-learning algorithm, the value $Q(s_t, a_t)$ is determined by [19]

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha[R(s_t, a_t) + \gamma \max_{\forall a_{t+1} \in A_{t+1}} Q(s_{t+1}, a_{t+1})] \quad (1)$$

where α and $\gamma \in [0, 1]$ are the learning rate and the discount factor, respectively.

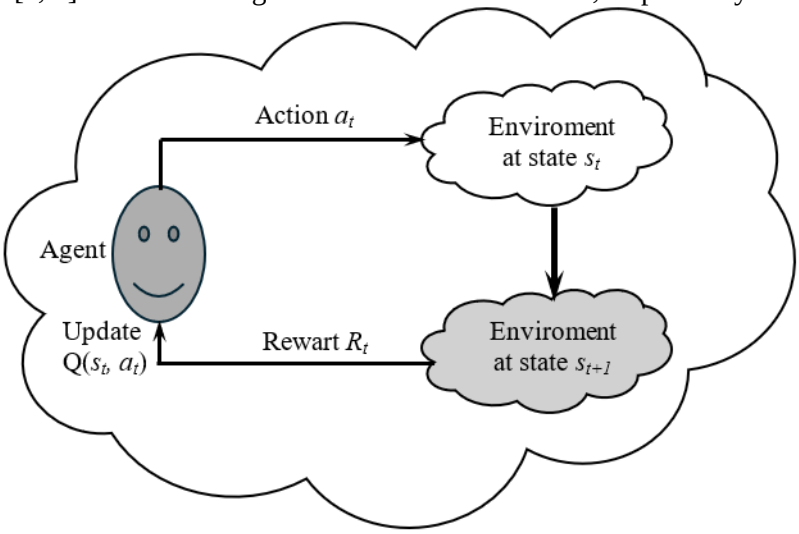

Figure 2. Illustrates the basic principles of reinforcement learning

RL can be applied to routing problems in several ways. In this section, we use the same method as in [1, 14], and [15], which uses the ACK packet to update the table of Q-values used for route selection. The process of updating the routing table at the WRs is modeled as an RL model, in which the agents are the WRs, the environment is the network system, states for each agent are a set of neighbor nodes, the status of wireless connections to those neighboring nodes, and the action is WR to select a neighbor node to transmit the data packet to the destination. In RL-based routing, the Q-value table is used as the routing table for each WR. The Q-value of each record represents the weight of the corresponding route. In the context of this study, the hop count is used as a routing metric. Therefore, the best Q-value was equivalent to that of the fewest number of hops. Therefore, the equation for updating the Q value as in (1) is modified as follows:

$$Q(c,n,d) = (1-\alpha)Q(c,n,d) + \alpha[R(c,n) + \frac{1}{\gamma}Q_{\min}(n,d)] \tag{2}$$

where $Q(c, n, d)$ represents the Q-value of the action where the current node (C) sends a data packet to the next node (N) for transmission to the destination node (D); α and $\gamma \in [0, 1]$ are the learning rate and discount factors, respectively; and $Q_{min}(n, d)$ is determined by

$$Q_{\min}(n,d) = \min_{\forall n' \in Ne(n)} Q(n,n',d) \tag{3}$$

where $Ne(n)$ is the set of neighbor nodes of N.

Figure 3 shows the flowchart of the algorithm for updating the Q-value table for each WR. First, each WR initializes the Q-value table with the structure of each record as $\{D, N, Q(c, n, d)\}$, where D represents the destination node, N represents the next node along the route to D, and $Q(cn, d)$ is the Q-value of route $C \rightarrow N \rightarrow \ldots, \rightarrow D$, where C is the current node. $Q(c, n, d)$ is set at the initialization time as follows:

$$Q(c,n,d) = \begin{cases} 1 & if \quad N \equiv D \\ X & otherwise \end{cases} \tag{4}$$

where X is a large enough value.

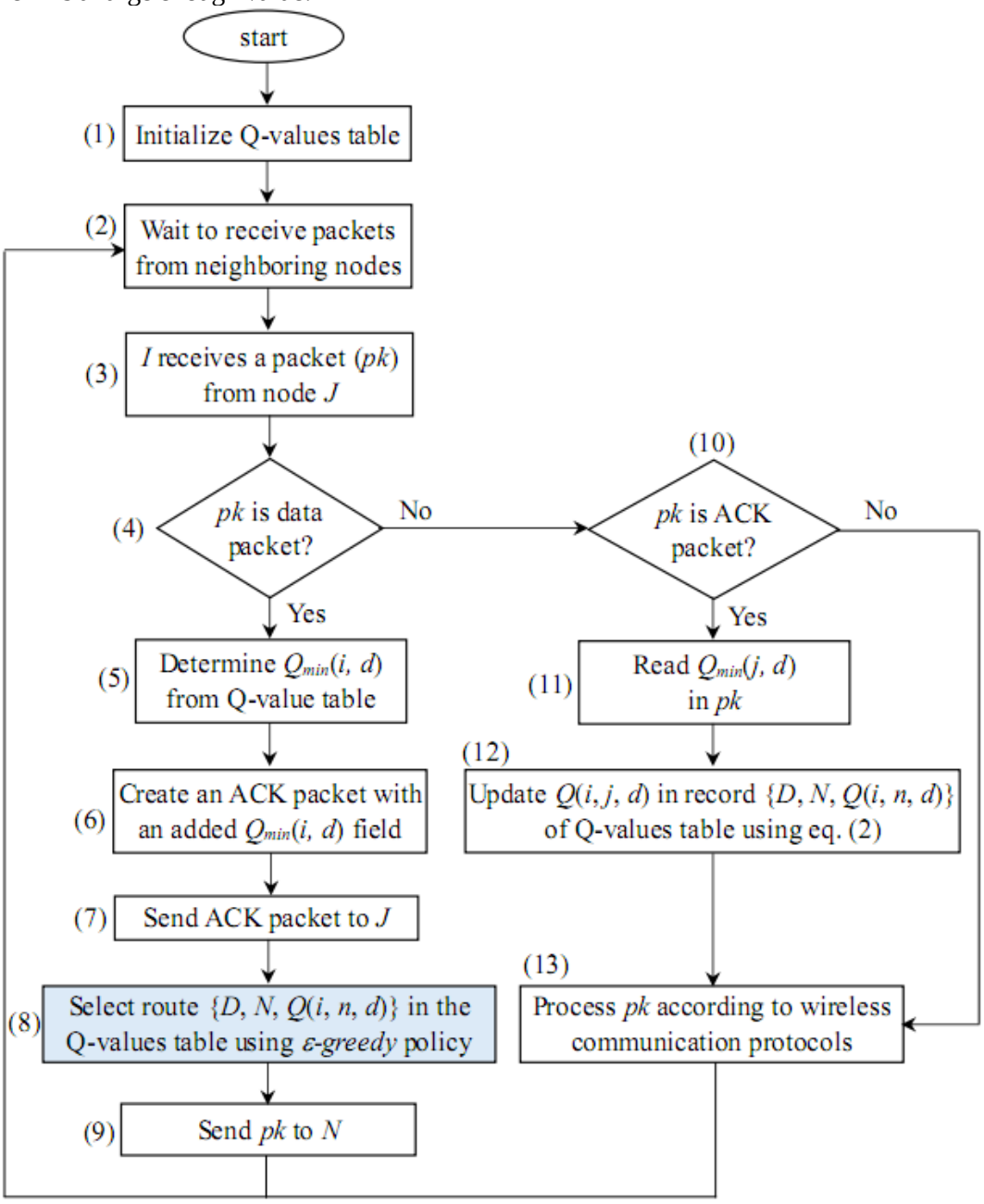

Figure 3. The algorithm updates the table of Q-values at each WR using RL

Consider an example as shown in Figure 4, at the time of initialization, node A has three neighbors, B, D, and K, in which D is the destination node. Therefore, the Q-value table was initialized using three records {D, D, 1}, {D, B, 100}, and {D, K, 100}. In this case, the value of X in (4) was set to 100. The Q-value table is updated regularly during network operations using data and ACK packets. Every time a node (I) sends a data packet to its neighbor (J), The Q-table of I is updated according to (2) if I receive an ACK packet from J. Thus, whichever neighbor node is selected to send a data packet, the Q-value of the route passing through that node is updated. For RL-based routing, an *ε-greedy* policy, as in [11], is often used to select a route for data transmission (step (8) in the algorithm of Figure 3). For this policy, the route with the best

Q-value will be chosen with a high probability of $1-\varepsilon$, and the remaining routes will be chosen with a low probability of ε. Let $\pi(c, n, d)$ be the probability of node I choosing neighbor node N to transmit the data packet to destination node D, according on ε-*greedy* policy, this probability is given by:

$$\pi(c,n,d)=\begin{cases} 1-\dfrac{\varepsilon}{|Q_{c,d}|} & if\ Q(n,n,d)=Q_{best} \\ \dfrac{\varepsilon}{|Q_{c,d}|} & otherwise \end{cases} \tag{5}$$

where $|Q_{c,d}|$ denotes the number of routes from the current node (C) to the destination node (D) in the Q-value table, Q_{best} is defined as:

$$Q_{best}=\min_{\forall n' \in Ne(n)} Q(n,n',d) \tag{6}$$

where $Ne(c)$ is the set of all neighbor nodes of node C.

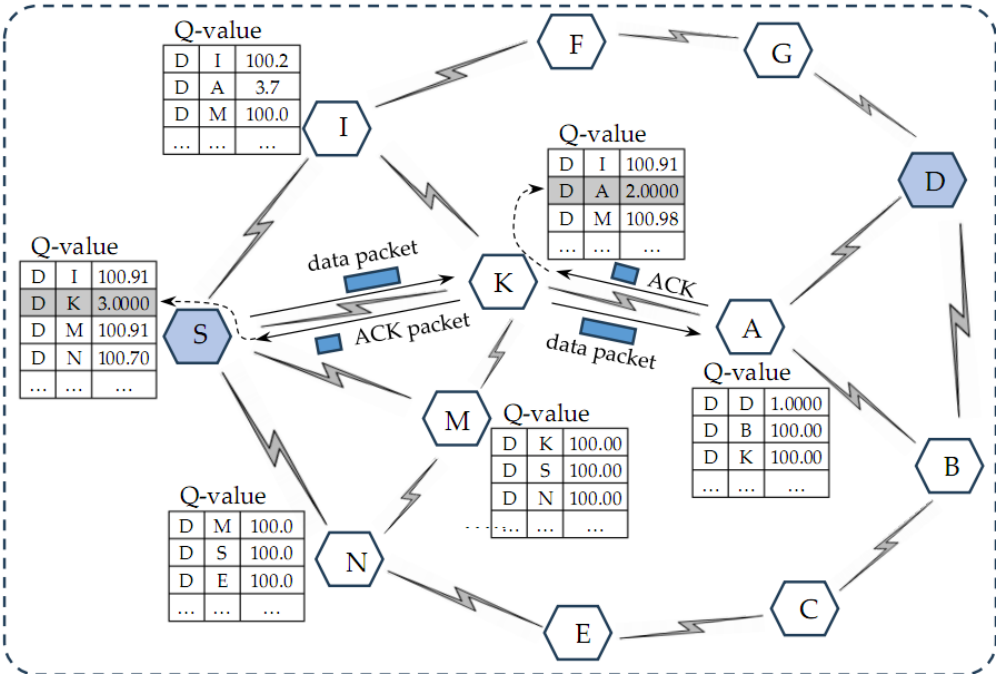

Figure 4. An example of updating Q-value table at nodes using an RL-based routing algorithm.

To clearly observe the process of updating the Q-value table using the RL-based routing algorithm, we consider an example, as shown in Figure 4. First, we analyzed the Q-value table of node K for the routes to node D. In the current state, K has three neighbor nodes, A, I and M The Q-value table of K initializes with three records, {d, a, 100}, {d, i, 100}, and {d, m, 100}, corresponding to three possible routes used to transmit data to D. Suppose the learning rate factor (α) and discount factor (γ) are set to 0.7 and 1, respectively. The factor ε of the ε-*greedy* policy is set to 0.1. After twenty times node K transmits data packets to node D, andthe Q-value table is updated, as shown in Table 1.When choosing the route to transmit the first data packet, all three routes to D have the same Q-value of 100. Therefore, each route was chosen with equal

probability $(1 - \varepsilon)/3 = 0.97$. Without a loss of generality, we assume that the route through node *I* is selected. Node *K* sends a data packet to node *I*, after receiving the ACK packet from *I*, value $Q(k, i, d)$ is updated according to (2), resulting in $Q(k, i, d) = 100.70$. For the next data packet, the two routes through *A* and *M* have better Q-values, which are chosen with a high probability. If node *M* is selected, after *K* successfully transmits the data packet to *M* and receives the ACK packet from this node, the value $Q(k, m, d)$ is updated according to (2), resulting in $Q(k, m, d) = 100.70$. From the third data transmission, the route through node *A* has the best Q-value, which will be chosen with a very high probability compared with the other two routes. After twenty times node *K* performs data transmission, the Q-value table is updated as shown in Table 1, and the route through *A* has the best Q-value. Similarly, the Q-value table of node *S* is also updated every time this node transmits data, and the results are shown in Table 2. From these results, if node *S* needs to transmit data to *D*, the route along $S \rightarrow K \rightarrow A \rightarrow D$ will be chosen with a high probability.

Table 1. The process of updating the Q-values table of node *K* using RL-based routing algorithm.

No.	Action	Update Q-values table		
		$Q(k, a, d)$	$Q(k, i, d)$	$Q(k, m, d)$
0		100.00	100.00	100.00
1	I	100.00	100.70	100.00
2	M	100.00	100.70	100.70
3	A	31.400	100.70	100.70
4	A	10.820	100.70	100.70
5	A	4.6460	100.70	100.70
6	A	2.7938	100.70	100.70
7	A	2.2381	100.70	100.70
8	A	2.0714	100.70	100.70
9	A	2.0214	100.70	100.70
10	A	2.0064	100.70	100.70
11	M	2.0064	100.70	100.91
12	A	2.0019	100.70	100.91
13	A	2.0006	100.70	100.91
14	A	2.0002	100.70	100.91
15	A	2.0001	100.70	100.91
16	A	2.0000	100.70	100.91
17	I	2.0000	100.91	100.91
18	A	2.0000	100.91	100.91
19	M	2.0000	100.91	100.97
20	A	2.0000	100.91	100.97

Table 2. The process of updating the Q-values table of node *S* using RL-based routing algorithm.

No.	Action	Update Q-values table			
		***Q*(*s, i, d*)**	***Q*(*s, k, d*)**	***Q*(*s, m, d*)**	***Q*(*s, n, d*)**
0		100.00	100.00	100.00	100.00
1	I	100.70			
2	M			100.70	
3	N				100.70
4	K		32.10		
5	K		11.73		
6	K		5.6190		
7	K		3.7857		
8	K		3.2357		
9	K		3.0707		
10	K		3.0212		
11	K		3.0064		
12	M			100.91	
13	K		3.0019		
14	K		3.0006		
15	K		3.0002		
16	K		3.0001		
17	K		3.0000		
18	K		3.0000		
19	K		3.0000		
20	I	100.91			

3. PROPOSED METHOD

In this section, we present the proposed algorithm called IRLR (Improved Reinforcement Learning-based Routing). The IRLR algorithm focuses on improving the route selection policy for data transmission in step (8) of the algorithm, as shown in Figure 3. According to the principle of *ε-greedy*, routes whose Q value is not the best are also selected to transmit data with a probability ε corresponding to the exploration phase in the RL-based routing algorithm. Data packets transmitted through these routes have a high probability of dropping because these routes may or may not exist. If it exists, the end-to-end delay of the data packet will also be large because these routes have high Q-values, meaning that they go through many transmission hops and intermediate nodes. To overcome this drawback, we propose a method to improve the learning action policy at each node, as shown in the algorithm flowchart in Figure 5. In contrast to the basic RL-based routing algorithm, the exploration phase is performed using EXP packets instead of data packets (steps (11) to (14) in the algorithm shown in Figure 5. All data packets are transmitted in the exploitation phase using routes with the best Q-values. This reduces the number of dropped data packets and end-to-end delay and increases the network throughput.

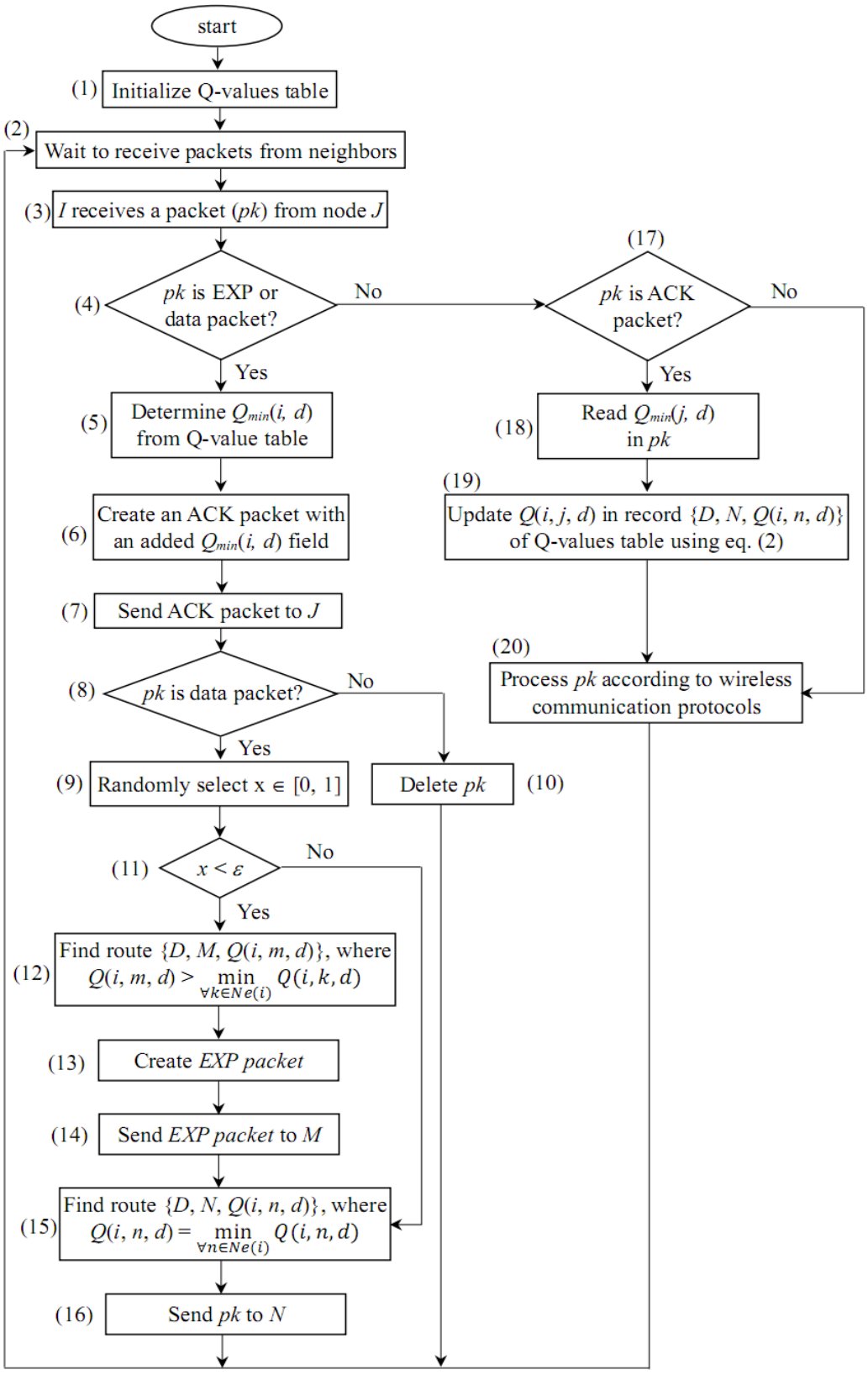

Figure 5. Flowchart of improved RL-based routing algorithm.

4. EVALUATE PERFORMANCE BY SIMULATION

In this section, we use simulations to evaluate the performance of the proposed method. The simulation was installed on the Ubuntu 22.04 operating system, using the open-source software OMNeT++ 6.0.3 [20] and INET framework 4.5.2 [21].

4.1. Simulation Scenarios

The simulation scenario is presented in Table 3. A WMN is installed in an area of 1000 ×1000 [m^2], the number of WRs varies from 30 to 50 in steps of 5, and the coverage radius of each WR is 250 [m]. The MAC protocol used was IEEE 802.11ac with a carrier frequency of 2.4 [GHz], and the data rate of each channel was 54 [Mbps]. The IRLR algorithm is compared with the basic RL-based routing algorithm [22] in terms of packet delivery ratio, network throughput, and end-to-end delay. Because of the randomness of the RL algorithm, each simulation scenario was repeated 20 times to ensure the accuracy of the results, and the results presented in this section are the average of 20 simulations.

Table 3. Simulation parameters.

Parameter	**Setting**
Network area	1000 × 1000[m2]
Number of WRs	30:5:50
Communication range	250 [m]
MAC protocol	IEEE 802.11ac
Data rate	54 [Mbps]
Learning rate factor (α)	0.7
Discount factor (γ)	1.0
ε factor of ε-greedy policy	0.1
Number of runs of a scenario	20

4.2. Simulation Results

The first performance metric examined in this section is packet delivery rate (PDR). In our context, PDR is calculated as the percentage of data packets that are successfully transmitted to the destination node and the number of data packets generated throughout the network. In Figure 6, we compare PDR using the basic RL-based routing (BRLR) algorithm and the proposed IRLR algorithm in the case where a WMN uses 45 WRs. The box charts in this figure represent data from 20 runs for each simulation scenario. We can observe that the higher the traffic load, the lower the PDR for both algorithms. However, the IRLR algorithm always yielded a higher PDR than the BRLR algorithm. Considering the case of an average traffic load of 1 Mbps, when using the BRLR algorithm, the PDR varied from 99.16% to 99.85% with a median and average of 99.69% and 99.60%, respectively. When using the IRLR algorithm, the value range of the higher PDR ranges from 99.40% to 100%, with a median and average of 99.71% and 99.70%, respectively. Thus, both the median and mean values of the IRLR algorithm were larger than those of the BRLR algorithm. Comparing the different cases of traffic load, we can observe that the higher the traffic load, the more effectively the IRLR algorithm operates. Considering the case of a 10 Mbps traffic load, the median and mean values of PDR are 97.57% and 97.56%, respectively, for the case of the BRLR algorithm. These values were 97.98% and 98.00%, respectively, for the IRLR algorithm. The results are also completely similar for the 50 WRs

simulation scenario, as shown in Figure 7, and the PDR of the IRLR algorithm is always higher than that of the BRLR algorithm.

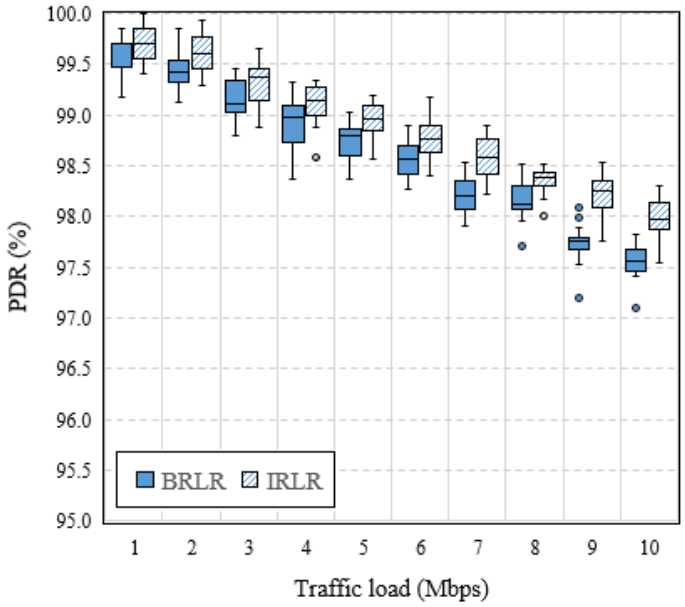

Figure 6. Evaluate PDR versus traffic load in the case of network size of 45 WRs.

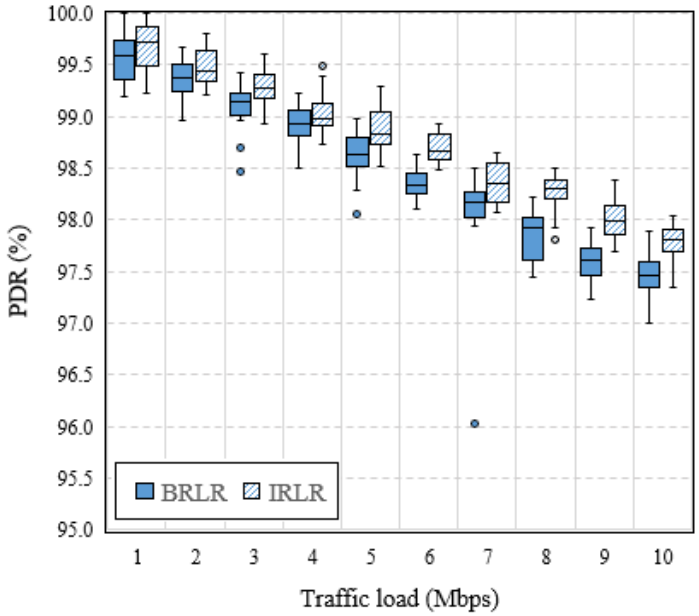

Figure 7. Evaluate PDR versus traffic load in the case of network size of 50 WRs.

When simulating scenarios with different total WRs, the ILRL algorithm always outperformed the BRLR algorithm in terms of PDR. This is clearer from the results obtained in Figure 8 and Figure 9, where we carefully studied five simulation scenarios with total WRs of 30, 35, 40, 45,and 50, respectively. The PDR of the IRLR algorithm is always higher than that of the BRLR algorithm for both cases, where the traffic load is 5 Mbps (Figure 8) and 10 Mbps (Figure 9).

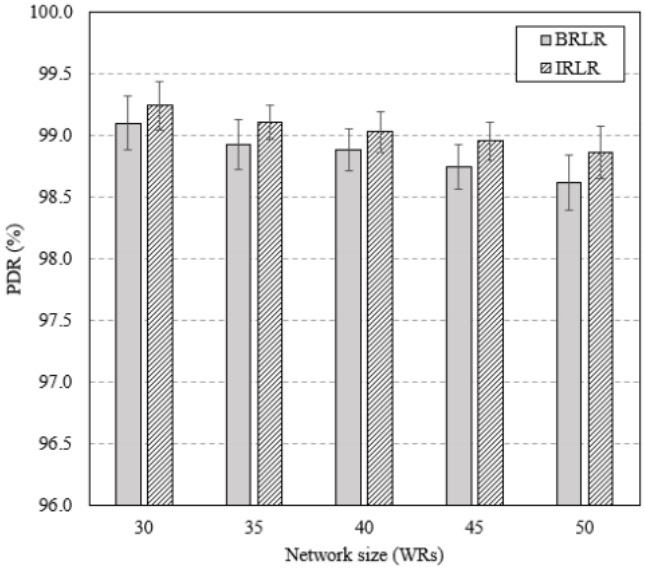

Figure 8. Evaluate PDR versus network size in the case of the traffic load of 5 Mbps.

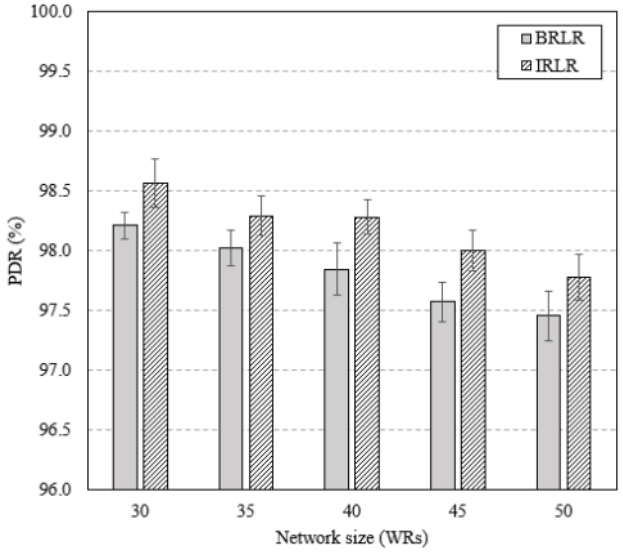

Figure 9. Evaluate PDR versus network size in the case of the traffic load of 10 Mbps.

The next important metric that is carefully studied in this section is end-to-end delay (EED). Figure 10 shows the results obtained when repeating 20 simulations for the scenario where the number of WRs is 50 and the average traffic load is 3 Mbps. We can easily observe that the IRLR algorithm yields a lower average EED than the BRLR algorithm does. Consider the case of a traffic load of 3 Mbps (results in Figure 10). When using the BRLR algorithm, the average EED of all 20 simulation runs was 1.601 [ms]. The value of the IRLR algorithm is 1.302 [ms]. Thus, the IRLR algorithm reduced the EED by an average value of 0.299 [ms] compared to the BRLR algorithms. The results are also completely similar to those of the simulation scenario, where the average traffic load is 10 Mbps, as shown in Figure 11. The average EED when using the BRLR and IRLR algorithms were 1.525 [ms] and 1.298 [m], respectively. Thus, the IRLR algorithm reduced EED by an average value of 0.23 [ms] compared to the BRLR algorithm for this simulation scenario.

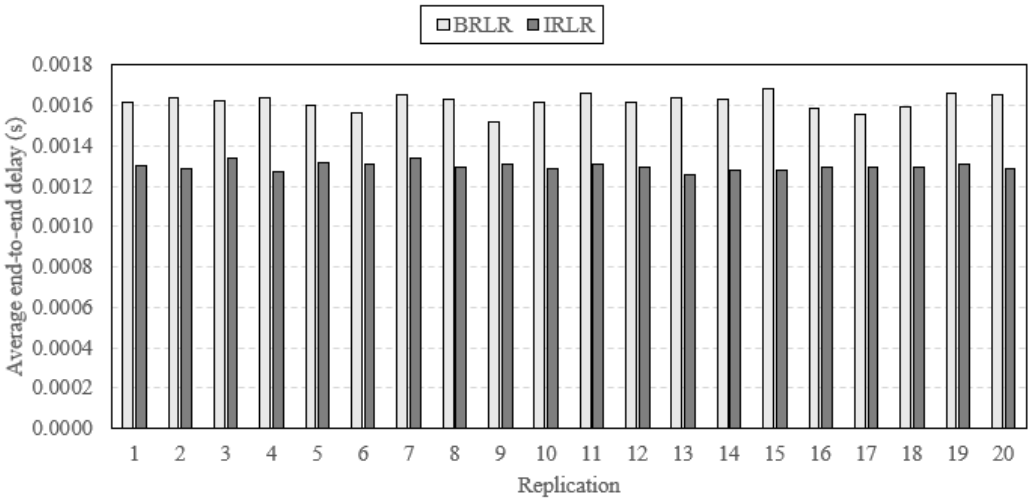

Figure 10. Evaluate the average end-to-end delay in the case of 50 WRs and 3 Mbps traffic load

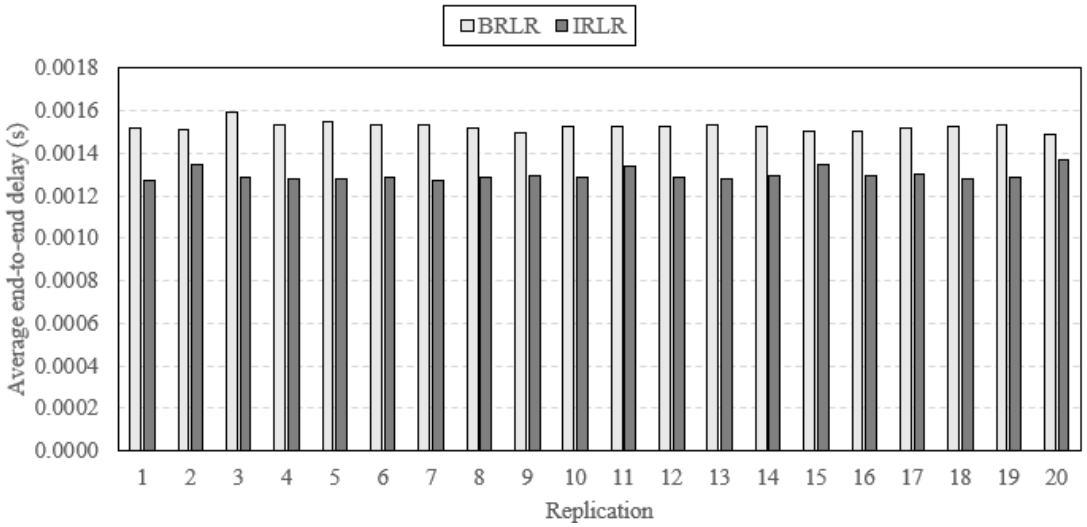

Figure 11. Evaluate the average end-to-end delay in case of 50 WRs and 10 Mbps traffic load

The influence of traffic load on EED was also investigated with the results shown in Figure 12 and Figure 13 for simulation scenarios where the number of WRs is 40 and 50, respectively. We observe that the average EED decreases when the traffic load is high. This is because, when the traffic load is high, the algorithm converges faster, leading to WRs quickly finding the best route to transmit data. Comparing the BRLR and IRLR algorithms, the IRLR algorithm always provides a better average EED than the BRLR algorithm. The average improvement was 0.224 [ms] and 0.203 [ms] for the 40 WRs and 50 WRs scenarios, respectively.

From the simulation results presented above, we conclude that the proposed algorithm, IRLR, outperforms the basic RL-based routing in terms of PDR and EDD. This is a very meaningful result for improving the WMN performance.

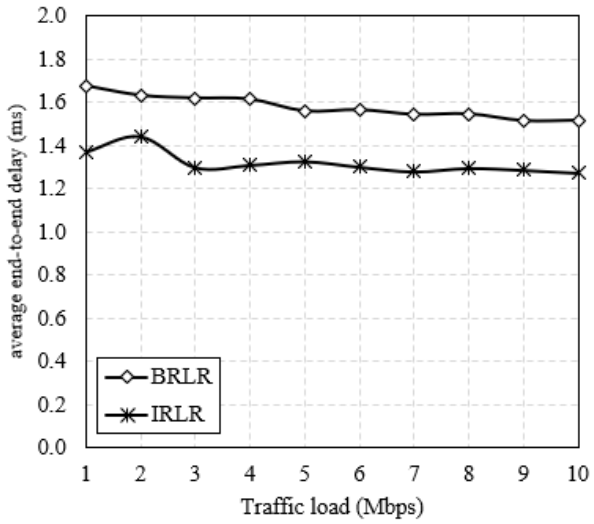

Figure 12. Evaluate the average end-to-end delay versus traffic load in the case of 40 WRs

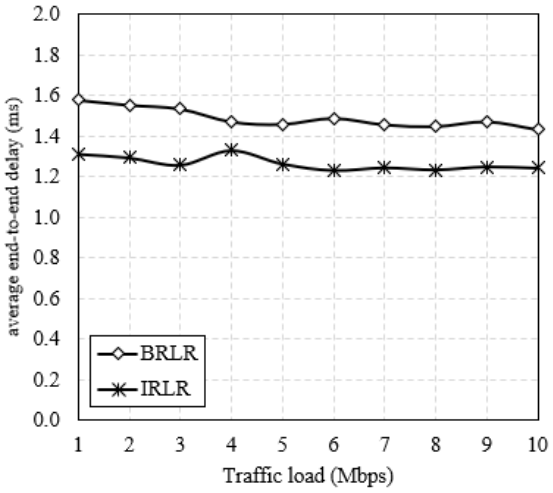

Figure 13. Evaluate the average end-to-end delay versus traffic load in the case of 50 WRs

5. CONCLUSIONS

To respond well to the current explosion in wireless network traffic demand, researching advanced routing techniques for WMN networks is particularly important to improve performance. RL-based routing has recently attracted the attention of several research groups owing to its many advantages compared to traditional routing protocols. In this study, we proposed an improved RL algorithm for routing in wireless mesh networks to further improve its performance. Our approach modifies the exploitation and exploration policies during the learning process, which are implemented using data and control packets. Simulation results using OMNeT++ showed that the proposed method provides superior performance compared to basic RL-based routing in terms of PDR and EED.

In the future, we will further develop this work by considering QoS-guaranteed routing techniques using reinforcement learning or deep reinforcement learning to further improve the performance of WMNs.

ACKNOWLEDGEMENTS

This work is sponsored by the Science and Technology Project of Hue University under grant number DHH2023-01-204.

REFERENCES

[1] L. H. Binh and N. N. Thuy, "Survey and evaluation of application methods of reinforcement learning for routing in wireless mesh networks," *Journal of Science and Technology, Hue University of Science, Issues in Mathematics - Information Technology - Physics - Architecture*, vol. 23, no. 1, pp. 1–14,2023.

[2] N. N. Abdelkader, A. Zibouda, A. Naouri, and H. Soufiene, "An efficient mesh router nodes placement in wireless mesh networks based on moth flame optimization algorithm," *International Journal of Communication Systems*, vol. 36, 2023.

[3] L. H. Binh and T.-V. T. Duong, "A Novel and Effective Method for Solving the Router NodesPlacement in Wireless Mesh Networks using Reinforcement Learning," *PLOS ONE*, vol. 19, no. 4,e0301073, 2024.

[4] T. S. Mekhmoukh, M. Yassine, G. A. Benmessaoud, M. Seyedali, Z.Atef, and R.-C.Amar, "Solving the Mesh Router Nodes Placement in Wireless Mesh Networks using Coyote Optimization Algorithm," *IEEE Access*, vol. 10, pp. 52 744–52 759,2022.

[5] L. H. Binh and T. T. Khac, "An Efficient Method for Solving Router Placement Problem in Wireless Mesh Networks Using Multi-Verse Optimizer Algorithm," *Sensors*, vol. 22, no. 15, 2022.

[6] E. O. Steven and E. Kamwesigye, "A Deep Learning-Based Routing Approach for Wireless Mesh Backbone Networks," *IEEE Access*, vol. 11, pp. 49 509–49 518, 2023.

[7] Binh LH, Duong TVT. Load balancing routing under constraints of quality of transmission in mesh wireless network based on software -defined networking, *Journal of Communications and Networks*, 2021;23(1):12–22, 2021.

[8] R. Ding, Y. Xu, F. Gao, X. Shen and W. Wu, "Deep Reinforcement Learning for Router Selection in Network With Heavy Traffic," *IEEE Access*, vol. 7, pp. 37109-37120, 2019.

[9] Raschellà Alessandro, Bouhafs Faycal, Mackay Michael, Shi Qi, Ortin Jorge, Gallego Jose Ramon, Canales María, A Dynamic Access Point Allocation Algorithm for Dense Wireless LANs Using Potential Game, *Computer Networks*, 167, 2019.

[10] Binh LH, Duong T-VT, "A novel and effective method for solving the router nodes placement in wireless mesh networks using reinforcement learning", *PLoS ONE*, vol.19, no.4, e0301073, 2024.

[11] Le T, Moh S, "An Energy-Efficient Topology Control Algorithm Based on Reinforcement Learning for Wireless Sensor Networks", *International Journal of Control and Automation*,vol.10, pp.233-244, 2017.

[12] C. Perkins, E. B. Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *RFC 3561*.

[13] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile AdHoc Networks for IPv4," *RFC4728*.

[14] L. H. Binh and T.-V. T. Duong, "An improved method of AODV routing protocol using reinforcement learning for ensuring QoS in 5G-based mobile ad-hoc networks," *ICT Express*, vol. 10,no. 1, pp. 97–103, 2024.

[15] M. Boushaba, A. Hafid, and A. Belbekkouche, "Reinforcement learning-based best path to best gateway scheme for wireless mesh networks," *in 2011 IEEE 7th Int. Conf. on Wireless and Mobile Computing, Netw. and Comm.*, 2011, pp. 373–379.

[16] M. Yin, J. Chen, X. Duan, B. Jiao, and Y. Lei, "QEBR: Q-Learning Based Routing Protocol for Energy Balance in Wireless Mesh Networks," *in 2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 280-284.

[17] A. R. Syed, K. A. Yau, J. Qadir, H. Mohamad, N. Ramli, and S. L. Keoh, "Route Selection for Multi-Hop Cognitive Radio Networks Using Reinforcement Learning: An Experimental Study, "*IEEE Access*, vol. 4, pp. 6304–6324, 2016.

[18] T.-V. T. Duong, L. H. Binh, and V.M. Ngo, "Reinforcement learning for QoS-guaranteed intelligent routing in Wireless Mesh Networks with heavy traffic load," ICT Express, vol. 8, no. 1, pp. 18–24,2022.

[19] V. Duong Thi Thuy and L. Binh, "IRSML: An intelligent routing algorithm based on machine learning in software defined wireless networking," *ETRIJournal*, vol. 44, 08 2022.

[20] A. Varga and OpenSim Ltd., OMNeT++ Simulation Manual, Version 6.x. [Online]. Available:https://omnetpp.org, 2024.

[21] A. Virdis and M. Kirsche, *Recent advances in network simulation - The OMNeT++ environment and its ecosystem*, Springer Nature Switzerland AG,2019.

[22] N. Q. Cuong, M. T. Tho, L. H. Binh and V. T. Tu, "RLMR: A method of applying Q-learning for routing in mobile adhoc networks", *Proceedings of the 16th National Conference on Fundamental and Applied Information Technology Research (FAIR'2023),* Danang, Vietnam, 28-29/3/2023, pp. 741-784.

AUTHORS

Le Huu Binh received his BE degree in Telecommunications and Electronics from Da Nang University of Technology, Vietnam, his MSc degree in Computer Sciences from Hue University of Sciences, Vietnam, and his PhD degrees in Informatics from Vietnam Academy of Science and Technology (GUST) in 2001, 2007, and 2020, respectively. He worked as a senior engineer with Transmission and Switching Exchange of the Hue Telecommunications Centre, Thua Thien Hue of the Vietnam Posts and Telecommunications Group (VNPT) from 2001 to 2009. From 2010 to 2021, he worked at the Hue Industrial College (HUEIC), Vietnam, where he was the dean of the Faculty of Information Technology and Telecommunications. Since the beginning of 2022, he has been with the Faculty of Information Technology, University of Sciences (HUSC), Hue University, Hue City, Vietnam, where he is now a lecturer. His current research interests are the next generation wireless network technology, software defined networking, the application of machine learning, and artificial intelligence in network technology.

Tu T. Vo is an associate professor in the Faculty of Information Technology, Hue University of Sciences, Hue University. He received B.E. degree in Physics from Hue University in 1987and Ph.D. degree in computer science from Institute of Information Technology, Vietnam Academy of Science and Technology in 2005. His fields of interest are network routing, analysis and evaluation of network performance, security wireless Ad hoc Network.

Le Duc Huy was born in Bac Ninh province, Vietnam in 1990. He received B.E. degree in Information Technology from Hanoi University of Business and Technology, 2012 and M.A. degree in Computer Science from the Thai Nguyen University Of Information And Communication Technology, 2015. He is currently studying for his Ph.D. in Graduate University of Sciences and Technology; Vietnam Academy of Science and Technology. His research interests include computer network and network security.