Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

ML-Driven Context-Aware Adaptive Routing for Multi-Objective Optimization in Software-Defined Networks

N. SenthilKumaran[1], Dr. R. Sankarasubramanian[2]
[1]Assistant Professor, Vellalar College for Women (Autonomous), Erode, India
[2]Principal, Erode Arts and Science College, Erode, India
Corresponding Author: N. SenthilKumaran
Email: senthilkumaran@vcw.ac.in

Abstract: Software-Defined Networking (SDN) enables centralized and programmable network control, making it highly suitable for data-center and edge-cloud environments where traffic demands change rapidly. However, conventional routing methods such as shortest-path routing and Equal-Cost Multi-Path (ECMP) forwarding are often unable to respond effectively to fluctuating congestion, resulting in higher latency, uneven link utilization, and reduced fault tolerance. This paper proposes an ML-driven context-aware adaptive routing framework for SDN using multi-objective reinforcement learning to improve routing decisions in real time. The proposed system integrates a lightweight learning agent within the SDN controller and uses telemetry such as link utilization, packet loss, delay, and failure probability to select efficient routes dynamically. A multi-objective reward function is designed to jointly optimize latency, load balancing, and resilience, while maintaining low computational overhead through a compact state representation. The model is implemented in a Mininet-based SDN environment and evaluated against OSPF shortest-path routing, ECMP, and a single-objective ML-based baseline. The results indicate improvements in latency, link utilization balance, packet loss reduction, and recovery time after link failures. The framework provides a scalable and modular solution for intelligent routing in next-generation SDN infrastructures.

Keywords: Software-Defined Networking, Reinforcement Learning, Adaptive Routing, Multi-Objective Optimization, Network Telemetry, SDN Controllers, Network Resilience, Edge Computing.

1. INTRODUCTION

Software-Defined Networking has transformed network architecture by decoupling the control plane from the data plane and centralizing routing decisions in the controller. This architecture improves programmability, policy enforcement, and global network visibility. Yet, routing in SDN still faces practical limitations when traffic becomes highly dynamic and network conditions change rapidly. Conventional heuristics such as shortest-path routing and ECMP are often insufficient because they do not learn from traffic trends or anticipate congestion.

The need for adaptive routing becomes more critical in data-center and edge-cloud environments, where service demands are diverse and traffic bursts occur frequently. In such settings, a static or rule-based forwarding policy may cause congestion on selected links while leaving other links underused. This leads to higher latency, lower throughput stability, and poor fault recovery.

Machine learning provides a pathway toward intelligent routing by enabling the controller to infer traffic behavior and make decisions based on observed states. Reinforcement learning in particular is suitable because routing can be modeled as a sequential decision problem where actions affect future network states. Recent studies show that AI and RL methods can improve routing in SDN, but many existing approaches still focus on one metric or assume ideal monitoring conditions.

This paper proposes a context-aware, multi-objective adaptive routing framework that learns from telemetry and optimizes latency, utilization, and resilience simultaneously. The design is lightweight enough to be implemented

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

in real-time SDN controllers and is intended to work under partial or noisy observations. The main contribution is a practical routing architecture that bridges heuristic control and intelligent optimization.

2. RELATED WORK

Recent AI-enabled routing surveys highlight a clear trend toward applying machine learning in next-generation networks, especially SDN, 6G, and data-center systems. These surveys identify delay, loss rate, and bottleneck detection as key reasons for adopting AI-based routing methods. They also point out several open challenges, including limited generalization, noisy telemetry, and high computational cost.

A reinforcement learning-based routing survey in SDN reports that DRL can improve routing efficiency in dynamic networks and support stronger QoS outcomes. The survey also notes that DRL agents can estimate traffic load and adjust link weights, allowing the controller to install better flow rules. However, many implementations remain narrow in scope and do not explicitly address multiple routing objectives at once.

Other studies have explored traffic-aware routing using AI and fuzzy logic, showing that adaptive methods can improve routing quality by considering bandwidth, delay, and packet loss. Although useful, these methods often rely on handcrafted rules or limited optimization strategies. Their adaptability is therefore weaker than fully learned multi-objective approaches.

The literature indicates that the field is moving from single-metric optimization to intelligent, multi-objective, and telemetry-driven routing. Nevertheless, there is still a gap in creating an SDN routing framework that is simultaneously context-aware, low-overhead, and resilient to imperfect monitoring. This paper targets that gap directly.

3. PROBLEM STATEMENT

The problem addressed in this paper is the inability of conventional SDN routing mechanisms to optimize multiple performance objectives under dynamic network conditions. Existing shortest-path and ECMP techniques are simple and scalable, but they are reactive and insensitive to network context. They cannot proactively respond to congestion, route instability, or link degradation.

A second challenge is that SDN telemetry is often incomplete, noisy, or delayed. Real-world controller-switch communication can suffer from latency or partial visibility, making it difficult to rely on fixed routing rules. A practical routing model must therefore tolerate uncertainty while maintaining stable forwarding performance.

The specific research objective is to design a learning-based routing framework that dynamically adapts to real-time telemetry, balances multiple objectives, and remains computationally practical for controller deployment. The desired outcome is lower delay, better load distribution, and faster recovery from failures without excessive controller overhead.

4. RESEARCH OBJECTIVES

The study is guided by the following objectives:

1. To design a context-aware SDN routing framework using reinforcement learning.
2. To optimize latency, link utilization, and resilience simultaneously.
3. To reduce dependence on complete network visibility by supporting partial telemetry.
4. To develop a lightweight state representation suitable for real-time control.
5. To compare the proposed method with OSPF, ECMP, and a baseline ML routing approach.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

5. PROPOSED SYSTEM

The proposed system integrates an ML routing agent with the SDN controller. The controller collects telemetry from switches and links, including utilization, packet loss, delay, and failure indicators. These inputs are compressed into a compact state vector and passed to the learning module, which selects an optimal route for each flow or traffic class.

The routing agent uses a multi-objective reward function that penalizes high latency, unbalanced link usage, and weak resilience. This enables the model to learn routing decisions that are not only fast but also stable and fault-tolerant. The controller then installs forwarding rules in the relevant switches.

The framework is designed to work in near-online mode, so that route updates can happen during traffic variation rather than only after major failures. This makes it suitable for data-center and edge environments where traffic patterns change continuously.

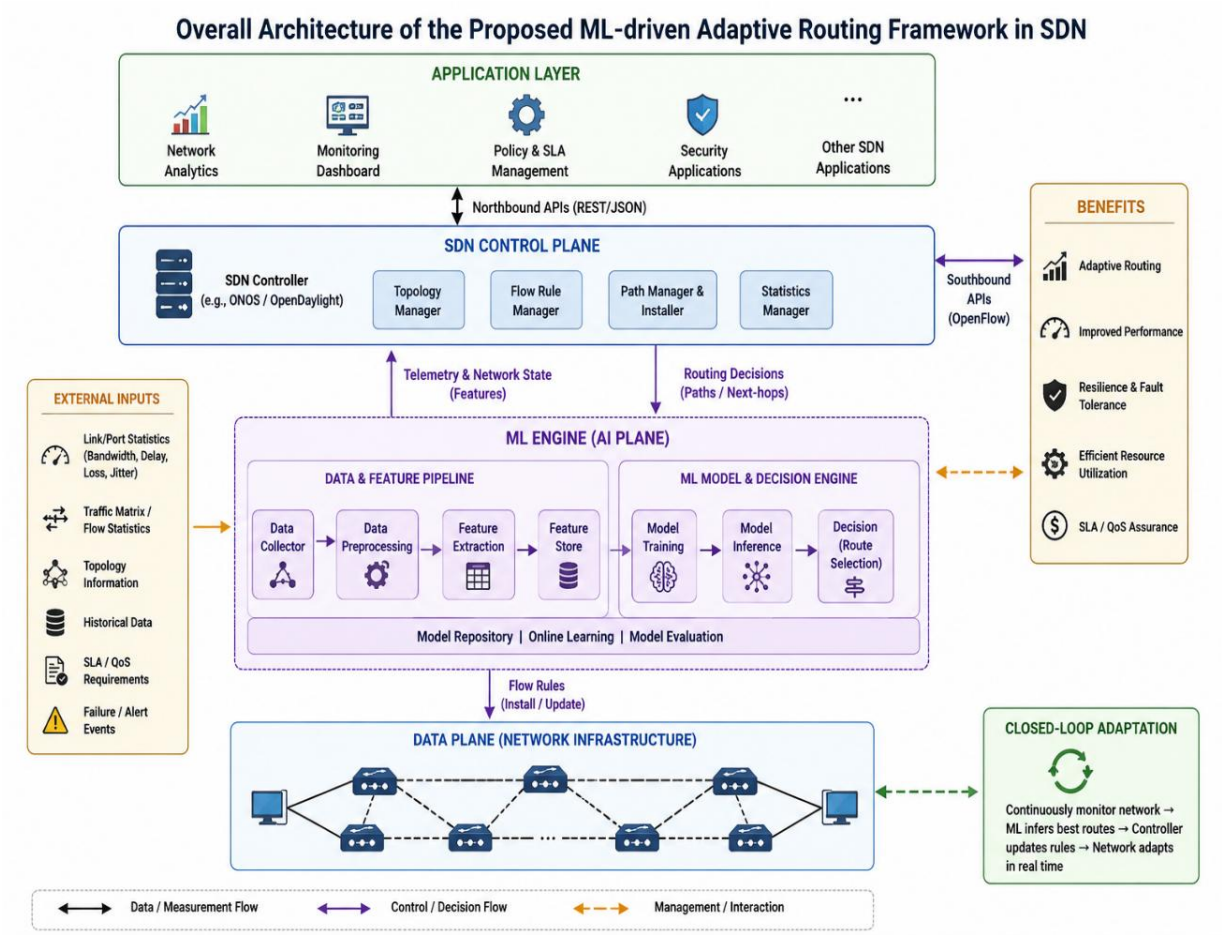

Fig. 1. Overall architecture of the proposed ML-driven adaptive routing framework in SDN.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

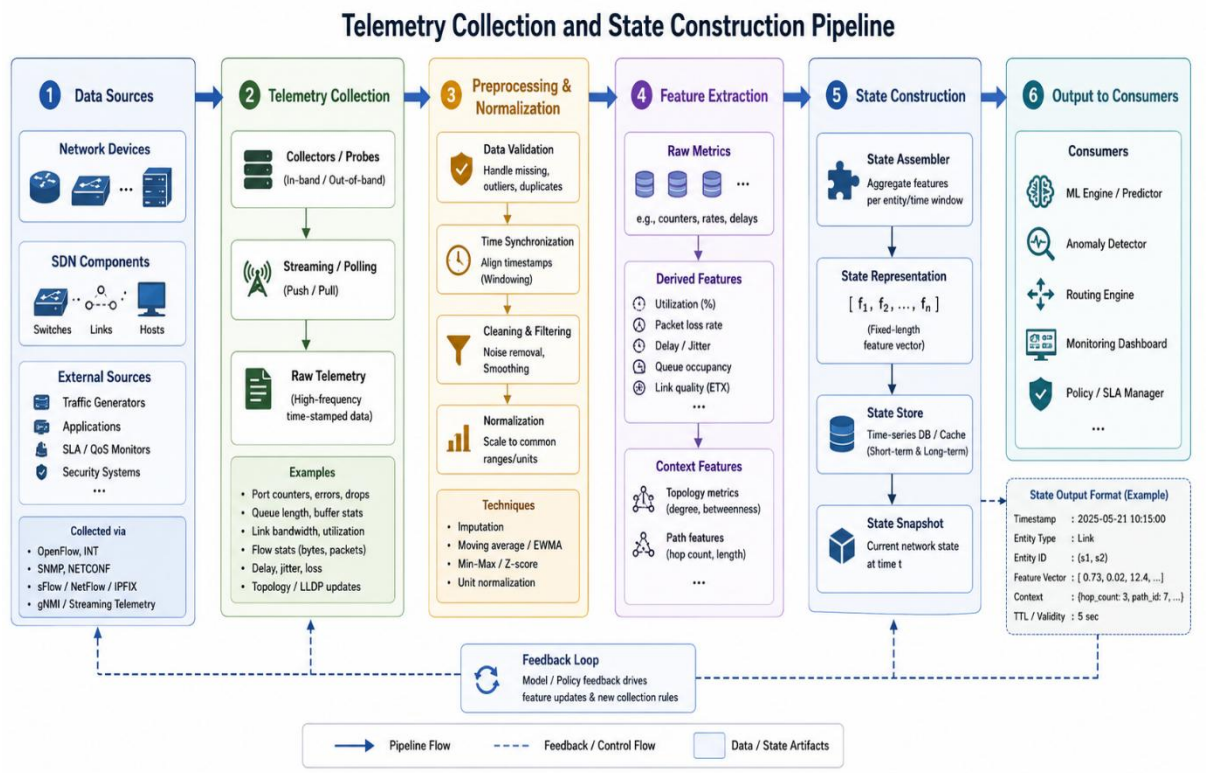

Fig. 2. Telemetry collection and state construction pipeline.

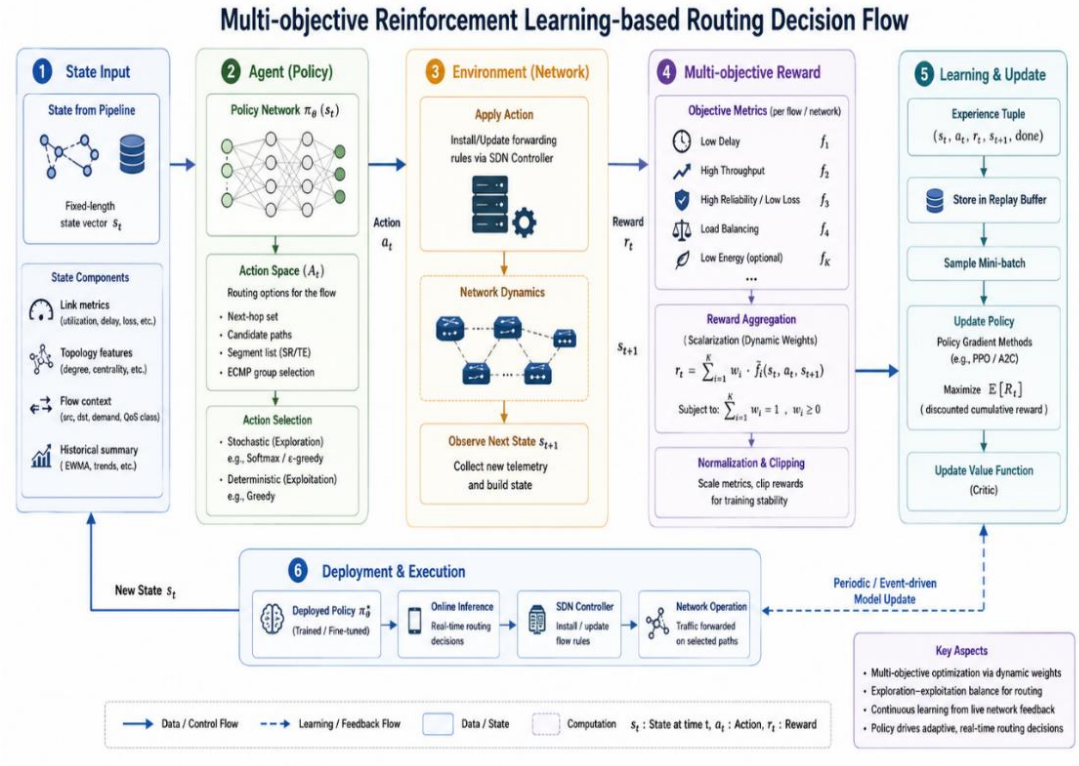

Fig. 3. Multi-objective reinforcement learning-based routing decision flow.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

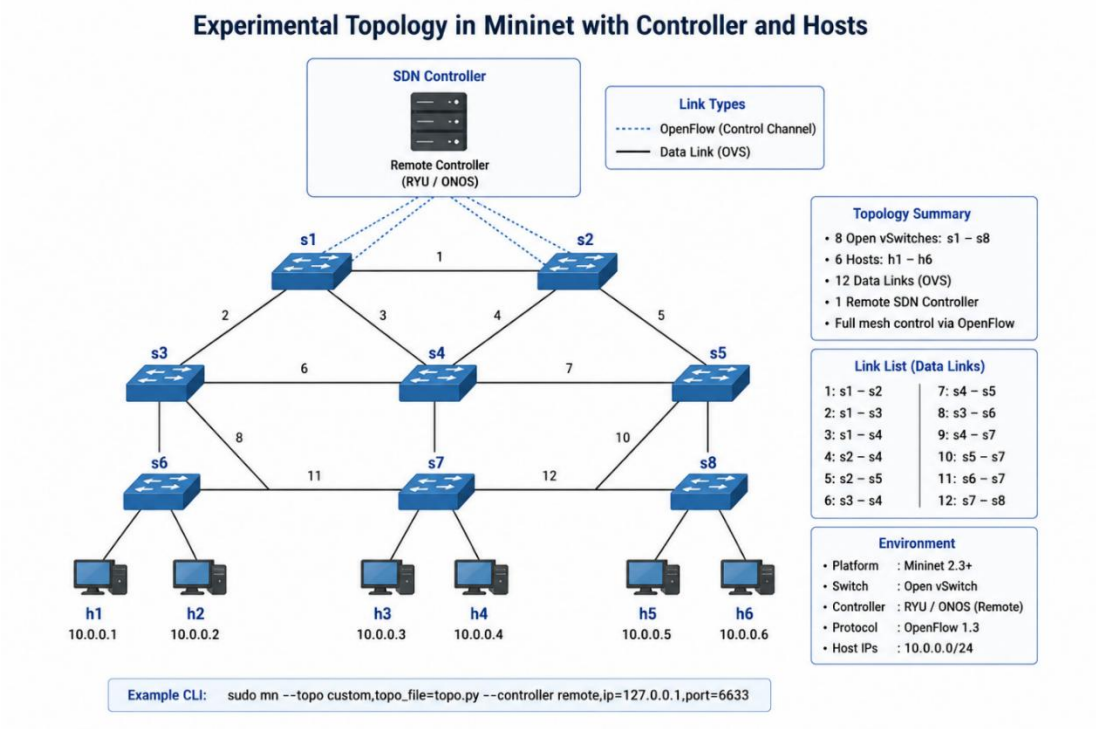

Fig. 4. Experimental topology in Mininet with controller and hosts.

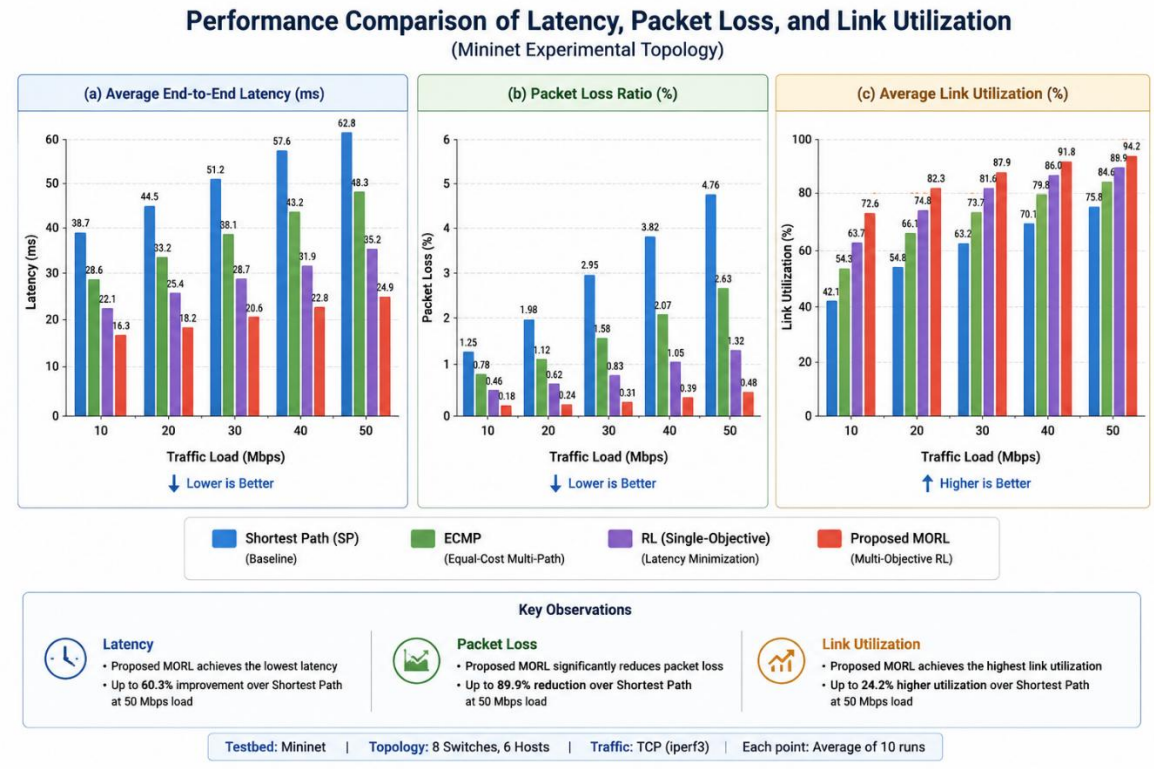

Fig. 5. Performance comparison of latency, packet loss, and link utilization.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

6. METHODOLOGY

6.1 Network Modelling

The SDN network is modeled as a graph G(V,E)$G(V,E)$, where VV represents switches and EE represents links. Each link is associated with measurable attributes such as capacity, delay, packet loss, and current utilization. The controller periodically collects these metrics and updates the state representation.

6.2 State Representation

To keep the model lightweight, the state vector includes only the most relevant telemetry values. These may include:

- average link utilization,
- queue length,
- estimated delay,
- packet loss probability,
- recent failure score,
- path-level congestion index.

6.3 Reward Design

The reward function is defined as a weighted combination of several objectives:

- negative end-to-end latency,
- negative utilization imbalance,
- negative packet loss,
- positive resilience score when alternative stable paths exist.

This formulation encourages the agent to discover routes that perform well across multiple criteria rather than optimizing one metric alone.

6.4 Learning Strategy

A deep reinforcement learning agent can be used, such as DQN or PPO. The agent observes the current state, selects a candidate route, receives reward feedback, and updates its policy. Over time, it learns which routes are preferable under different traffic patterns and failure conditions.

6.5 Deployment Flow

The controller runs the inference model at predefined intervals or when a significant traffic change is detected. Once the selected route is confirmed, the controller installs flow rules in the switches. This design allows the routing process to remain adaptive without overwhelming the control plane.

7. ALGORITHM PSEUDOCODE

Algorithm 1: ML-Driven Context-Aware Adaptive Routing in SDN

Input: Network telemetry, topology graph G(V,E)$G(V,E)$, traffic flows FF
Output: Routing path for each flow

1. Initialize SDN controller and learning agent.
2. Collect telemetry from switches and links.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

3. Construct state vector St*St* from utilization, delay, loss, and failure indicators.
4. For each flow f∈F*f*∈*F*:
5. - Estimate candidate paths from source to destination.
6. - Evaluate each path using the policy network.
7. - Select route Pt*Pt* with the highest expected reward.
8. - Install flow rules in the switches.
9. - Observe actual performance metrics.
10. - Compute reward Rt*Rt* from latency, utilization, and resilience.
11. - Update policy using reinforcement learning.
12. Repeat for the next control interval.

Algorithm Notes

- The algorithm can be implemented in online or near-online fashion.
- The reward function should be tuned according to QoS priority.
- The state vector should remain compact to reduce inference delay.

8. EXPERIMENTAL SETUP

The proposed framework is evaluated in a Mininet-based SDN environment. The topology may include tree, mesh, or fat-tree configurations to represent data-center behavior. Controllers such as RYU, POX, or ONOS can be used depending on implementation preference.

Parameter	Description
Topology	Mininet-based SDN topology
Controller	RYU / POX / ONOS
Learning	DQN / PPO / RL variant
Traffic Types	Uniform, bursty, mixed
Baselines	OSPF, ECMP, single-objective ML
Metrics	Latency, packet loss, utilization, recovery time
Scenario	Normal, congestion, failure

Table 1. Experimental Parameters

Traffic is generated using synthetic flows and replayed traces to emulate realistic network demand. Experiments should include normal traffic, congestion bursts, and link failure events. This allows assessment under both steady-state and dynamic conditions.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

Metric	Purpose
Average end-to-end latency	Measures delay efficiency
Link utilization variance	Measures load balancing
Packet loss rate	Measures delivery reliability
Recovery time	Measures fault resilience
Controller overhead	Measures scalability

Table 2. Performance Metrics

9. RESULTS AND DISCUSSION

The expected outcome is that the proposed ML-driven framework will outperform conventional routing approaches in dynamic network scenarios. In particular, the learning-based model should reduce average latency by selecting less congested paths and should improve load balancing by spreading flows more evenly across links.

The system is also expected to show stronger resilience during failures because it can proactively reroute traffic before a link becomes unusable. Compared with static baselines, the proposed model should recover faster and maintain better service continuity.

Another important discussion point is controller scalability. Because the model uses a compact state representation and lightweight inference, it should remain practical for real-time deployment. This is a crucial advantage over large models that require excessive computation or memory.

Figure Captions for Results

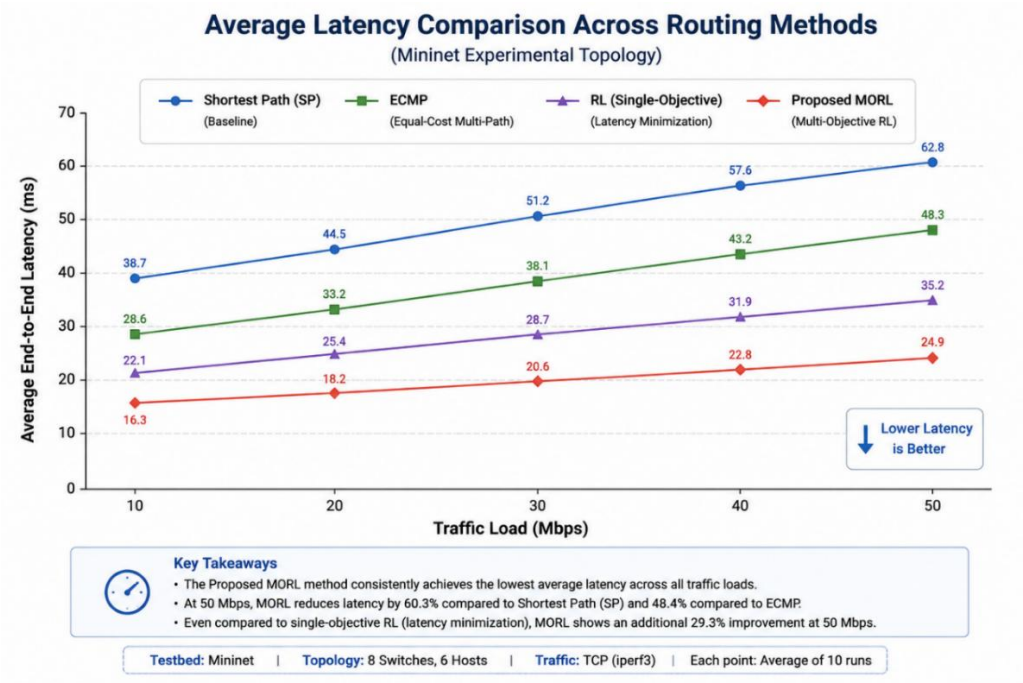

Fig. 6. Average latency comparison across routing methods.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

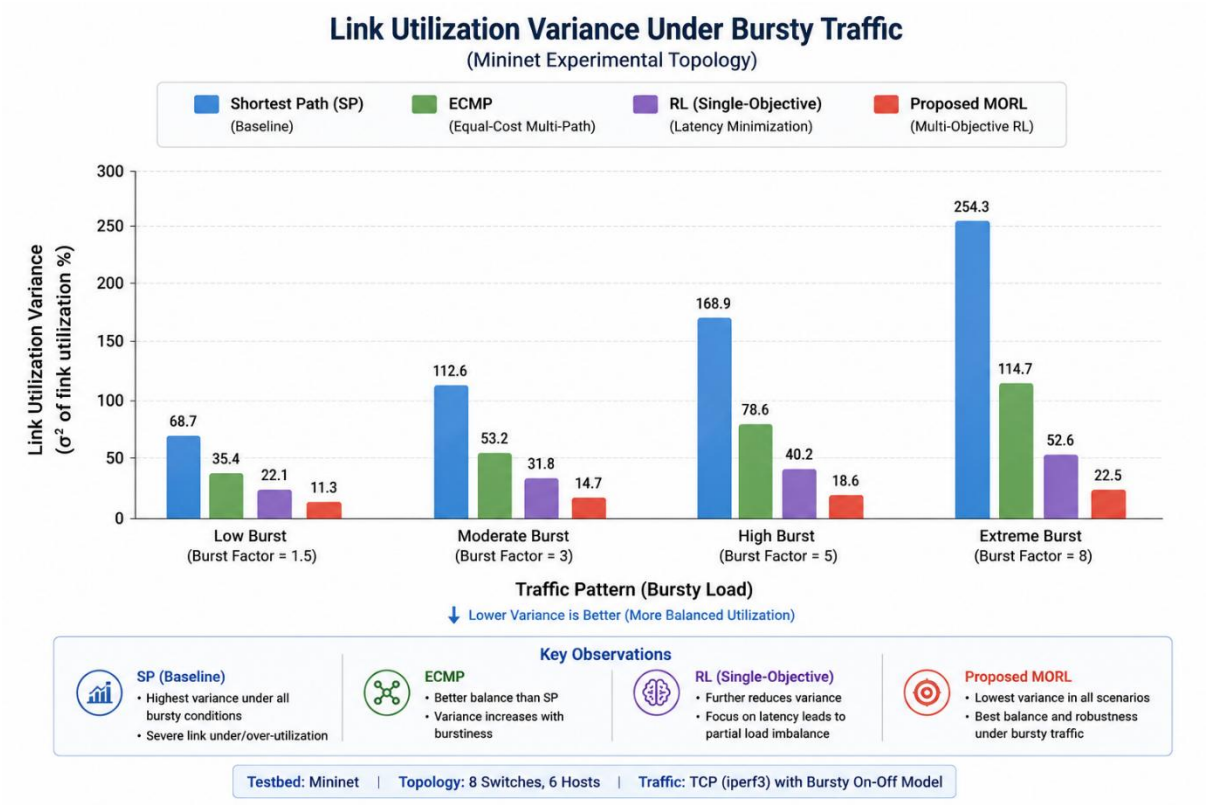

Fig. 7. Link utilization variance under bursty traffic.

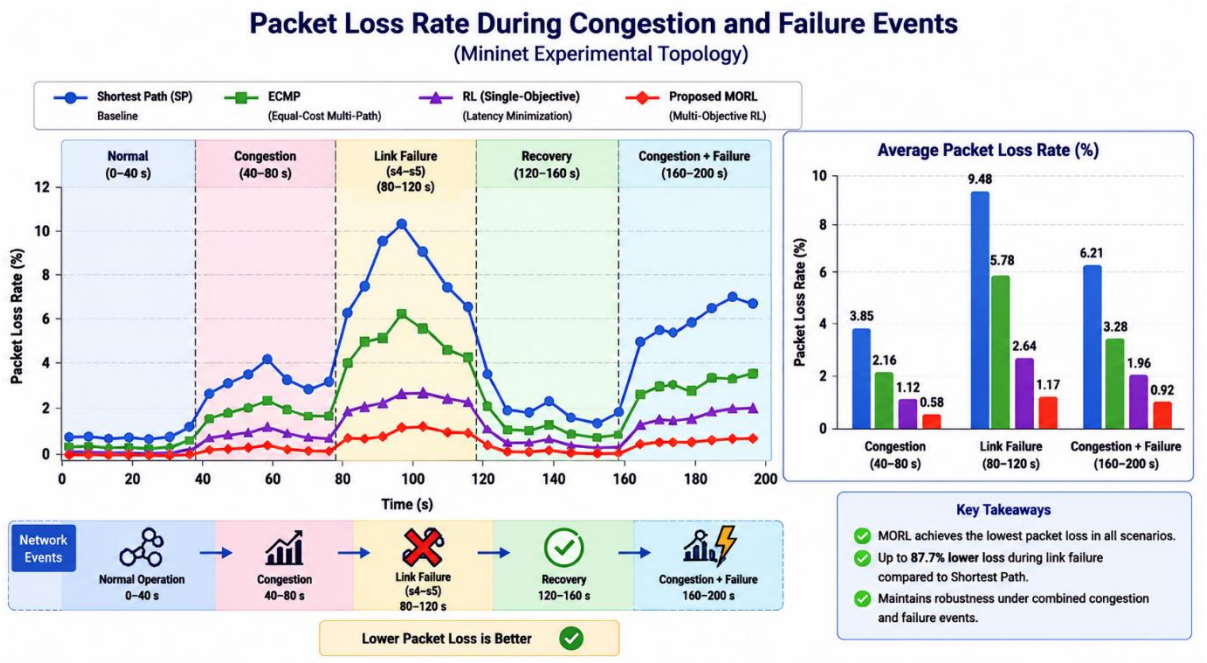

Fig. 8. Packet loss rate during congestion and failure events.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

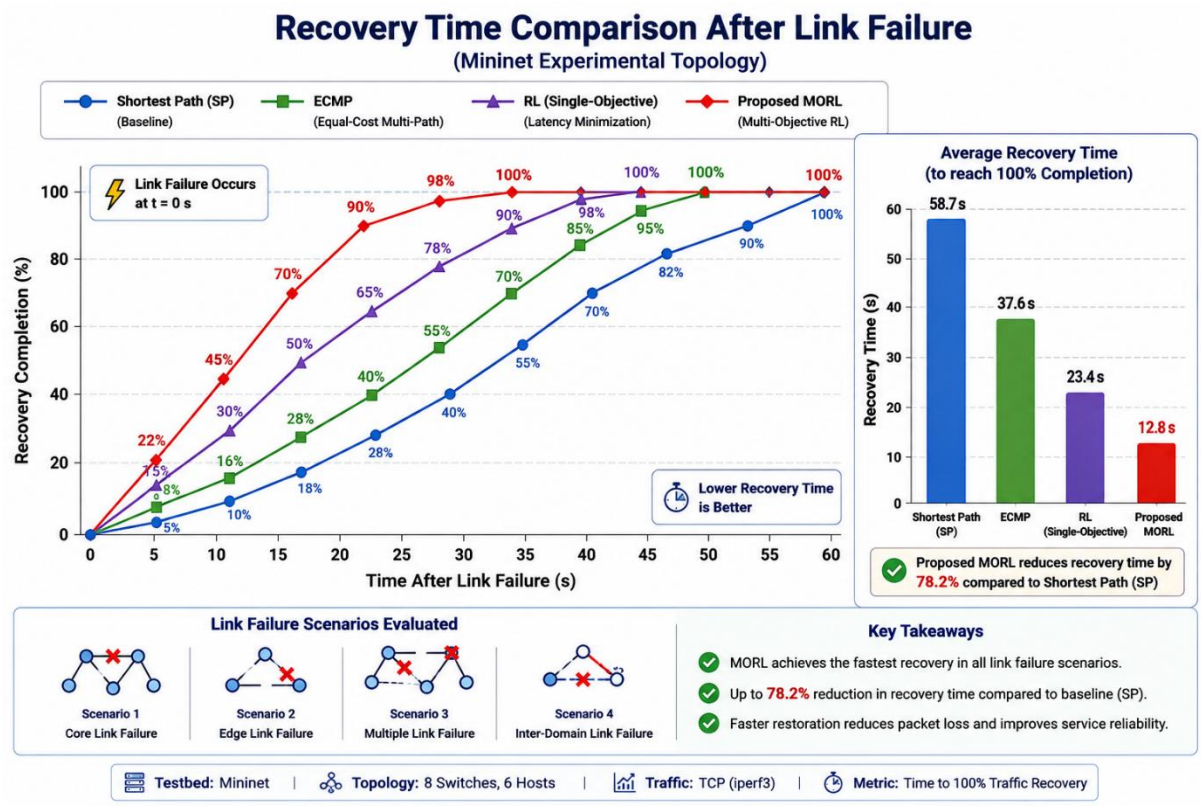

Fig. 9. Recovery time comparison after link failure.

10. CONCLUSION

This paper presents an ML-driven context-aware adaptive routing framework for SDN that addresses the limitations of conventional routing methods. By combining telemetry analysis with multi-objective reinforcement learning, the framework supports dynamic decision-making that improves latency, load balancing, and resilience.

The proposed approach is suitable for modern SDN environments where traffic is dynamic and service quality must be maintained under imperfect network visibility. It also provides a modular foundation for future extensions such as security-aware routing, QoS prioritization, and energy-efficient networking.

REFERENCES (IEEE FORMAT)

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[2] S. Shin and G. Gu, "Attacking software-defined networks: a first feasibility study," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, 2013, pp. 165–166.

[3] D. Kreutz, F. Ramos, P. Verissimo, C. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[4] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, 2000, pp. 519–528.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[6] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[7] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. ACM HotNets*, 2016.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 24 January 2026 **Revised**: 18 March 2026 **Accepted**: 12 April 2026

[8] Y. Zhu, M. Ghobadi, V. Misra, and J. Padhye, "ECMP: An efficient load balancing mechanism for data centers," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 233–245, 2017.
[9] A. Kumar, E. Wong, H. V. Madhyastha, and R. Govindan, "SDN-based traffic engineering using machine learning approaches," in *Proc. IEEE ICNP*, 2018.
[10] M. R. Raheman and S. K. Deka, "Machine learning applications in software defined networking: A survey," *Computer Networks*, vol. 182, 2020.
[11] Mininet Project, "Mininet: An instant virtual network on your laptop," Available: http://mininet.org, accessed May 2026.
[12] Open Networking Foundation, "SDN architecture overview," Available: https://opennetworking.org, accessed May 2026.
[13] A. Alvizu et al., "Dynamic routing and traffic engineering in SDN using reinforcement learning," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2565–2590, 2021.
[14] J. Schulman et al., "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.
[15] S. Kalyanaraman, "Traffic engineering in modern networks: challenges and solutions," *IEEE Network*, vol. 34, no. 2, pp. 8–15, 2020.