

Adaptive Integration of Distributed Deep Q-Networks for Enhancing OLSR Routing in Dynamic Mobile Ad-Hoc Networks

Alon Jala Tirta Segara*¹, Arief Rais Bahtiar², Muhammad Raafi'u Firmansyah³, Fahrudin Mukti Wibowo⁴

¹⁻²Software Engineering, Telkom University, Indonesia

³Informatic, Telkom University, Indonesia

⁴Information Technology, Universiti Tun Husein Onn, Malaysia

E-mail: alonhs@telkomuniversity.ac.id¹, ariefbahtiar@telkomuniversity.ac.id², raafi@telkomuniversity.ac.id³, hi200015@siswa.uthm.edu.my⁴

Abstract. Adaptive routing in Mobile Ad-Hoc Networks (MANETs) poses considerable difficulty owing to the network's dynamic characteristics, lack of stable infrastructure, and swift topology alterations. The Optimized Link State Routing (OLSR) protocol provides a proactive routing mechanism via topology dissemination and MultiPoint Relay (MPR) selection. Nevertheless, it exhibits diminished responsiveness to real-time topology alterations, as it depends on periodic updates and does not explicitly account for link quality. This paper suggests the incorporation of the Deep Q-Network (DQN) methodology into OLSR as a reinforcement learning strategy to improve routing adaptability and efficiency. The DQN model employs network metrics like latency, ETX, buffer occupancy, and neighbor count as state inputs, with actions determined by Q-values obtained via environmental interactions. Simulations conducted with NS-3 and PyTorch demonstrate that OLSR-DQN enhances Packet Delivery Ratio (PDR) by as much as 20%, decreases delay by 15–25%, and markedly boosts throughput in dynamic MANET situations.

Keywords: MANET, OLSR, Deep Q-Network, adaptive routing, reinforcement learning

1. Introduction

A Mobile Ad-Hoc Network (MANET) is a wireless network composed of mobile nodes that may autonomously establish a network topology without the necessity of fixed infrastructure. In a MANET, each node operates both as a host and as a router, capable of forwarding data packets to other nodes. The value of MANET lies in its adaptability and capacity to function in unanticipated circumstances, including war contexts, natural disasters, or remote locations inaccessible to traditional networks [1]. Nonetheless, the adaptability of MANET poses considerable difficulties, especially with routing. The fast changes in network architecture due to high mobility complicate the selection of the ideal path [2]. Traditional routing protocols frequently encounter difficulties in effectively adapting to network changes. Optimized Link State Routing (OLSR) is a widely utilized proactive technique that employs comprehensive topological data to regularly establish routes and incorporates the MultiPoint Relay (MPR) concept to minimize the overhead associated with broadcasting routing information [3]. While OLSR exhibits benefits in relatively stable

networks, its efficacy significantly diminishes under high-mobility conditions. This occurs due to the lag in identifying topological alterations, as information updates are performed periodically rather than on demand. Moreover, the MPR selection and route construction algorithms in OLSR rely solely on hop count, disregarding connection quality factors such as delay, congestion, or packet loss rate. Consequently, OLSR demonstrates reduced adaptability to dynamic network conditions, leading to higher delays, increased packet loss, and decreased throughput [4]. To overcome these limitations, machine learning approaches, particularly Deep Reinforcement Learning (DRL), have been proposed as viable alternatives. The Deep Q-Network (DQN) is a widely used DRL technique that enables agents to learn optimal actions through interactions with the environment, guided by reward values [5]. In the context of MANET routing, DQN can be employed to enhance MPR selection and modify routing tables according to real-time network conditions. This research proposes the integration of DQN into the OLSR protocol, creating a new version termed OLSR-DQN, and evaluates its performance through simulations on key parameters such as Packet Delivery Ratio (PDR), delay, throughput, and convergence time. Recent studies have explored reinforcement learning-based approaches in ad-hoc and vehicular networks. Sharma et al. proposed an adaptive deep reinforcement learning method for MANET routing using NS-3, which improved efficiency but did not address the constraints of VANET environments, such as restricted mobility patterns and frequent topology changes [6]. Zhang et al. applied DQN in the Internet of Vehicles (IoV) context using SUMO–NS-3 integration, demonstrating improved PDR but focusing only on simple grid-based topologies without optimizing OLSR or GPSR [7]. Wang et al. extended reinforcement learning to urban VANET scenarios, but their centralized learning approach raised scalability issues [8].

In contrast, the novelty of this study lies in the integration of a distributed Deep Q-Network (DQN) into the OLSR protocol, enabling each MANET node to autonomously learn optimal forwarding strategies based on local parameters (ETX, delay, buffer occupancy, and neighbor density). Unlike prior studies that relied on centralized or offline training, OLSR-DQN facilitates decentralized and real-time decision-making derived from node-specific experiences. This research is among the first to directly embed reinforcement learning into OLSR while preserving its proactive framework, thereby offering enhanced adaptability and scalability. Furthermore, the proposed protocol is validated in realistic mobility scenarios using integrated SUMO–NS-3 simulation, providing comprehensive insights into the trade-offs between routing performance and overhead.

2. Research Methodology

The research process is structured into five sequential phases to ensure systematic development and evaluation (Figure 1). The first phase, Architecture Design, focuses on defining the overall system structure, including the integration of OLSR with Deep Q-Network (DQN). Next, the DQN Implementation phase involves developing reinforcement learning modules that adaptively optimize routing decisions. The third phase, Simulation Setup (NS-3 + PyTorch), establishes the experimental environment by combining the NS-3 network simulator with PyTorch for machine learning execution. Following this, the Parameter Testing phase is conducted to evaluate the influence of various networks and learning parameters on performance. Finally, in the Result Analysis phase, the outcomes are examined through key performance metrics such as Packet Delivery Ratio (PDR), end-to-end delay, and routing overhead to determine the effectiveness of the proposed approach.

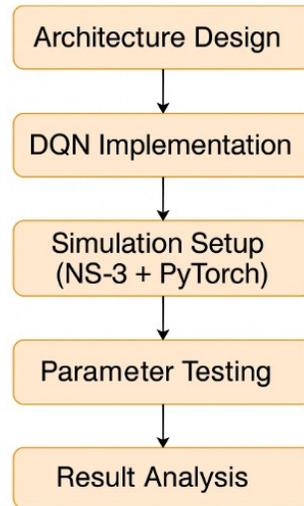


Figure 1. Research process flow

2.1. *Architecture OLSR-DQN*

The OLSR-DQN architecture integrates the OLSR protocol with a Deep Q-Network (DQN) agent that functions on each node inside a MANET, as depicted in Figure 2. Each agent is assigned the responsibility of making routing decisions adaptively according to the prevailing local network conditions. The state monitored by the agent encompasses delay to neighbors, ETX value, buffer occupancy, neighbor count, and residual energy. In accordance with this state, the agent selects actions include updating the MPR, substituting routing table entries, or preserving current routes [4]. The Q value for each action is acquired through experiential engagement, with rewards allocated depending on improvements in PDR, reductions in time, or enhancements in path stability [5]. The DQN model is a three-layer neural network, with 64 neurons per layer, utilizing ReLU activation [6]. The ϵ -greedy strategy facilitates exploration and exploitation, with updates executed via Q-learning and the mean squared error loss function. The inference process occurs online at each node to facilitate real-time decision-making. This method enables the OLSR-DQN protocol to adeptly and effectively modify routing algorithms in response to alterations in network topology.

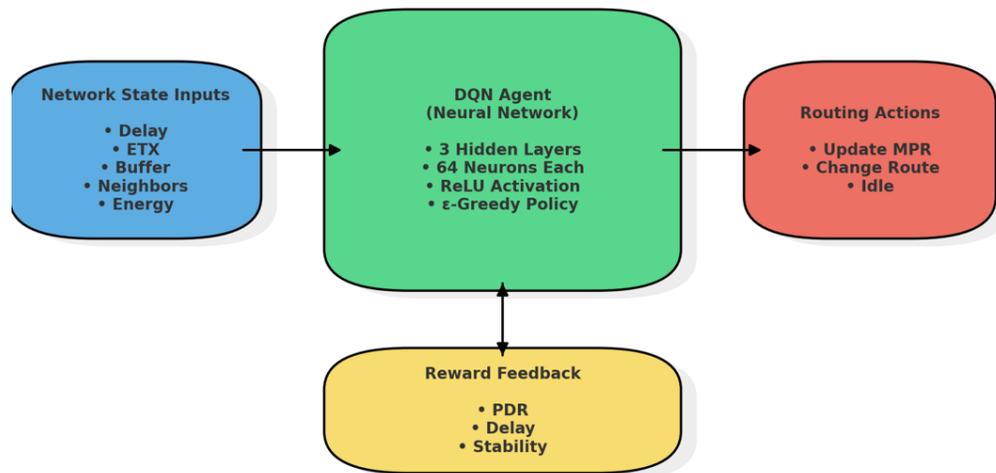


Figure 2. Block diagram OF OLSR-DQN Architecture

2.2. Algoritma OLSR-DQN

The OLSR-DQN technique is formulated inside the Deep Reinforcement Learning framework, particularly utilizing the Deep Q-Network (DQN), implemented in a decentralized fashion at each node within the network [7]. The primary aim of this technique is to allow each node to adaptively choose the optimal path and MultiPoint Relay (MPR) based on local network condition data. The algorithm functions through an interaction cycle between agents (nodes) and the environment (network topology) as outlined below [8]:

1. Initialization
Each node initializes the DQN model in the form of a deep neural network to map the Q values from state-action combinations. Replay buffer is also prepared to store experiences (s, a, r, s').
2. Observasi State (s)
Node observes its environment conditions, which are represented as a state vector [9]:

$$s=[dlink,ETX,qnode,Nneigh,Eres]$$

3. Action Selection (a)
Using the ϵ -Greedy strategy, the node selects an action:
 - Updating MPR selection
 - Changing routes in the routing table
 - Not making any changes (idle)
 - Exploration is done when $\text{rand}() < \epsilon$ and exploitation is done by selecting the action with the highest Q value.
4. Execution of Action and Observation of Reward (r)
After the action is executed, the node calculates the reward based on the impact on network performance [10], such as:

$$r = w1 \cdot PDR - w2 \cdot Delay - w3 \cdot Loss$$

Where $w1, w2, w3$ are weighed empirically.

5. Learning and Q-value Update
Experience (s, a, r, s') is stored in the replay buffer. Mini-batch samples from the buffer are used to update the DQN model with the loss function [11]:

$$Loss = (r + \gamma \cdot a' \max_{a'} Q(s', a') - Q(s, a))^2$$

The discount factor is denoted by γ .

6. Decrease in ϵ Value

To reduce exploration over time, the value of ϵ is gradually decreased [6]:

$$\epsilon_{new} = \max(\epsilon_{min}, \epsilon \cdot decay)$$

7. Cycle Repetition

The process is repeated periodically until the network reaches stability or the simulation is completed. With this approach, OLSR-DQN does not only rely on static routing logic like standard OLSR but can adjust routing strategies based on historical experience and real-time changing network conditions [12].

2.3. Simulation Settings

The performance evaluation of the OLSR-DQN protocol is conducted using several quantitative parameters commonly used in MANET network research. Each parameter provides a comprehensive overview of the efficiency, reliability, and adaptability of the protocol to dynamic topological conditions [13]. Here is a detailed explanation of each evaluation metric along with its formula and interpretation:

1. Packet Delivery Ratio (PDR)

Definition: The ratio of the number of packets successfully received at the destination side to the number of packets sent from the source side [14].

Formula:

$$PDR = (P_{delivery} / P_{received}) \times 100\%$$

Unit: Percentage (%)

Interpretation: This metric reflects the reliability of the protocol in delivering data. The higher the PDR, the better the protocol's performance.

2. Average End-to-End Delay

Definition: The average time it takes for a package to reach its destination since being sent from the source [7].

Formula:

$$Delay = \frac{1}{n} \sum_{i=1}^n (t_{received, i} - t_{delivery, i}) \quad (7)$$

Unit: Milliseconds (ms)

Interpretation: Measuring transmission time efficiency. A low delay value is important for real-time applications such as VoIP or tactical military systems.

3. Throughput

Definition: The amount of data successfully received by the receiver per unit of time during the simulation [15].

Formula:

$$Throughput = \frac{Simulation\ duration}{Total\ bits\ received}$$

Unit: Kilobits per second (Kbps) or Megabits per second (Mbps)

Interpretation: Indicates the network's capacity to transmit data effectively. The higher the throughput, the better the protocol's efficiency.

4. Packet Loss Ratio

Definition: The percentage of packets that are lost or fail to reach their destination during transmission [5].

Formula:

$$Loss = (P \text{ delivery} - P \text{ received}) \times 100\%$$

Unit: Percentage (%)

Interpretation: This metric indicates the stability of the communication path. A low value indicates a stable protocol that is not prone to packet loss.

5. Routing Overhead

Definition: The number of control packets sent by the routing protocol compared to the total number of data packets [15].

Formula:

$$Overhead = Data \text{ packet}/control \text{ packet ratio}$$

Unit: Ratio (unitless)

Interpretation: This metric measures the efficiency of the protocol in delivering routing information. Low values are expected to reduce network load.

2.4. Test Parameters

The simulation was conducted with the following configuration, as presented in Table 1:

Table 1. Test Parameters

| Parameter | Value | Description |
|-----------------------|--------------------------------------|---|
| Simulator | NS-3 | Used to simulate the topology of a MANET network |
| Backend DQN | PyTorch | For training and inference of the DQN model |
| Simulation Area | 1000 x 1000 meter | Square area without physical obstacles |
| Number of Nodes | 20, 30, dan 50 | Representing light, medium, and dense scenarios |
| Mobility Model | Random Waypoint | Node moves randomly with a constant speed. |
| Node Speed | 1 – 20 m/s | Mimicking the dynamics of fast-moving vehicles or personnel |
| Traffic | CBR (UDP), 512-byte, interval 0.25 s | Constant traffic between nodes with fixed packet size |
| Simulation Duration | 300 detik | Long enough to see the medium-term effects |
| Number of Repetitions | 20 times | Each configuration is rerun for statistical averaging. |

| Parameter | Value | Description |
|---------------------|----------------------------|------------------------------------|
| Comparison Protocol | OLSR (standar) vs OLSR-DQN | To measure performance improvement |

3. Results and Discussion

The simulation was conducted to compare the performance of the standard OLSR protocol with the OLSR-DQN protocol on a MANET network. The evaluation was conducted on three node count scenarios, namely 20, 30, and 50 nodes, to observe the impact of network density on routing performance. Four main parameters were evaluated: Packet Delivery Ratio (PDR), Average End-to-End Delay, Throughput, and Routing Overhead.

20-Node Scenario (Low Density)

In this configuration, the number of vehicles is relatively small, resulting in a sparse network. Routing becomes simpler due to reduced interference; however, there is a higher risk of link disconnection as the distance between nodes increases. OLSR-DQN mitigates this risk by adaptively selecting routes based on link quality metrics, thereby maintaining a stable Packet Delivery Ratio (PDR), as illustrated in Figure 3.

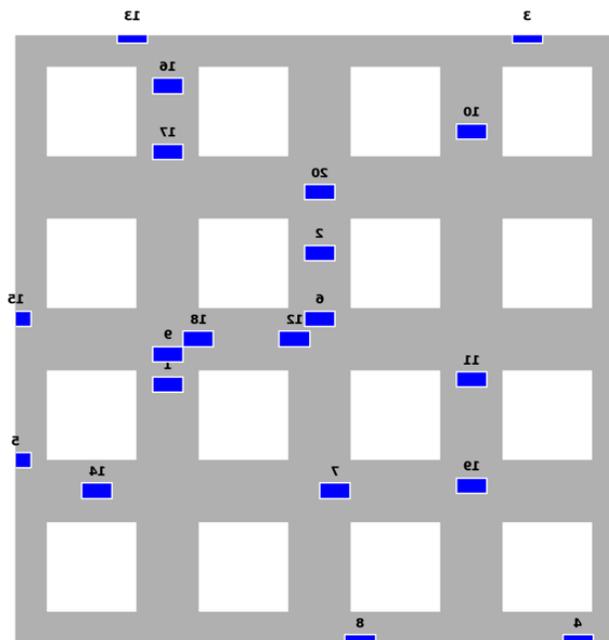


Figure 3. Urban Road With 20 Vehicles (20 node)

30-Node Scenario (Medium Density)

At medium density, the network achieves greater stability because the average distance between nodes is shorter, allowing multiple alternative routes to form. OLSR-DQN demonstrates noticeable improvements compared to standard OLSR, particularly in reducing end-to-end delay and increasing throughput, as the DQN agent avoids congested paths and dynamically selects higher-quality routes, as shown in Figure 4.

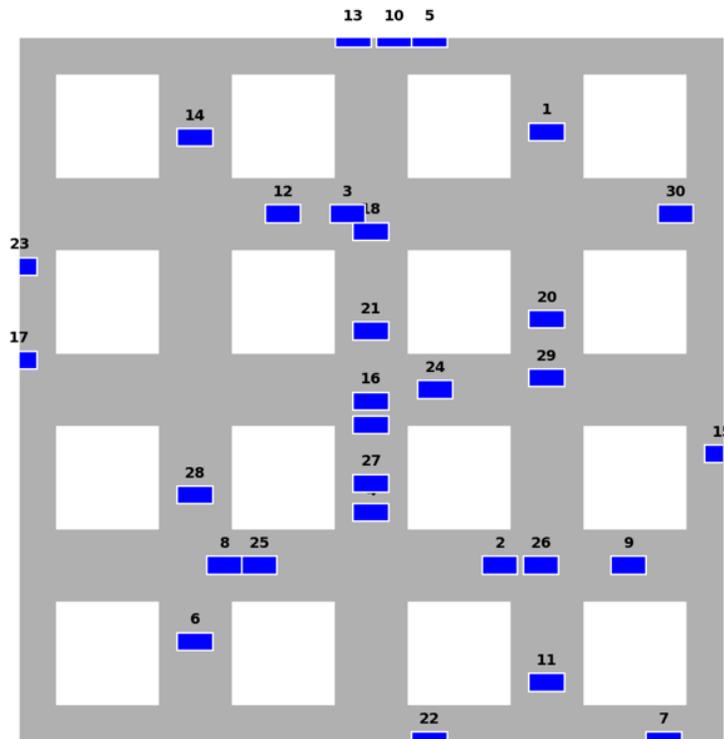


Figure 4. Urban Road With 30 Vehicles (30 node)

50-Node Scenario (High Density)

This condition resembles urban traffic congestion during peak hours, where the network is highly dense and complex with overlapping communication links. While higher density enhances connectivity, it also increases routing overhead. OLSR-DQN effectively addresses this challenge by adaptively updating routes in real time, which sustains higher PDR and lower delay, even though a slight increase in overhead is observed compared to the standard OLSR, as illustrated in Figure 5.

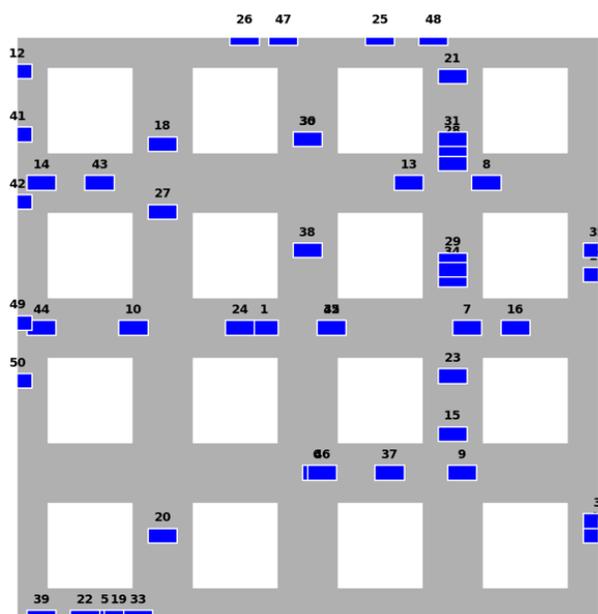


Figure 5. Urban Road With 50 Vehicles (50 node)

3.1. Packet Delivery Ratio (PDR)

The results indicate that OLSR-DQN consistently outperforms the standard OLSR protocol in terms of PDR across all network sizes (20, 30, and 50 nodes), as shown in Figure 6. At lower density (20 nodes), the improvement is modest because connectivity is more sporadic, and the availability of alternative routes is limited. However, OLSR-DQN still achieves a higher PDR by dynamically selecting stable multi-hop paths using learned policies. In the medium (30 nodes) and high-density (50 nodes) scenarios, the difference becomes significant. The reinforcement learning agent within each node adapts its forwarding decisions by considering state information such as Expected Transmission Count (ETX), queue length, and neighbor count. This enables OLSR-DQN to maintain reliable paths even when frequent topology changes occur. The findings suggest that the learning-based approach is particularly effective in dense and highly dynamic VANET environments, where conventional OLSR tends to experience link breakages and packet drops.

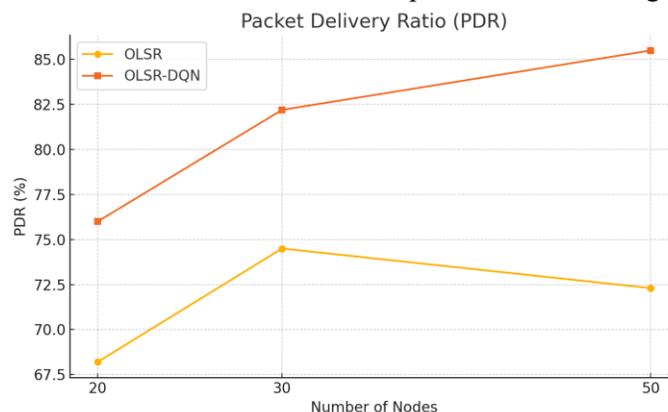


Figure 6. Packet Delivery Ratio Simulation

3.2. Average End-to-End Delay

OLSR-DQN exhibits consistently lower delay compared to OLSR, as presented in Figure 7. At 20 nodes, the delay reduction is not substantial due to fewer available routes, which sometimes forces both protocols to use longer paths. However, as the number of nodes increases, OLSR-DQN demonstrates clear advantages. The DQN agent reduces delay by avoiding unstable links and congested nodes, thereby selecting paths with higher reliability and lower retransmission rates. In contrast, standard OLSR relies heavily on periodic control messages and predefined MPR (Multi-Point Relay) selection, which cannot adapt quickly enough to dynamic topological changes. Particularly in the 50-node scenario, the delay improvement of OLSR-DQN becomes critical, as the protocol manages to keep the latency within acceptable bounds for real-time applications such as collision warning systems, whereas OLSR suffers from increasing queuing and retransmission delays.

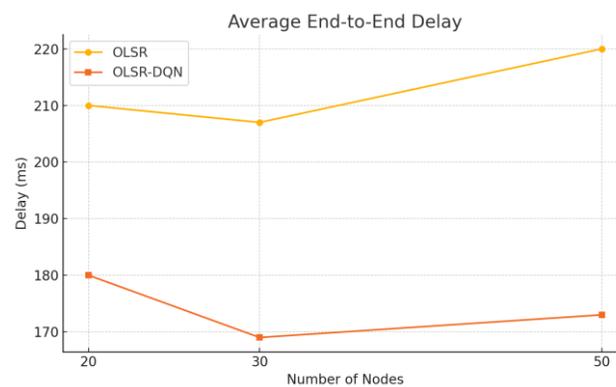


Figure 7. Average End-to-End Delay

3.3. Throughput

The throughput results confirm the advantages of OLSR-DQN, as depicted in Figure 8. The protocol achieves higher throughput across all scenarios, reflecting its ability to sustain reliable end-to-end communication under varying network densities. The improvement correlates with the higher PDR and lower delay observed in Figures 6 and 7. By distributing traffic across multiple stable routes, OLSR-DQN prevents bottlenecks and reduces packet loss due to collisions and retransmissions. This adaptive load balancing is a direct result of the reinforcement learning mechanism, which optimizes forwarding actions based on environmental feedback. In high-density networks, where throughput degradation is typically severe due to congestion and interference, OLSR-DQN demonstrates resilience by maintaining higher data delivery rates than OLSR. These results imply that OLSR-DQN can better support bandwidth-intensive VANET applications, such as video streaming or cooperative perception in autonomous driving.

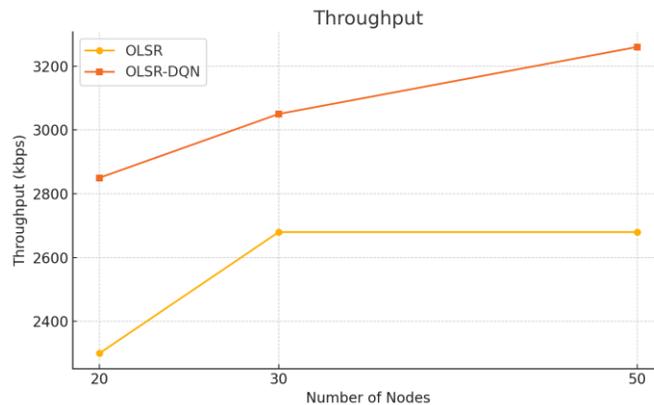


Figure 8. Throughput

3.4. Routing Overhead

While OLSR-DQN introduces a slight increase in routing overhead compared to standard OLSR, the overhead remains within an acceptable range, as shown in Figure 9. This increase stems from the additional computational and decision-making processes required by the DQN agent at each node. Nevertheless, the overhead is offset by the substantial performance gains in PDR, delay, and throughput. In other words, the additional control cost is justified, as it enables smarter and more adaptive routing behavior. Importantly, the overhead growth does not scale linearly with network size, suggesting that the DQN-based optimization maintains efficiency even as node density increases. In dense networks, the overhead trade-off is particularly worthwhile, since conventional OLSR already suffers from high control message frequency without providing corresponding performance improvements.

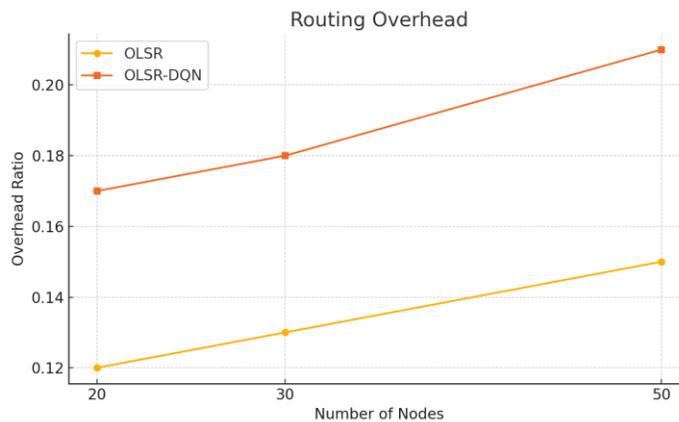


Figure 9. Routing Overhead

When the results from Figures 6–9 are considered collectively, a consistent pattern emerges: OLSR-DQN improves reliability (PDR), responsiveness (delay), and efficiency (throughput) at the cost of a manageable increase in routing overhead. The relationship between these metrics highlights the central advantage of reinforcement learning: by making routing decisions based on observed environmental states and expected future rewards, OLSR-DQN adapts dynamically to mobility and congestion, unlike the static decision-

making in OLSR. This synergy across performance indicators demonstrates that OLSR-DQN is more suitable for highly dynamic and safety-critical vehicular communication scenarios.

4. Conclusion

This research has shown that the incorporation of Deep Q-Network (DQN) into the OLSR protocol in a distributed fashion markedly improves routing efficacy in dynamic Mobile Ad-Hoc Networks (MANETs). The OLSR-DQN protocol surpasses traditional OLSR in Packet Delivery Ratio (PDR), end-to-end delay, and throughput by allowing each node to independently make decisions based on local observations using reinforcement learning. Simulation results indicate an enhancement in Packet Delivery Ratio (PDR) of up to 13%, a decrease in delay by up to 47 ms, and uniform throughput improvements across diverse node densities. Despite a minor rise in routing overhead, it remains within acceptable parameters considering the enhancements in total network performance.

This work's novelty is in its entirely decentralized, real-time learning system that aligns with the proactive characteristics of OLSR while incorporating intelligent, adaptable behavior.

These discoveries enhance the current capabilities in intelligent routing for MANETs and create opportunities for practical use in tactical military communications, disaster response, and autonomous vehicle networks. Future research topics encompass enhancing the computational efficiency of the DQN model, assessing performance in extensive and heterogeneous MANET environments, and using cross-layer network metrics to further improve routing decisions.

5. References

- [1] Syah K, Mayya K, Makarim IF, Yudi Z, Darva A. Implementasi Metode Convolutional Neural Network untuk Memilah Foto Pemandangan Pantai dan Pegunungan. 2024;2(2):62–72.
- [2] D PK, Sandhya E, Sk KS, Mantena SV, Desanamukula VS, Koteswararao C, et al. Enhancing security and efficiency in Mobile Ad Hoc Networks using a hybrid deep learning model for flooding attack detection. 2025;1–16.
- [3] El-shafai W, Member S. AI-Driven Ensemble Classifier for Jamming Attack Detection in VANETs to Enhance Security in Smart Cities. IEEE Access. 2025;13(February):50687–713.
- [4] Jamal M, Din N, Zafri M, Ma M, Mohammed H, Al-shareeda MA, et al. Routing protocols strategies for flying Ad-Hoc network (FANET): Review , taxonomy , and open research issues. Alexandria Eng J [Internet]. 2024;109(September):553–77. Available from: <https://doi.org/10.1016/j.aej.2024.09.032>
- [5] Ramalingam R, Muniyan R, Dumka A, Singh DP, Mohamed HG, Singh R, et al. Routing Protocol for MANET Based on QoS-Aware Service Composition with Dynamic Secured Broker Selection. 2022;
- [6] Danilchenko K, Azoulay R, Reches S. Deep Learning for MANET routing. IEEE Trans Mach Learn Commun Netw. 2023;PP:1.
- [7] Jala A, Segara T, Wijayanto A. Optimization of Mobile Ad-Hoc Network using Particle Swarm Optimization. *International Journal of Computer Networks and Communications*. 2021;13(5):468–74. doi:10.5121/ijcnc.2021.13503.
- [8] Nourildean SW, Hassib MD. e-Prime - Advances in Electrical Engineering , Electronics and Energy IoT-based MANET performance improvement against jamming attackers in different mobile applications. e-Prime - Adv Electr Eng Electron Energy [Internet]. 2024;8(July 2023):100615. Available from: <https://doi.org/10.1016/j.prime.2024.100615>
- [9] Khalid K, Woungang I, Dhurandher SK, Singh J. Internet of Things Reinforcement learning-based fuzzy geocast routing protocol for opportunistic networks. Internet of Things [Internet]. 2021;14:100384. Available from: <https://doi.org/10.1016/j.iot.2021.100384>

- [10] Kaviani S, Ryu B, Ahmed E, Larson K, Le A, Yahja A, et al. DeepCQ + : Robust and Scalable Routing with Multi-Agent Deep Reinforcement Learning for Highly Dynamic Networks. 2021;(January 2022).
- [11] Bohra N, Kumari A, Mishra VK, Soni PK. Intelligence-Based Strategies with Vehicle-to-Everything Network : A Review. 2025;1–40.
- [12] Vishwanathrao BA, Vikhar PA. INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Reinforcement Machine Learning-based Improved Protocol for Energy Efficiency on Mobile Ad-Hoc Networks. 2024;12:654–70.
- [13] Ahmed T, Khan I, Rauf A. Secure and efficient communication protocols for tactical MANETs: A survey. *Computer Standards & Interfaces*. 2023;86:103706. doi:10.1016/j.csi.2023.103706.
- [14] Jala A, Segara T, Wijayanto A, Gustalika MA, Ramadhani AD. Implementasi Mobile Ad-Hoc Network Pada Daerah Pasca Bencana Dengan Protokol DSR. 2022;9(4):834–9.
- [15] Zhao F, Chen X, Li J. Performance analysis of TCP and UDP traffic over AODV protocol in naval MANET communications. *Journal of Communications and Networks*. 2023;25(1):53–8. doi:10.1109/JCN.2023.000007.