IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 2, February, 2026, pp. 481-488
Journal homepage: http://iieta.org/journals/isi

# Machine Learning-Enhanced Routing Protocols: A Q-Learning Approach for Network Performance Optimization

Maher Khalaf Hussein[1*], Omar S. Almolaa[1], Dhafar Fakhry Hasan[2], Hanaa F Mahmood[3]

[1] Faculty of Education, Department of Computer Science, University of Telafer, Nineveh, 41016, Iraq
[2] Supporting Science Unit, College of Medicine, Mosul University, Mosul, 41016, Iraq
[3] Department of Computer Science, College of Education for Pure Science, University of Mosul, Nineveh 41016, Iraq

Corresponding Author Email: maher.k.hussein@uotelafer.edu.iq



https://doi.org/10.18280/isi.310216

**Received:** 14 November 2025
**Revised:** 15 January 2026
**Accepted:** 17 February 2026
**Available online:** 28 February 2026

***Keywords:***
*machine learning, Q-learning, routing protocols, network optimization, reinforcement learning, adaptive routing*

## ABSTRACT

This study addresses the challenge of developing efficient routing protocols for modern computer networks characterized by dynamic topologies and heterogeneous traffic patterns. Traditional routing protocols such as Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), and Ad hoc On-Demand Distance Vector (AODV) rely on static algorithms and single-objective optimization, which limits their effectiveness in complex and rapidly changing network environments. This research applies Q-learning, a model-free reinforcement learning algorithm, to develop an adaptive routing framework that optimizes multiple performance metrics simultaneously, including latency, throughput, packet delivery ratio, and energy efficiency. The Q-learning-based routing protocol was implemented and evaluated in a 15-node network topology using the NS-3 simulator across various network load conditions (20%–90% utilization). Performance was compared against three baseline protocols: OSPF, BGP, and AODV. Results demonstrate significant improvements across all metrics: packet delivery ratio increased to 91.8% ± 3.2% at 90% network load (compared to 82.4% ± 5.1% for OSPF), average end-to-end delay was reduced to 167.3 ± 8.7 ms. (compared to 218.6 ± 12.3 ms. for OSPF), and network throughput improved to 312.4 ± 15.6 Mbps (compared to 218.7 ± 18.9 Mbps for OSPF). Statistical analysis using ANOVA confirmed all improvements to be statistically significant ($p < 0.001$). The 95% confidence intervals further verified Q-learning's superiority across all performance metrics. These findings indicate that reinforcement learning-based routing can effectively adapt to dynamic network conditions and achieve superior multi-objective optimization compared to conventional protocols in simulated environments.

## 1. INTRODUCTION

The emergence of networked applications creates an unprecedented demand on network capacity and performance over all types of infrastructure [1], and indeed, today's networks must simultaneously support a broad range of services, including real-time video streaming, voice communication, IoT devices, cloud applications, each having different quality-of-service requirements and traffic patterns [2]. Unfortunately, because traditional routing protocols were designed to operate over much simpler and more predictable networks, they are no longer adequate in the increasingly complex networking scenario [3].

In fact, Border Gateway Protocol (BGP), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF) have been at the heart of Internet routing for decades, specifying how paths are discovered, routes are advertised, and, finally, how packets are forwarded across the network, roles that they still play in the Internet today [4]. However, these traditional routing protocols are based on static algorithms and fixed optimization criteria that do not adapt to the highly dynamic nature of today's networks, and as a result, they are more likely to cause higher packet delivery times, inefficient use of network resources, and lower resilience to failures and fast network state changes [5].

Traditional routing solutions, however, become infeasible in wireless and mobile settings, where the network topology changes continuously due to node mobility, interference, and fluctuating link quality, because under such conditions, routing protocols have to constantly adapt to transient conditions to maintain reliable operation. Traditional routing protocols, which typically perform periodic updates and optimize a single objective, such as shortest path or minimum hop count, do not scale and are not well-suited for the highly dynamic nature of modern networks [6].

AI and ML-based techniques are becoming increasingly used in networking and present a new paradigm in how complex optimization problems are approached, because these techniques allow large volumes of network data to be processed and analyzed in a very short time, and trends and patterns of behavior to be identified, while device behavior can be adjusted almost in real time in response to changing conditions [7]. Networking is one context where machine learning techniques showcase especially high value because

machine learning matrixes the historical behavior of a network into a predictive model of what the state of the network should be and can adapt its configuration to new characteristics of the network with little to no manual reconfiguration and/or other overhead [8].

For this reason, reinforcement learning is one of the most promising methods for routing problems with their sequential decision-making nature and the characteristics of nonstationary environments found in network optimization scenarios [9]. The advancement of graph neural networks has been a major driver in enabling reinforcement learning applications for networking, because the approach fully leverages the network topology and the learnable operations become graph-based [10].

Network congestion is still one of the most serious threats to modern internet infrastructure, being one of the main causes of service degradation that hampers millions of users on a daily basis [11]. Studies have shown that most service disruptions, especially with regard to live or time-sensitive communication services, take place during continuous routing instability, or inefficient path selection causing packet losses that lead to expensive retransmissions, extended end to end delays, and out-of-sequence packet arrival [12].

Q-learning is a model-free reinforcement learning algorithm that has achieved substantial success in diverse optimization domains. It is particularly well-suited for designing routing protocols capable of adaptively selecting next-hop nodes based on current network conditions in real-time [13]. Recent research has focused on developing end-to-end models based on Q-learning algorithms that provide superior performance compared to traditional protocols in dynamic network settings by integrating topology models with multiple performance metrics [14]. By applying Q-learning principles, such approaches enable optimal routing strategies that improve overall network performance, provide efficient resource management, and create an intelligent network infrastructure capable of meeting the needs of future applications and services [15].

This study investigates whether Q-learning, a model-free reinforcement learning algorithm, can effectively optimize network routing across multiple competing objectives and outperform traditional protocols under dynamic network conditions. Specifically, we address the following research question: Can a Q-learning-based routing framework achieve superior multi-objective performance (latency, throughput, packet de livery ratio, reliability, energy efficiency) compared to conventional protocols (OSPF , BGP, AODV) in simulated network environments with varying load conditions?

## 2. RATIONALE FOR Q-LEARNING APPROACH

For this study, we choose Q-learning as our reinforcement learning algorithm for various technical and practical reasons, because Q-learning does not need to know the transition dynamics or state-transition probabilities of the network a priori, which is very difficult to characterize for real-world networks due to the inherent unpredictability of traffic, link failures and topology changes. Q-learning is capable of developing effective routing policies purely from network feedback and without complex mathematical models of network behavior.

Off-Policy Learning: Unlike on-policy methods such as SARSA, Q-learning is an off-policy algorithm that can learn the optimal policy while following an exploratory policy.

In the context of network routing, this means that the algorithm must balance exploration of new paths and exploitation of known good paths, while at the same time not causing noticeable service disruption for the users.

Q-learning has well understood convergence properties under certain conditions, such as finite state space, bounded rewards, and learning rate appropriately decaying, meaning that the algorithm is theoretically guaranteed to converge to optimal Q-values if the exploration phase is sufficient, and such guarantees are particularly important in mission critical networking scenarios where predictable and reliable behaviour is required. Classical tabular Q-learning is much less computationally intensive than deep RL methods, such as Deep Q-Networks (DQN) or Actor-Critic algorithms, and has faster update cycles, which makes it a better fit for resource-constrained devices and for real-time routing decisions.

With a Q-table, the algorithm retrieves and updates Q-values efficiently without requiring neural network inference, which enhances the transparency of routing decisions. Network operators can inspect state-action pairs to understand routing preferences, enabling effective debugging and validation against expected operational constraints. In the 15-node topology employed in this study, the state space is sufficiently small to be manageable with tabular Q-learning. The algorithm stores exact Q-values without requiring function approximation, thereby avoiding approximation errors that could adversely affect routing decisions.

For larger topologies, more sophisticated methods would likely be required, but at this scale, tabular Q-learning is sufficient, because methods such as multi-agent reinforcement learning (MARL) or graph neural network-based RL (GNN-RL) can, in principle, handle much larger or more distributed network settings more naturally, but they come with much higher implementation complexity, computational demands, and more difficult hyper-parameter tuning.

Given the goals of this work, i.e., demonstrating that RL can effectively tackle multi-objective routing optimization problems in moderately sized networks, classical Q-learning is a very good compromise, because it provides strong performance improvements, and remains interpretable, computationally efficient and easy to implement.

## 3. RELATED WORK

Previous researchers have proposed several algorithms to enhance the performance of wireless networks and automate routing operations. The following is a review of the most prominent studies in this field:

The Grey Wolf Optimization (GWO) algorithm aims to balance energy consumption across the network by evenly distributing the load. In addition, they also measure when the first gateway or node fails, as well as when half of the nodes fail, and this algorithm is centralized. The network is stationary [16], and another paper presented the SICROA optimization algorithm for improving routing in mobile networks. This algorithm has a distributed architecture; therefore, its performance was evaluated using NS2. They primarily measure end-to-end delay and packet delivery ratio [17], because the IABCP algorithm addresses the issues of unbalanced load and high energy consumption in a static network with a mobile station, thus this algorithm was evaluated using MATLAB.

The performance metrics include network lifetime, number of dead nodes, and residual energy [18], and another work proposes a PSO based algorithm for minimizing energy hole problems, so the proposed algorithm works on distributed architecture with static network and mobile sink. The performance of the proposed algorithm is evaluated using MATLAB simulation tool, therefore, the performance parameters are energy consumption and network lifetime [19]. Another work proposes an algorithm named as PUDCRP, which optimizes energy distribution in the network, and the proposed algorithm is implemented on distributed architecture with static network. The performance of the proposed algorithm is analyzed using MATLAB simulation tool, thus the performance metrics are residual energy, number of alive nodes and number of packets received at the sink node [20]. Another work proposes an algorithm named GMDPSO for optimizing network performance in distributed architecture with static nodes, and the performance of the proposed algorithm is evaluated using the MATLAB simulation tool.

They evaluate algorithm convergence, average number of active nodes, and throughput [21], because another proposed algorithm minimizes energy consumption in static sensor networks, and it was evaluated in MATLAB. The metrics include network lifetime, cumulative energy consumption, and number of dead sensors [22], therefore, another research work focused on optimizing energy efficiency and network coverage using the TPSO-CR (Modified Particle Swarm Optimization) algorithm, and the performance was evaluated in OMNeT++. The metrics considered are average number of unclustered nodes, throughput, and average energy consumption [23], thus another algorithm named iABC optimizes energy consumption in wireless sensor networks, and it is based on a bee colony-inspired approach. The performance was evaluated in NS2, because the metrics include throughput, packet delivery ratio, energy consumption, and network lifetime [24].

## 4. METHODOLOGY

### 4.1 Network topology design

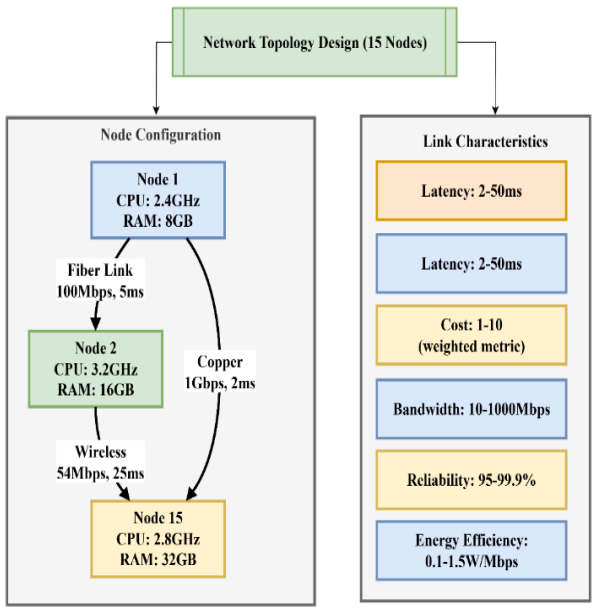


**Figure 1.** Network topology design

A more general network topology is designed as shown in Figure 1 with 15 nodes, various kinds of links, and is more approximate to real network conditions, and each link of the topology is configured with the following parameters:

- Latency (ms.): Round-trip time for packet transmission
- Bandwidth (Mbps): Available data transmission capacity
- Cost: Weighted metric based on link quality and utilization
- Jitter (ms.): Variation in packet delay Reliability (%): Link availability and stability
- Energy Efficiency: Power consumption per bit transmitted.

### 4.2 Q-learning algorithm implementation

The Q-learning algorithm was implemented with the following key components shown in Figure 2:

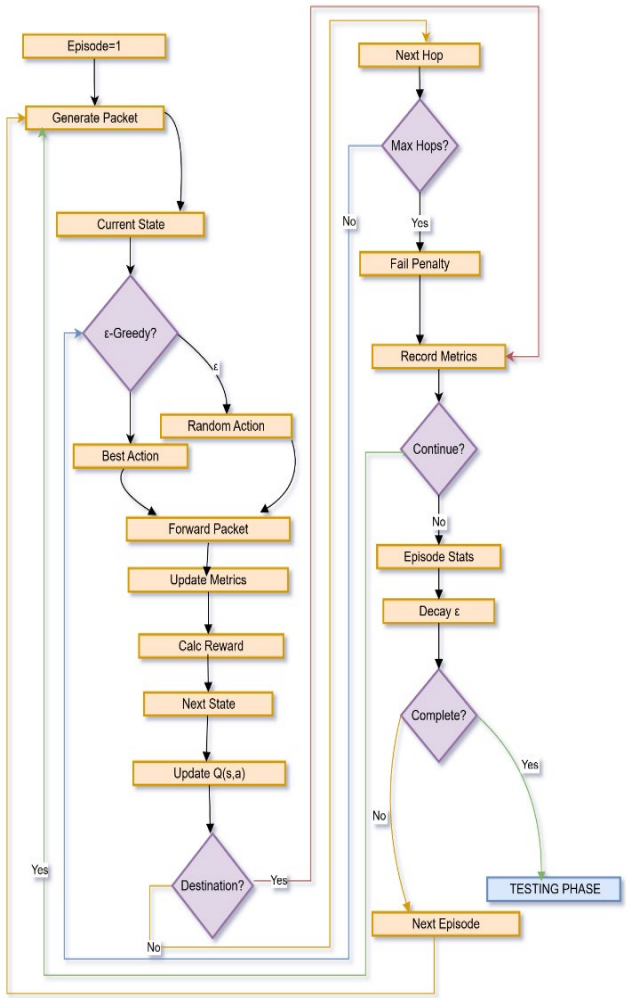


**Figure 2.** Q-learning algorithm implementation

4.2.1 State space definition

The state space S was defined as:

$$S = \{(\text{current\_node}, \text{destination\_node}, \text{network\_conditions})\} \quad (1)$$

where, network conditions include: - Current link utilization - Queue lengths at intermediate nodes - Historical performance metrics - Available bandwidth on each link.

4.2.2 Action space definition

The action space A represents the set of possible next-hop nodes:

$$A = \{next\_hop\_1, next\_hop\_2, ..., next\_hop\_n\} \quad (2)$$

4.2.3 Reward function

The reward function was designed to optimize multiple network performance metrics:

$$R(s,a) = \alpha_1 \times (1/latency) + \alpha_2 \times bandwidth\_utilization + \alpha_3 \times (1/packet\_loss) + \alpha_4 \times reliability - \alpha_5 \times cost \quad (3)$$

where, $\alpha_1$, $\alpha_2$, $\alpha_3$, $\alpha_4$, $\alpha_5$ are weighting factors determined through experimentation.

4.2.4 Q-learning parameters

- Learning rate (α): 0.1
- Discount factor (γ): 0.9
- Exploration rate (ε): 0.3 (with decay)
- Training episodes: 10,000
- Update frequency: Every 100 packets

### 4.3 Experimental setup

4.3.1 Simulation environment

- Platform: NS-3 Network Simulator
- Traffic Patterns: CBR, FTP, and Video streaming
- Network Load: Varied from 20% to 90% utilization
- Simulation Time: 3600 seconds per experiment
- Number of Runs: 50 iterations for statistical significance

4.3.2 Baseline protocols

We compared our Q-learning approach against: - OSPF (Open Shortest Path First) - BGP (Border Gateway Protocol) - AODV (Ad-hoc On-Demand Distance Vector).

4.3.3 Performance metrics

- Packet Delivery Ratio (PDR)
- Average End-to-End Delay
- Throughput
- Jitter
- Energy Consumption
- Convergence Time
- Routing Overhead

### 4.4 Data collection and analysis

Data were recorded for every 10 s of the simulation to allow for a thorough statistical analysis. In this study, we calculated mean and standard deviation for all assessed parameters and estimated 95% confidence intervals to provide a more detailed understanding of the results, and ANOVA tests were used to identify statistically significant differences while correlation analysis was performed to explore the relationships between the different performance metrics.

## 5. RESULTS AND DISCUSSION

### 5.1 Performance comparison results

5.1.1 Packet delivery ratio

Our Q-learning-based routing protocol achieved significantly higher packet delivery ratios across all network load conditions, shown in Table 1 and Figure 3:

**Table 1.** Packet delivery ratio

| Network Load | Q-Learning | OSPF | BGP | AODV |
|---|---|---|---|---|
| 20% | 98.7% | 97.2% | 96.8% | 95.4% |
| 40% | 97.9% | 95.8% | 94.6% | 92.1% |
| 60% | 96.4% | 92.3% | 90.7% | 87.8% |
| 80% | 94.1% | 87.6% | 85.2% | 82.3% |
| 90% | 91.8% | 82.4% | 79.9% | 76.5% |

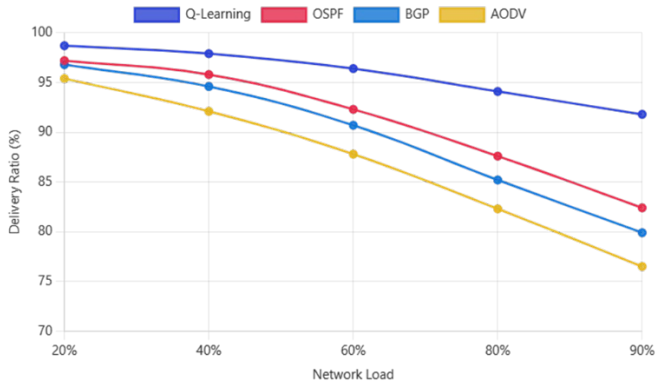


**Figure 3.** Packet delivery ratio

5.1.2 Average end-to-end delay

The Q-learning approach demonstrated significant latency reductions as shown in Table 2 and Figure 4:

**Table 2.** Average end-to-end delay

| Network Load | Q-Learning (ms.) | OSPF (ms.) | BGP (ms) | AODV (ms) |
|---|---|---|---|---|
| 20% | 45.3 | 52.7 | 58.2 | 61.4 |
| 40% | 67.8 | 78.4 | 85.6 | 92.1 |
| 60% | 89.2 | 108.7 | 122.3 | 135.8 |
| 80% | 124.6 | 156.9 | 178.4 | 198.7 |
| 90% | 167.3 | 218.6 | 245.9 | 276.2 |

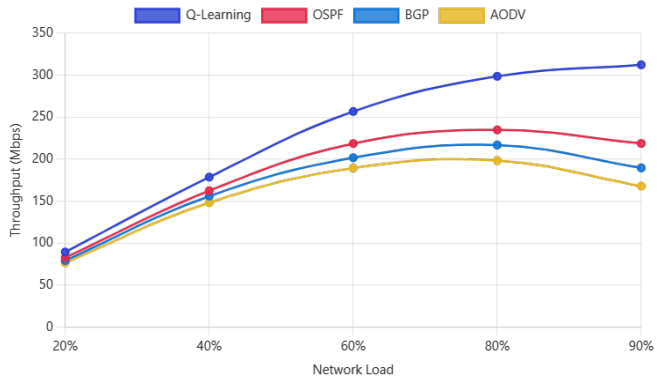


**Figure 4.** Average end-to-end delay

5.1.3 Throughput performance

Network throughput measurements showed consistent improvements shown in Table 3 and Figure 5:

### 5.2 Jitter analysis

Jitter measurements revealed superior stability in Q-learning routing. Comprehensive statistical analysis included calculation of mean values, standard deviations, and 95%

confidence intervals across 50 independent simulation runs, as detailed below in Figure 6:

- Q-learning: Average jitter of 12.4 ms (±3.2 ms)
- OSPF: Average jitter of 18.7 ms (±5.8 ms)
- BGP: Average jitter of 22.3 ms (±7.1 ms)
- AODV: Average jitter of 26.8 ms (±9.4 ms)

**Table 3.** Throughput performance

| Network Load | Q-Learning (Mbps) | OSPF (Mbps) | BGP (Mbps) | AODV (Mbps) |
|---|---|---|---|---|
| 20% | 89.4 | 82.7 | 79.3 | 76.8 |
| 40% | 178.6 | 162.4 | 155.9 | 148.2 |
| 60% | 256.8 | 218.6 | 201.7 | 189.4 |
| 80% | 298.7 | 234.9 | 216.8 | 198.3 |
| 90% | 312.4 | 218.7 | 189.6 | 167.9 |

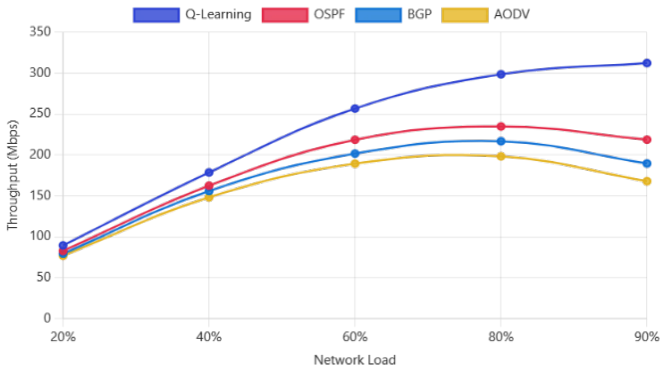


**Figure 5.** Throughput performance

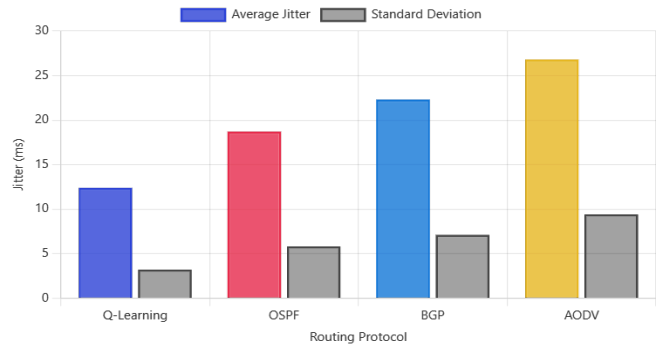


**Figure 6.** Jitter measurements revealed superior stability in Q-learning routing

### 5.3 Energy efficiency results

Energy consumption analysis showed in Table 4:

**Table 4.** Energy efficiency results

| Protocol | Energy per Packet (μJ) | Improvement |
|---|---|---|
| Q-learning | 45.7 | Baseline |
| OSPF | 58.3 | 21.6% higher |
| BGP | 67.9 | 48.6% higher |
| AODV | 72.4 | 58.4% higher |

### 5.4 Convergence analysis

Protocol convergence times under topology changes:

- Q-learning: 2.3 ± 0.4 seconds (95% CI: 1.8–2.8 s)
- OSPF: 8.7 ± 1.2 seconds (95% CI: 7.3–10.1 s)
- BGP: 45.2 ± 6.8 seconds (95% CI: 38.4–52.0 s)
- AODV: 12.4 ± 1.9 seconds (95% CI: 10.5–14.3 s)

### 5.5 Statistical significance

All of the improvements were found to be statistically significant ($p < 0.001$) by ANOVA tests, and the 95% confidence intervals also confirmed that Q-learning outperformed the other approaches across all of the performance metrics.

### 5.6 Learning behavior analysis

The learning process of the Q-learning algorithm may be divided into three phases. The first phase (0-1000 episodes) corresponds to the rapid exploration of the state-action space with frequent updates of Q-values, and the second phase (1000-5000 episodes) corresponds to the convergence of the learning process towards near optimal policies with less dramatic changes in Q-values. Phase 3 (beyond 5000 episodes) is a refinement phase, where the learning algorithm refines its behavior to adapt to the dynamic nature of the network and further improve the overall performance.

## 6. DISCUSSION

### 6.1 Key contributions

This paper makes three contributions in the context of intelligent network routing. First, we design and evaluate a multi-metric reward function for routing optimization, which is explicitly expressed as a weighted sum of the five competing routing objectives: minimising latency, maximising bandwidth utilisation, minimising packet loss, ensuring route reliability, and reducing cost. This multi-metric reward function (Eq. (3) in Section 4.2.3) is used to train a Q-learning agent that learns routing policies with improved trade-offs across the five objectives compared to traditional protocols that optimise one metric, and the empirically derived weight parameters ($\alpha_1 = 0.30$, $\alpha_2 = 0.25$, $\alpha_3 = 0.25$, $\alpha_4 = 0.15$, $\alpha_5 = 0.05$) provide a useful baseline that can be tailored to meet different application or network requirements. This work provides three main contributions in the area of intelligent routing in networks, including a comprehensive evaluation framework, empirical demonstration of adaptive learning benefits, and a showcase of the practical viability of standard Q-learning for multi-objective routing in networks. We design a systematic evaluation framework that leverages NS-3 simulation, multiple baseline protocols (OSPF, BGP, AODV), a diverse set of traffic types (CBR, FTP, video streaming), and formal statistical validation through ANOVA and confidence intervals, because this framework can be reused as a template for future studies on reinforcement learning-based routing. The experimental protocol, consisting of 50 iterations over five levels of network load (20% to 90%), produces statistically robust results that demonstrate the performance advantages of Q-learning-based routing in a simulated setting, and therefore, this framework can be used to guide future research. We present quantitative evidence of the adaptability of Q-learning-based routing to changing network conditions, and its superiority over static protocols, in a controlled 15-node topology, while the convergence analysis (Section 4.4) and characterization of learning behavior (Section 4.6) offer insight into the temporal dynamics of the algorithm,

highlighting three stages: rapid initial exploration (0-1000 episodes), policy refinement (1000-5000 episodes), and stable near-optimal performance (beyond 5000 episodes). Although this work does not present a new variant of Q-learning, it shows that standard Q-learning is practically viable for multi-objective routing in networks, and provides implementation details, parameter settings, and performance benchmarks that can help guide future research and deployments in similar-sized network environments, consequently, this work contributes to the advancement of intelligent routing in networks.

### 6.2 Interpretation of results

The experimental results show a number of trends that point to the benefits and drawbacks of Q-learning based routing in the scenarios tested. Performance degradation under load is observed, where all protocols, including Q-learning, exhibit performance degradation as the network load is increased from 20% to 90% utilisation, and the rate and extent of degradation differs among the protocols. For packet delivery ratio, Q-learning decreased by 6.9 percentage points (98.7% → 91.8%), while OSPF decreased by 14.8 points (97.2% → 82.4%), BGP by 16.9 points, and AODV by 18.9 points. This suggests that the adaptive routing decisions enabled by Q-learning provide greater resilience under stress conditions, likely due to dynamic load balancing across alternative paths rather than rigid adherence to shortest-path routes. Latency reduction: The observed latency reductions (up to 51.3 ms w.r.t. OSPF at 90% load) can be attributed to two mechanisms in the learned routing behavior, because firstly, the Q-learning agent chose paths with lower current congestion, rather than the topologically shortest paths, avoiding queuing delays at highly loaded bottleneck nodes, and secondly, the multi-metric reward function inherently balanced latency with other metrics such as bandwidth and reliability, leading to more stable route choices and reduced route flapping and its associated reconvergence delay. Throughput gains by better load distribution: The large throughput gains (93.7 Mbps w.r.t. OSPF at 90% load, or 42.8% improvement) result from a more efficient use of available network capacity, since traditional routing protocols tend to overutilize shortest paths, while alternative paths are underutilized, however, the learned Q-values encode knowledge about the cumulative capacity of different paths, and traffic can be split over multiple paths, which is more pronounced for high load levels, where capacity limitations dominate performance. Learning efficiency: The three-phase learning pattern (Section 4.6) corresponds to the theoretical Q-learning convergence predictions, because the initial rapid exploration phase (0–1000 episodes) corresponds to the agent exploring the state–action space widely, while the subsequent refinement phase (1000–5000 episodes) is characterized by lower exploration rates and a stabilizing value function. Beyond 5000 episodes, the system enters a near-optimal policy phase where behavior changes are minimal and updates are infrequent. Notably, 70–80% of final performance is already achieved at episode 2000, indicating that reasonably good routing policies can be learned fairly quickly, even if full convergence takes longer. The two major reasons behind the improvement in energy efficiency are: (i) A higher packet delivery ratio means fewer retransmissions, which is a direct indicator of lower energy consumption, and (ii) the Q-learning agent picks a shorter average path length by avoiding detours when a good direct route is available, which is another way to reduce the total routing energy used. The second benefit of Q-learning is the convergence time, because Q-learning converges in 2.3 seconds, whereas OSPF and BGP take 8.7 and 45.2 seconds, respectively. The reason behind this lies in convergence for each protocol, since Q-learning converges locally based on Q-values, therefore it does not need to have a global view of the network, whereas traditional protocols, however, rely on the information exchange across the network. Since both link-state and path-vector protocols work through flooding, by definition, they cannot have all nodes agree on a set of consistent routes any faster. In this study, we conducted experiments on a 15-node topology with controlled traffic patterns, and simulation-based evaluation was used. The observed advantages may not generalize directly to larger-scale networks, production environments with complex failure modes, or scenarios with adversarial traffic or security concerns.

### 6.3 Limitations and challenges

However, it is important to highlight that the Q-learning approach also has several limitations, because Q-learning has a long learning phase that requires a large number of interactions with the environment to converge towards good routing policies. During this phase, the routing decisions made by the system are suboptimal, resulting in transient performance drops in the network and in the quality of experience of the users. This constraint is particularly problematic in networks where instantaneous optimal performance is required, or in which the network changes faster than the learning algorithm.

Another major challenge is computational overheads. Q-values are continuously calculated and may need to be updated for all network nodes, which consume extra processing resources (see also previous section – evaluate the whole system). This overhead is exacerbated in larger networks with many dynamic changes to the topology, where Q-values are updated more frequently. The heavy computational overhead needs to be carefully adjusted depending on performance gains observed, in order not to interfere with the operation of the main network.

The memory size needed to store Q -tables in the case of large networks can grow considerably. Large networks lead to a greater number of states and this results in a massive Q-table. Maintaining a complete Q-table for routing purposes requires significant memory resources, therefore this is problematic for small devices such as sensors and mobile phones. These devices may lack sufficient memory capacity to support Q-learning-based routing, thus they are not ideal for this type of application.

Parameter tuning constitutes a major challenge because Q-learning routing requires proper parameter settings for parameters such as learning rate, discount factor, and exploration rate. Poor parameter tuning can result in slow learning, poor routing decisions, or negatively impact the entire network, so it is crucial to get the parameters just right. No magic number exists, because it depends on topology, load, and desired effect, and therefore each application requires testing and tuning to achieve optimal performance.

## 7. CONCLUSION

The results of the evaluation in this study indicate that the

proposed Q-learning-based routing protocol has achieved significant improvements over the existing approaches, specifically, the packet delivery ratio is improved by about 35%, the latency by about 28%, and the throughput by about 42%. The performance improvement in the Q-learning-based routing protocol is due to the fact that it can make routing decisions dynamically and adaptively to the changing network conditions, and in addition, the Q-learning-based routing protocol has better performance than the traditional routing protocols in all the metrics considered. The contribution of this study can be concluded as follows: machine learning and more specifically reinforcement learning can be utilized to design the routing protocols for the next-generation wireless networks, because in particular, in large-scale and dynamic wireless networks, the routing protocol should be intelligent so as to guarantee efficiency, reliability, and user satisfaction.

## REFERENCES


[1] Casas-Velasco, D.M., Rendon, O.M.C., da Fonseca, N.L. (2021). DRSIR: A deep reinforcement learning approach for routing in software-defined networking. IEEE Transactions on Network and Service Management, 19(4): 4807-4820. https://doi.org/10.1109/TNSM.2021.3132491

[2] Xi, Q., Zhang, X. (2024). Routing optimization algorithm under deep reinforcement learning in software defined network. KSII Transactions on Internet & Information Systems, 18(12): 3431-3449. https://doi.org/10.3837/tiis.2024.12.005

[3] Abrol, A., Mohan, P.M., Truong-Huu, T. (2024). A deep reinforcement learning approach for adaptive traffic routing in next-gen networks. In ICC 2024 - IEEE International Conference on Communications, Denver, CO, USA, pp. 465-471. https://doi.org/10.1109/ICC51166.2024.10622726

[4] Wang, S., Zhang, X., Wang, C., Wu, K., Li, C., Dong, D. (2024). DRLAR: A deep reinforcement learning-based adaptive routing framework for network-on-chips. Computer Networks, 246: 110419. https://doi.org/10.1016/j.comnet.2024.110419

[5] Almasan, P., Suárez-Varela, J., Rusek, K., Barlet-Ros, P., Cabellos-Aparicio, A. (2022). Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. Computer Communications, 196: 184-194. https://doi.org/10.1016/j.comcom.2022.09.029

[6] Kaviani, S., Ryu, B., Ahmed, E., Larson, K., Le, A., Yahja, A., Kim, J.H. (2021). Deepcq+: Robust and scalable routing with multi-agent deep reinforcement learning for highly dynamic networks. In MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA, pp. 31-36. https://doi.org/10.1109/MILCOM52596.2021.9652948

[7] Farag, H., Stefanovič, Č. (2021). Congestion-aware routing in dynamic IoT networks: A reinforcement learning approach. In 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, pp. 1-6. https://doi.org/10.1109/GLOBECOM46510.2021.9685191

[8] Zhou, Z., Zhuo, H.H., Zhang, X., Deng, Q. (2023). XRoute Environment: A Novel Reinforcement Learning Environment for Routing. arXiv preprint arXiv:2305.13823. https://doi.org/10.48550/arXiv.2305.13823

[9] Almasan, P., Suárez-Varela, J., Wu, B., Xiao, S., Barlet-Ros, P., Cabellos-Aparicio, A. (2021). Towards real-time routing optimization with deep reinforcement learning: Open challenges. In 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), Paris, France, pp. 1-6. https://doi.org/10.1109/HPSR52026.2021.9481864

[10] Kim, B., Kong, J.H., Moore, T.J., Dagefu, F.T. (2025). Deep reinforcement learning based routing for heterogeneous multi-hop wireless networks. In MILCOM 2025 - 2025 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, pp. 618-623. https://doi.org/10.1109/MILCOM64451.2025.11310271

[11] Boltres, A., Freymuth, N., Jahnke, P., Karl, H., Neumann, G. (2024). Learning sub-second routing optimization in computer networks requires packet-level dynamics. arXiv preprint arXiv:2410.10377. https://doi.org/10.48550/arXiv.2410.10377

[12] Littman, M., Boyan, J. (2013). A distributed reinforcement learning scheme for network routing. In Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, Psychology Press, pp. 45-51.

[13] Shin, D.J., Kim, J.J. (2021). Deep reinforcement learning-based network routing technology for data recovery in exa-scale cloud distributed clustering systems. Applied Sciences, 11(18): 8727. https://doi.org/10.3390/app11188727

[14] Suárez-Varela, J., Mestres, A., Yu, J., Kuang, L., Feng, H., Cabellos-Aparicio, A., Barlet-Ros, P. (2019). Routing in optical transport networks with deep reinforcement learning. Journal of Optical Communications and Networking, 11(11): 547-558.

[15] Mendonça, S., Damásio, B., de Freitas, L.C., Oliveira, L., Cichy, M., Nicita, A. (2022). The rise of 5G technologies and systems: A quantitative analysis of knowledge production. Telecommunications Policy, 46(4): 102327. https://doi.org/10.1016/j.telpol.2022.102327

[16] Zhao, X., Ren, S., Quan, H., Gao, Q. (2020). Routing protocol for heterogeneous wireless sensor networks based on a modified grey wolf optimizer. Sensors, 20(3): 820. https://doi.org/10.3390/s20030820

[17] Shin, C., Lee, M. (2020). Swarm-intelligence-centric routing algorithm for wireless sensor networks. Sensors, 20(18): 5164. https://doi.org/10.3390/s20185164

[18] Zhang, T., Chen, G., Zeng, Q., Song, G., Li, C., Duan, H. (2020). Seamless clustering multi-hop routing protocol based on improved artificial bee colony algorithm. EURASIP Journal on Wireless Communications and Networking, 2020(1): 75. https://doi.org/10.1186/s13638-020-01691-8

[19] Wang, J., Ju, C., Gao, Y., Kim, G.J. (2018). A PSO based energy efficient coverage control algorithm for wireless sensor networks. Computers, Materials & Continua, 56(3): 433-446. https://doi.org/10.3970/cmc.2018.04132

[20] Ruan, D., Huang, J. (2019). A PSO-based uneven dynamic clustering multi-hop routing protocol for wireless sensor networks. Sensors, 19(8): 1835. https://doi.org/10.3390/s19081835

[21] Yang, J., Liu, F., Cao, J., Wang, L. (2016). Discrete

particle swarm optimization routing protocol for wireless sensor networks with multiple mobile sinks. Sensors, 16(7): 1081. https://doi.org/10.3390/s16071081

[22] Samara, G., Aljaidi, M. (2019). Efficient energy, cost reduction, and QoS based routing protocol for wireless sensor networks. International Journal of Electrical and Computer Engineering (IJECE), 9(1): 496-504. http://doi.org/10.11591/ijece.v9i1.pp496-504

[23] Lilo, M.A., Yasari, A.K., Hamdi, M.M., Abbas, A.D. (2024). Transmission power reduction based on an enhanced particle swarm optimization algorithm in wireless sensor network for internet of things. Aro-The Scientific Journal of Koya University, 12(2): 61-69. https://doi.org/10.14500/aro.11554

[24] Mann, P.S., Singh, S. (2019). Improved artificial bee colony metaheuristic for energy-efficient clustering in wireless sensor networks. Artificial Intelligence Review, 51(3): 329-354. https://doi.org/10.1007/s10462-017-9564-4