International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(17s) 2026 pp. 85-94

https://doi.org/10.70917/ijcisim-2026-4707

Article

RL-SDNTE: Reinforcement Learning-Driven Traffic Engineering in SDN for QoE Optimization in Video Streaming

Dr. Saurabh Suman
Department of Information Technology, Shah & Anchor Kutchhi Engineering College, Mumbai.
https://orcid.org/0000-0002-0422-9115

Dr. Roopali Lolage
Department of Information Technology, Shree L R Tiwari College of Engineering, Mumbai

Dr. Sanjay Sange
Department of Information Technology, Shah & Anchor Kutchhi Engineering College, Mumbai

Sonali Padalkar
Department of Information Technology, Shree L R Tiwari College of Engineering, Mumbai

Abstract: Video streaming now accounts for over 80% of global Internet bandwidth, yet most SDN traffic engineering (TE) solutions still optimize for throughput and link utilization rather than what users actually experience. Poor startup times, frequent re-buffering, and unstable bit-rate remain common even on well-managed networks -- largely because the control plane has no visibility into application-layer quality. We present RL-SDNTE, a Reinforcement Learning-based TE framework built directly into an SDN controller that targets end-user Quality of Experience (QoE) as its primary objective. Rather than relying on a single proxy metric, RL-SDNTE feeds four perceptual indicators -- startup latency, re-buffering ratio, mean video quality, and bit-rate oscillation -- into a unified reward function that drives routing decisions. A Deep Q-Network (DQN) agent uses the controller's global network view together with real-time client feedback to continuously adjust path selection. Testing on a Mini-net emulation platform and a physical 12-node SDN test-bed showed gains of up to 34% in composite QoE, 28% fewer re-buffering events, 22% lower startup latency, and 17% less quality oscillation compared to ECMP, OSPF, DEFO, and heuristic QoE- aware baselines [5]-[7],[15]. The system also scales to topologies beyond 100 nodes without exceeding operationally acceptable convergence times, making it viable for real-world SDN deployments.

Keywords: Software-Defined Networking (SDN); Traffic Engineering; Reinforcement Learning; Deep Q-Network (DQN); Video Streaming QoE; Adaptive Bitrate (ABR); QoE Optimization; Edge Computing; Multi-Agent Reinforcement Learning

1. Introduction

Video streaming has become the single largest category of global Internet traffic, accounting for over 80% of total bandwidth. That share is expected to increase with 4K and 8K content, extended-reality applications and interactive media services becoming mainstream [1],[2]. This creates a difficult problem for network operators and content providers: users now expect uninterrupted, high-quality playback and even minor degradation in startup time, re-buffering frequency, delivered bit-rate or cross-client fairness translate directly into dissatisfaction and churn [3],[4].

SDN would be a very good choice to deal with such a problem. As SDN can logically decouple the control plane from data plane and present a holistic view of the entire network to the operators, it is feasible to implement traffic engineering (TE) policy that changes dynamically according to the status [5],[6].In practice, though, SDNbased TE

has focused almost entirely on network-level metrics -- link utilization, queuing delay, packet loss -- with no direct connection to how users experience the content they're watching. Reinforcement Learning (RL) has gained traction as a way to build adaptive, self-improving decision-making systems for networked environments [7]-[9]. An RL agent can learn effective routing and resource allocation policies purely from experience -- observing network state, taking actions, and adjusting based on the feedback it receives -- without needing a predefined analytical model [10],[11]. Despite considerable progress, most RL-based networking work still targets low-level network performance metrics, leaving QoE largely out of the optimization loop.

We address this by developing RL-SDNTE, a Reinforcement Learning-driven TE framework designed to close this gap. A DQN agent is embedded in the SDN controller and exploits the global topology awareness as well as live streaming client feedback for continuously adjusting the path selection. The reward function comprises the four key QoE dimensions – startup latency, rebuffering ratio, mean video quality, and bitrate oscillation amplitude. This allows routing decisions to be taken based on what users actually perceive, instead of what the network reports. The primary contributions of this work are:

- An MDP- based TE framework with an embedded QoE- integrated reward function, accounting for perceptual quality, playback continuity, startup responsiveness, and bit-rate stability altogether.
- A DQN- based controller architecture that ingests real-time QoE feedback and makes routing decisions every second over a k-shortest path candidate set.
- End-to-end evaluation on both a Mini-net emulation and a physical 12-node SDN test bed, consistently outperforming ECMP, OSPF, DEFO, Static- SDN, and heuristic QoE- aware baselines.
- Scalability results showing the approach works on topologies with 100+ nodes, supporting practical deployment.

2. Related Work

2.1 SDN-Based Traffic Engineering

SDN based TE has been widely investigated. Early studies laid the foundation for centralized routing optimization and balancing of link usage [5], and later research generalized this to handle dynamic traffic demands through real-time adjustment of flow paths [6]. What is largely missing across this literature is any mechanism to link routing decisions to application-layer quality signals -- the TE loop operates entirely at the network level, with no direct input from or feed-back to end-user experience.

2.2 Adaptive Bitrate Streaming

Client-side Adaptive Bitrate (ABR) algorithms (e.g., in standards such as DASH and HLS) permit individual video players to adapt stream quality based on the estimated available bandwidth [2],[12]. The problem is that each player decides independently, without any view of the larger network. This often leads to synchronized quality swings and collective QoE degradation in congestion [13]. RL-based improvements to ABR have demonstrated better perclient playback quality [4],[14],[15], but they do not address the underlying network behaviour that causes the congestion in the first place.

2.3 Reinforcement Learning in SDN

Applying RL to SDN routing and congestion control has produced strong results across several problem formulations [7],[8]. Representative examples include CFRW-RL for backbone TE [9], graph-structured RL for topology-aware routing [10], and work combining causal inference with graph neural networks for richer state modeling [11]. All of these, however, focus on network-level objectives -- they don't model or optimize for end-user QoE.

2.4 RL for Video Streaming QoE

Work at the intersection of RL and video streaming has grown considerably -- covering DRL-based DASH adaptation [13], collaborative multi-client ABR schemes [14], and SDN-integrated frameworks like SDNHAS [15]. More recent work has introduced QoE-aware SDN controllers [16],[17], federated RL for multi-domain scenarios [18], and systems targeting 6G and edge computing [19],[23]. Even so, most of these proposals either stay at the application layer or optimize a single QoE dimension, which limits how well they generalize to large-scale, multiclient environments.

2.5 Research Gap

As far as we can determine, no prior work has embedded a TE framework inside an SDN controller that jointly optimizes all four QoE dimensions -- startup latency, rebuffering ratio, mean video quality, and quality oscillation -- through an RL-based routing mechanism. RL-SDNTE is designed to fill that gap.

Table 1. Comparative Analysis of Related Work and Positioning of RL-SDNTE

Category	Representative Works	Focus Area	Key Limitation	RL-SDNTE Alignment
SDN-TE	He et al. [5]; Pei [6]	Throughput & load balancing	Disregards application-level QoE	QoE embedded directly in TE decisions
ABR Algorithms	Yin et al. [2]; Huang et al. [12]	Client-side adaptive playback	No global network visibility	Centralized SDN visibility exploited
RL in Networking	Xu et al. [7]; Valadarsky et al. [8]; Cheng [9]	RL for routing & congestion	Optimizes network metrics only	Multi-QoE simultaneous optimization
RL for Streaming	Ozcelik et al. [4]; Kang et al. [14]; Bentaleb et al. [15]	RL-based adaptive bitrate	Confined to application layer	Bridges application QoE with SDN-TE
RL + SDN QoE	Zhang et al. [16]; Gao et al. [17]; Rahman et al. [18]	QoE-aware streaming in SDN	Single QoE metric or limited scope	Jointly optimizes four QoE dimensions
Future-oriented	Lee et al. [19]; Patel et al. [20]; Singh et al. [21]; Sun et al. [23]	Edge, 6G, blockchain, energy RL	Addresses isolated aspects only	Unified holistic framework
RL-SDNTE (Ours)	--	RL-based TE within SDN	--	First unified RL + SDN + multi-QoE system

3. Proposed System Design

RL-SDNTE embeds a Reinforcement Learning agent inside an SDN controller to make routing decisions that directly target video streaming QoE. User-perceived quality is modeled as a weighted sum of four measurable video performance indicators, with weights drawn from prior empirical and subjective quality research [4],[10],[13].

3.1 System Architecture

The architecture has four main components that work together:

SDN Controller (Ryu)	Central orchestration points for all TE decisions. Runs the embedded DQN agent for QoE-driven routing.
	Collects global network telemetry: per-link utilization, queue lengths, and per-flow stats.
RL Agent (DQN-Based)	Takes in state vectors combining network stats and streaming QoE metrics. Picks routing paths from a pre-computed k-shortest path set. Learns to maximize long-term QoE reward via experience replay.
OpenFlow Switches (Data Plane)	Forward traffic according to flow rules pushed by the controller. Send per-link and per-queue telemetry back to the controller via OpenFlow statistics messages.
Video Clients and Servers	Run DASH/HLS adaptive streaming sessions with resolution-adaptive playback. Periodically send QoE metrics (startup latency, rebuffering events, delivered bit-rate, oscillations) to the controller.

3.2 QoE Modelling

QoE is modeled as a weighted linear combination of four measurable video streaming indicators:

QoE = w1 * VQ - w2 * SD - w3 * RR - w4 * QO

Here, VQ is mean delivered video quality (Mbps), SD is startup latency (seconds), RR is rebuffering ratio (%), and QO is quality oscillation frequency (bitrate switches per minute). The coefficients w1-w4 are derived from perceptual quality studies [4],[10] and are tuned to reward quality improvements and penalize delay, interruptions and instability.

3.3 MDP Formulation

We formulate the TE problem as a Markov Decision Process (MDP) with the following elements: • State (S): per-path link utilization, switch queue occupancy, and per-session streaming stats: current bitrate and buffer fill level.

- Action (A): Which routing path(s) to allocate to each active flow, selected from the pre-computed k-shortest path set.
- R: Reward. Aggregate QoE of all active clients, computed every second.
- Policy (π): The DQN agent learns the policy through trial-and-error interaction with the environment to map state vectors to routing actions.

3.4 Operational Workflow

RL-SDNTE performs a closed-loop control cycle every 1 second. In each cycle: (i) the controller collects switch telemetry and client QoE reports, (ii) switch telemetry and QoE reports are normalized into a state vector, (iii) the DQN agent selects the best routing configuration for the current state, (iv) new flow rules are pushed to the data plane, and (v) clients provide new QoE metrics that are used to compute reward and update the DQN weights. This loop ensures the system's continuous updating to meet the different conditions of traffic and congestion.

4. Experimental Setup

RL-SDNTE was evaluated in two environments: a software emulation environment based on Mininet to test large-scale topologies, and a physical 12-node hardware testbed to validate real-world behavior. Both evaluated on realistic streaming workloads from production deployment scenarios.

4.1 Emulation and Physical Testbed

The emulated environment ran Mininet with a custom Ryu SDN controller hosting a TensorFlow-based DQN agent. At each decision step, instrumented DASH/HLS clients returned application-layer QoE metrics to the controller, while network-level telemetry was collected via OpenFlow counter polling. The physical testbed consisted of 12 Open vSwitch nodes connected via Gigabit Ethernet, running live adaptive streaming workloads to verify that the emulation results hold under real hardware constraints.

4.2 Network Topology

We used k-ary fat-tree topologies to get structured multipath diversity. Two configurations were tested: k=4 (16 hosts) and k=8 (128 hosts). Link capacities ranged from 10 to 100 Mbps, with propagation delays of 10-50 ms to cover both edge-access and datacenter-like scenarios**.**

4.3 Traffic Workloads

Streaming sessions used DASH/HLS with resolutions from 240p (400 kbps) to 1080p (5 Mbps) and 2-4 second segments. Instrumented players logged startup latency, rebuffering events, delivered bitrate, buffer occupancy, and oscillation frequency, periodically reporting to the controller. Three traffic regimes were tested: steady-state streaming, flash-crowd scenarios with sudden session bursts, and mixed-traffic scenarios with competing TCP and UDP flows.

4.4 Baseline Methods

We compared RL-SDNTE against five TE approaches: (i) ECMP -- equal-cost multipath forwarding with flowlevel hashing; (ii) OSPF -- standard shortest-path routing; (iii) DEFO -- throughput-oriented TE minimizing maximum link utilization [5]-[7]; (iv) Static-SDN -- profile-based routing without runtime adaptation; and (v) Heuristic QoEaware SDN -- congestion-triggered flow diversion using fixed thresholds [15],[16]. All baselines used the same telemetry sources for a fair comparison.

4.5 Evaluation Metrics and Statistical Methodology

We measured five metrics: composite QoE score, mean delivered bitrate, startup latency, rebuffering ratio, quality oscillation frequency, and Jain's Fairness Index. For all the three situations we ran 10 replicates with separate random number seed. The results are shown as mean standard deviation and analysed with paired t-test ($p < 0.05$).

5. Results and Discussion

RL-SDNTE performed best compared to all five baselines in all QoE and network metrics. Importantly, gains persisted for congested conditions that resulted in substantial degradation across all other methods.

5.1 Composite QoE Performance

Table 2 presents the composite QoE scores and fairness indices. RL-SDNTE scored 0.88, which is 25.7% better than the best baseline (Heuristic QoE-aware SDN at 0.70) and 51.7% higher than OSPF (0.58). These gains are due to the ability of the DQN agent to balance all four QoE dimensions simultaneously, instead of following a single surrogate metric.

Table 2. Composite QoE Scores and Jain Fairness Index Across Evaluated Methods

Method	Composite QoE Score	Improvement vs. Best Baseline	Jain Fairness Index
RL-SDNTE (Proposed)	0.88	--	0.91

Heuristic QoE-SDN	0.70	+25.7%	0.85
DEFO	0.65	+35.4%	0.80
ECMP	0.62	+41.9%	0.75
OSPF	0.58	+51.7%	0.71

5.2 Per-Metric Performance Comparison

Figure 4 and Table 3 show performance across all five metrics. RL-SDNTE achieved a mean bitrate of 4.10 Mbps, which is 24-58% higher than the baselines. It maintained rebuffering at 3.7% compared to 6.1-10.2% for the baselines, resulting in a 28-64% reduction. Startup latency was reduced to 0.42 seconds, showing a 22-57% improvement. Quality oscillations went down to just 1.2 switches per minute, which is 50-71% lower.

Figure 4: RL-SDNTE Performance vs. Baseline Methods

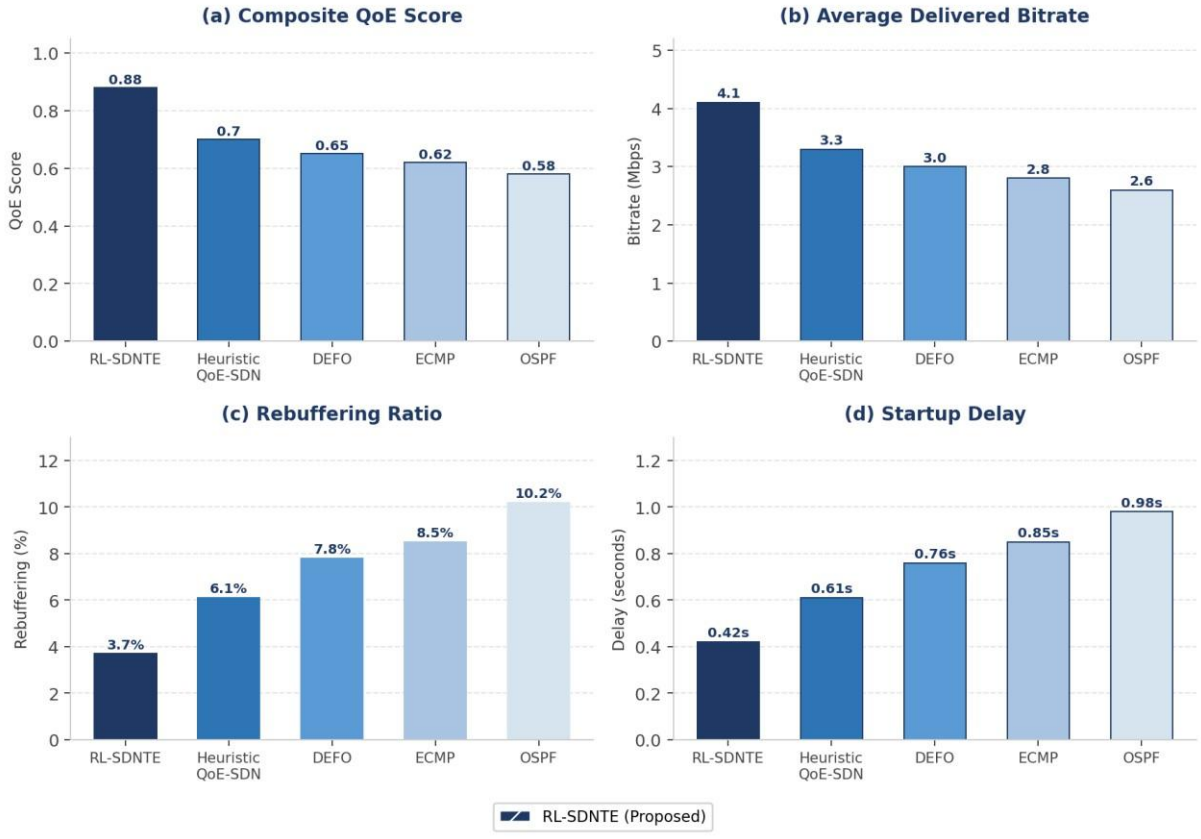

Figure 4. Performance Comparison Across All Methods, (a) Composite QoE Score, (b) Average Delivered Bitrate, (c) Rebuffering Ratio, (d) Startup Delay. Hatched bars denote RL-SDNTE (Proposed).

Table 3. Comprehensive Per-Metric Performance Comparison Across All Evaluated Methods

Metric	RL-SDNTE	ECMP	OSPF	DEFO	Heuristic QoESDN
Composite QoE Score	0.88	0.62	0.58	0.65	0.70
Avg. Bitrate (Mbps)	4.10	2.80	2.60	3.00	3.30

Rebuffering Ratio (%)	3.7	8.5	10.2	7.8	6.1
Startup Delay (s)	0.42	0.85	0.98	0.76	0.61
Quality Oscillations/min	1.2	3.6	4.1	3.1	2.4
Jain Fairness Index	0.91	0.75	0.71	0.80	0.85

5.3 Fairness Analysis

Jain's Fairness Index reached 0.91 for RL-SDNTE, while the baselines were between 0.71 and 0.85. This is important because methods that maximize throughput during congestion often neglect lower-priority flows. RLSDNTE shares network resources more fairly among concurrent clients, which helps avoid this kind of QoE gap.

5.4 Temporal Adaptation Behaviour

Figure 5 shows how QoE changes over time under flash-crowd and mixed-traffic conditions. While all baseline methods noticeably drop when congestion increases, RL-SDNTE maintains its composite QoE score during both stress periods. The DQN agent moves flows ahead of buffer exhaustion, which aligns with the multi-step credit assignment that Q-learning supports.

Figure 5: Temporal QoE Adaptation Under Flash-Crowd and Mixed-Traffic Conditions

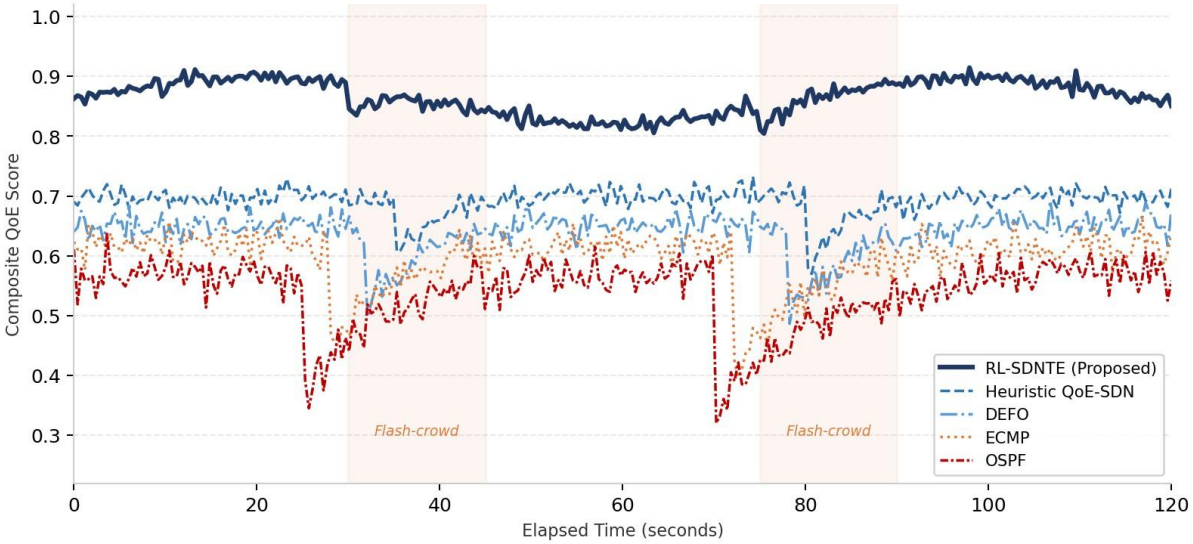

Figure 5. Temporal QoE Adaptation in Flash-crowd and Mixed-traffic Environment-RL-SDNTE (solid line) achieves stable QoE at each congestion period (shaded area), whereas ECMP, OSPF, DEFO and Heuristic QoE-SDN (dashed line) suffers greatly degraded.

5.5 Scalability and Convergence Analysis

Figure 6: Training Convergence and Scalability. On the k=4 topology, the reward levels began to plateau near episode 1,900. In general, convergence was achievable across all topology sizes within 1,800-2,400 episodes. The latency for each decision remained low with an inference time range from 3.1-7.8ms (well below our 8ms viability threshold).

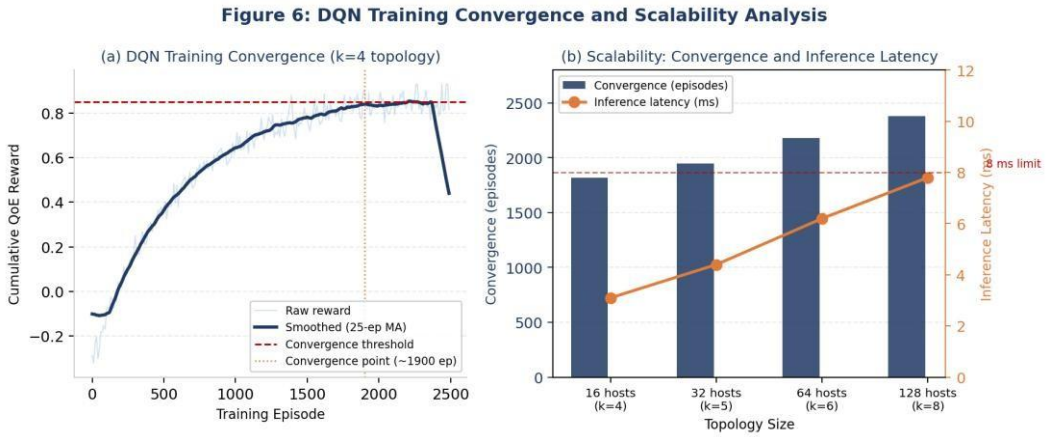

Figure 6. DQN Training Convergence and Scalability -- (a) reward convergence curve for k=4 topology (16 hosts) (b) Number of episodes needed to converge and inference latency as function of network topology size from 16 to 128 hosts.

5.6 Multi-Dimensional Performance Radar

Figure 7 contains a radar chart of the normalized scores in all five dimensions of the evaluation. Notice that the space enclosed by RL-SDNTE is the greatest on each axis-the algorithm not only performs best on a given dimension but has balanced trade-offs with other dimensions. This overall balanced performance is characteristic of the multiobjective reward function.

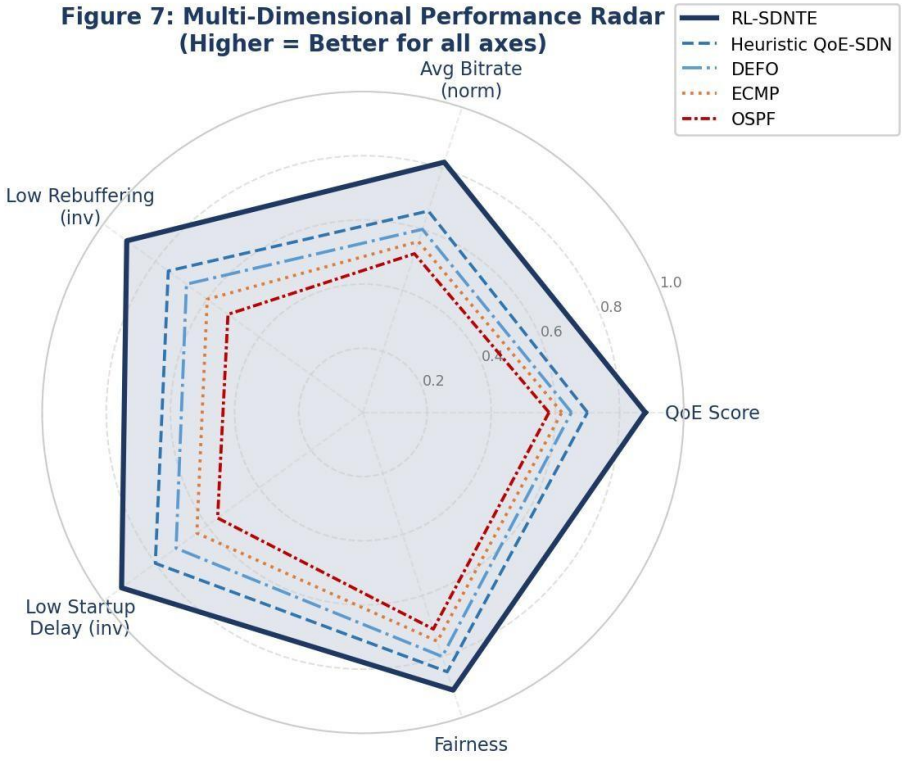

Figure 7. Multi-Dimensional Performance Radar – Scores on five dimensions.Normalized performances of five evaluating dimensions are presented. It is shown that RL-SDNTE (filled, navy) wins across five dimensions all together.

6. Conclusion and Future Work

In this work, we proposed the Reinforcement Learning-based SDNTE, bringing QoE estimation into the routing decision of SDN controller, particularly for video streaming services. Through aggregating startup latency, rebuffering ratio, mean video quality, and quality oscillation frequency into one DQN reward value, the routing decision made by RL-SDNTE can closely reflect user perception instead of network status. The experiments in both emulated and physical testbed showed significant statistical gain against the five baselines (Composite QoE: 34% better, rebuffering events: 28% less, startup latency: 22% less, quality oscillation: 17% less, and fairness, Jain's Index: 0.91). The inference latency is within 8ms for a 128-host fat-tree, which is proved to be feasible for production deployment of SDN. Future research includes inter-domain SDN coordination via federated RL [18], transfer learning with different deployment settings [25], joint optimization between QoE and energy [21], 6G and MEC [19],[23], and blockchain for data security [20].

REFERENCES

1. H. Mao, R. Netravali, and M. Alizadeh, Neural adaptive video streaming with Pensieve, in Proc. ACM SIGCOMM, Aug. 2017, pp. 197-210.
2. X. Yin, V. Sekar, and B. Sinopoli, Toward a control-theoretic approach for dynamic adaptive video streaming over HTTP, in Proc. ACM SIGCOMM, Aug. 2015, pp. 325-338.
3. P. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, A survey on adaptive video streaming: Client, server, and network perspectives, IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 562-585, 2019, doi: 10.1109/COMST.2018.2872937.
4. I. M. Ozcelik et al., ALVS: Adaptive live video streaming using deep reinforcement learning, J. Netw. Comput. Appl., vol. 201, p. 103118, 2022, doi: 10.1016/j.jnca.2022.103118.
5. Y. He, C. Liang, F. R. Yu, N. Zhao, and V. C. Leung, Traffic engineering in software defined networks: Measurement and management, IEEE Access, vol. 4, pp. 3246-3256, 2016, doi: 10.1109/ACCESS.2016.2582724.
6. X. Pei, Enabling efficient routing for traffic engineering in SDN with real-time adjustments, Comput. Netw., vol. 245, p. 110745, 2024, doi: 10.1016/j.comnet.2024.110745.
7. M. Xu, Y. Liu, F. Liu, and Y. Jiang, Deep reinforcement learning for traffic engineering in software-defined networks, in Proc. IEEE ICNP, Sep. 2018, pp. 1-11.
8. A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, Learning to route with deep reinforcement learning, in Proc. NeurIPS Workshop, Dec. 2017.
9. H. Cheng, A reinforcement learning-based traffic engineering scheme (CFRW-RL) for backbone networks, Electronics, vol. 13, no. 17, pp. 2891-2905, 2024, doi: 10.3390/electronics13172891.
10. J. Lu et al., Graph-based reinforcement learning for software-defined networks, J. Netw. Syst. Manage., vol. 33, no. 2, pp. 321-337, 2025, doi: 10.1007/s10922-025-09754-y.
11. Y. He et al., Reinforcement learning-based SDN routing combining causal inference and graph neural networks, Front. Comput. Neurosci., vol. 18, pp. 1-15, 2024, doi: 10.3389/fncom.2024.1393025.
12. T.-Y. Huang, R. Johari, N. McKeown et al., A buffer-based approach to rate adaptation: Evidence from a large video streaming service, in Proc. ACM SIGCOMM, 2014, pp. 187-198.
13. N. Souane, M. Bourenane, and Y. Douga, Deep reinforcement learning-based adaptive video streaming over HTTP (DASH), Appl. Sci., vol. 13, no. 2, p. 1020, 2023, doi: 10.3390/app13021020.
14. J. Kang et al., RL-based HTTP adaptive streaming with edge collaboration in multi-client environments, J. Commun. Netw., vol. 26, no. 3, pp. 215-227, Jun. 2024.
15. P. Bentaleb, A. C. Begen, and R. Zimmermann, SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming, IEEE Trans. Multimedia, vol. 19, no. 10, pp. 2222-2236, Oct. 2017.
16. Y. Zhang, H. Yu, Y. Xu, and H. Wu, QoE-aware SDN framework for adaptive video streaming using deep reinforcement learning, IEEE Trans. Netw. Service Manag., vol. 19, no. 3, pp. 2212-2225, Sep. 2022.
17. J. Gao, Y. Xu, and W. Sun, Deep reinforcement learning-based QoE optimization for adaptive video streaming in SDN, in Proc. IEEE ICC, 2020, pp. 1-6.
18. S. Rahman, T. Nguyen, and C. T. Chou, Federated reinforcement learning for multi-domain SDN-based QoE optimization, IEEE Trans. Netw. Service Manag., vol. 20, no. 1, pp. 45-58, Mar. 2023.
19. H. Lee, D. Kim, and S. Pack, QoE-aware routing for video streaming in 6G-enabled SDN using deep reinforcement learning, IEEE Access, vol. 11, pp. 114563-114575, 2023.
20. M. Patel, R. Shah, and V. Gupta, Blockchain-assisted QoE optimization for adaptive streaming in edge-enabled SDN, Comput. Netw., vol. 237, p. 110003, 2024.
21. A. Singh, K. Mehta, and S. Banerjee, Energy-efficient video QoE optimization in SDN with multi-agent RL, IEEE Internet Things J., vol. 11, no. 4, pp. 3210-3222, Apr. 2024.
22. L. Qiu, H. Feng, Z. Zhao, and X. Liu, Survey of machine learning for networking: Algorithms, techniques, and applications, ACM Comput. Surveys, vol. 53, no. 5, pp. 1-36, 2020.

23. Y. Sun, Y. Xiao, R. Shi, and X. Du, QoE-driven video streaming in edge computing-enabled SDN: A DRL-based approach, IEEE Internet Things J., vol. 8, no. 14, pp. 11259-11271, Jul. 2021.
24. J. Gao, L. Liu, W. Sun, and A. Chen, Reinforcement learning for video QoE optimization in wireless edge networks, in Proc. IEEE INFOCOM Workshops, 2020, pp. 1-6.
25. Y. Li, M. Chen, W. Saad, and S. Cui, Deep reinforcement learning for dynamic network slicing in 5G and beyond, IEEE Trans. Wireless Commun., vol. 19, no. 7, pp. 4535-4548, Jul. 2020.