

A LINEAR EMULATION AND MACHINE LEARNING APPROACH FOR OPTIMIZED ROUTING ALGORITHMS

Azamova Saodat Fayzullayevna

Abstract

Optimizing routing decisions in networks with high traffic variability and stringent real-time requirements is a persistent challenge. This paper proposes a hybrid approach that integrates linear emulation with machine learning to produce low-latency, resource-efficient routing policies. A lightweight linear emulator provides fast estimates of local network metrics (latency, packet loss, available bandwidth) while a machine learning model—configured for either supervised regression or reinforcement learning—captures global and historical traffic patterns to predict optimal next-hop selections. The system fuses emulator outputs and ML predictions in real time to drive adaptive load distribution and route reconfiguration under changing conditions. We evaluate the approach through packet-level simulations and a hardware testbed, comparing it against static routing schemes and baseline dynamic algorithms. Results show meaningful reductions in end-to-end latency and packet loss, improved bandwidth utilization, and modest computational overhead suitable for deployment in SDN and resource-constrained IoT environments. The paper discusses trade-offs related to model generalization, real-time constraints, and emulator accuracy, and outlines extensions such as graph neural networks for topological awareness and online learning for continuous adaptation. Overall, the proposed method offers a practical pathway to enhance routing performance in modern, heterogeneous networks.

Keywords: routing optimization, linear emulation, machine learning, supervised regression, reinforcement learning, adaptive routing, SDN, IoT, latency reduction, packet loss mitigation, real-time network control, emulator-ML fusion

Introduction

Optimizing routing decisions in modern networks is a multifaceted problem driven by rapid growth in traffic volumes, heterogeneity of devices and applications, and stringent quality-of-service (QoS) demands. Traditional routing protocols—designed for relatively static topologies and predictable traffic patterns—struggle to maintain performance when faced with bursty flows, microbursts in datacenter fabrics, or highly dynamic wireless and IoT environments. These challenges manifest as increased end-to-end latency, higher packet loss, unfair bandwidth allocation, and inefficient utilization of network resources. As networks evolve toward software-defined architectures and embrace programmable data planes, there is an opportunity to reimagine routing mechanisms that combine fast, low-overhead modeling of network behavior with data-driven decision policies that adapt to temporal patterns and complex dependencies.

The core idea behind a hybrid linear emulation and machine learning approach is to exploit complementary strengths: linear emulation offers lightweight, interpretable, and fast approximations of local network metrics based on immediate measurements, while machine learning models capture non-linear, global, and historical relationships that are difficult to express analytically. A linear emulator can be tuned to quickly estimate latency, queuing delays, or expected packet loss using recent measurements and simplified queueing models; these estimates are produced with minimal computational cost and predictable latency. Conversely, ML models—trained on historical telemetry, flow statistics, and topological features—can predict future link loads, infer congestion propagation patterns, and recommend routing adjustments that account for broader context beyond local observations. By fusing these two sources of information in real time, a control mechanism can issue routing policies or next-hop decisions that are both responsive and informed.

Existing literature spans several relevant paradigms. Model-based approaches, including analytic queueing theory and linear approximations, provide solid baseline predictions with strong interpretability and bounded complexity.

Purely data-driven strategies—ranging from supervised learning for traffic prediction to reinforcement learning (RL) for control—have demonstrated the ability to discover novel policies, but they can suffer from sample inefficiency, instability under distribution shifts, and high inference costs. Recent advances in hybrid systems and model-guided RL suggest that integrating simple physics or analytic priors with ML components yields better sample efficiency, more robust generalization, and safer behavior in production environments. In networking specifically, there is growing work on ML-assisted congestion control, learned scheduling, and adaptive routing; however, many proposals rely on heavy models or lack timely, accurate local measurements for closed-loop control.

Motivated by these observations, this work develops a practical framework for optimized routing that (1) maintains low computational overhead suitable for deployment on controllers or edge devices, (2) preserves the interpretability and reliability of analytic estimates, and (3) leverages ML capabilities to capture complex spatio-temporal patterns and to adapt over time. The framework centers on three core components: a lightweight linear emulator, a machine learning predictor (either supervised regression or RL-based policy), and a fusion layer that combines emulator outputs and ML signals to produce routing actions. The linear emulator targets short-horizon, per-link or per-path estimations using a small set of features (e.g., current queue lengths, recent RTT samples, measured throughput) and simple linear models that can be updated incrementally. The ML predictor is trained on aggregated historical telemetry, including flow arrival rates, time-of-day effects, known traffic classes, and topological embeddings; when implemented as an RL agent, it explores routing choices in simulation or controlled testbeds and optimizes long-term objectives such as average latency or tail latency percentiles.

Key research questions guiding the study include: How should the emulator and ML components be architected and interfaced to balance responsiveness, accuracy, and computational cost? What fusion strategy (weighted ensemble, gating network, or rule-based override) yields robust performance across diverse

traffic regimes? How does the hybrid approach compare with strong baselines—static shortest-path routing, ECMP, and adaptive heuristics—under a range of workloads and failure scenarios? And importantly, what practical considerations (model update cadence, telemetry sampling rate, and safety constraints) must be addressed for feasible deployment in SDN controllers or resource-constrained edge nodes?

To answer these questions, the work adopts a mixed evaluation strategy: microbenchmarks validate emulator fidelity and ML predictor accuracy on synthetic traces; packet-level simulations evaluate end-to-end performance metrics under controlled variability; and hardware testbed experiments assess real-world feasibility, latency of decision pipelines, and interaction with existing control planes. Performance measures include mean and tail latencies, packet loss rates, bandwidth utilization, and computational overhead (CPU, memory, and decision latency). The contributions of the paper are threefold: (1) a design of a computationally lightweight linear emulator tailored for rapid network metric estimation; (2) a combined emulator-ML fusion architecture with concrete fusion mechanisms and safety constraints; and (3) an empirical demonstration that the hybrid approach achieves significant improvements in latency and packet loss compared to baseline methods while keeping resource usage within practical limits for SDN and IoT contexts.

By framing routing optimization as a joint modeling and learning problem, this research seeks to bridge the gap between analytically tractable approximations and flexible data-driven controllers. The resulting method aims not only to improve immediate performance metrics but also to provide a pathway for continuous learning and adaptation: as telemetry accumulates, the ML component refines predictions; as topology or traffic regimes change, the emulator can be reparameterized quickly; and the fusion logic enforces safety and stability, preventing abrupt or counterproductive routing oscillations. This pragmatic orientation—favoring deployable architectures over purely theoretical gains—

guides the methodological choices and evaluation scenarios presented in the remainder of the manuscript.

Conclusion

The hybrid approach combining linear emulation with machine learning offers a compelling trade-off for routing optimization in contemporary networks: it harnesses the speed and interpretability of simple analytic models while leveraging ML's capacity to model complex, time-dependent phenomena. Across simulation and testbed experiments, the proposed system consistently reduced average and tail latencies, lowered packet loss, and improved bandwidth utilization relative to static and several adaptive baselines. These gains were obtained with modest computational overhead, demonstrating that practical, real-time decision pipelines are achievable without resorting to prohibitively complex models.

Several insights emerged from the evaluation. First, the linear emulator provides a reliable short-horizon predictor that excels when local conditions dominate performance. Its fast inference allows the control loop to respond to sudden changes such as transient congestion or link degradation. Second, ML predictors—particularly those incorporating temporal context and lightweight topological features—capture recurring load patterns and anticipate congestion, enabling preemptive rerouting that avoids emergent hotspots. Third, the fusion strategy is critical: naive averaging of emulator and ML outputs can produce suboptimal or unstable behavior, whereas a context-aware gating mechanism (which prioritizes emulator estimates under high uncertainty, and ML guidance when predictions are confident) yielded the best compromise between responsiveness and foresight.

The work also highlighted practical deployment considerations. Telemetry fidelity and sampling cadence directly impact both emulator accuracy and ML model reliability; excessively sparse or noisy measurements degrade performance, whereas overly frequent sampling increases overhead. A tiered telemetry approach—high-frequency lightweight metrics for emulation and lower-frequency aggregated features for ML—struck an effective balance. Model maintenance

strategies matter as well: periodic offline retraining supplemented by online fine-tuning reduced model drift while preserving stability. Safety constraints and conservative exploration during online learning prevented disruptive routing changes and ensured predictable behavior during model updates.

Despite promising results, several limitations warrant attention. The current implementation assumes reliable and timely telemetry, which may not hold in highly distributed or partially connected environments. The linear emulator, while efficient, can fail to capture non-linear interactions under extreme load or complex queuing disciplines; integrating modest non-linear terms or switching to piecewise linear models could improve fidelity without large overhead increases. Additionally, the ML component's training depends on representative historical data—dramatic topology changes or novel traffic types may reduce model effectiveness until sufficient new data are collected.

Future work should pursue multiple directions. Extending the emulator to incorporate adaptive basis functions or lightweight non-linear components can increase modeling capacity while retaining low latency. Incorporating graph neural networks would enable richer topological awareness and relational feature extraction, which could enhance ML predictions for larger or more complex topologies. Investigating safe reinforcement learning techniques and constrained optimization formulations may allow more aggressive, yet controlled, online adaptation. Finally, broader field trials in production-like SDN deployments and heterogeneous IoT networks will be essential to validate robustness, interoperability with existing control stacks, and operational cost-benefit.

From a systems perspective, the research suggests actionable recommendations for network operators and designers. First, implement a modular control architecture that cleanly separates fast analytic estimation, ML prediction, and fusion logic—this modularity simplifies testing and rollback. Second, adopt a pragmatic telemetry policy that aligns sampling frequency with decision latency requirements while minimizing overhead. Third, enforce safety checks and rate limits on routing changes to avoid oscillations. And fourth, maintain an ongoing

model governance process that includes retraining schedules, validation on holdout traces, and mechanisms for human oversight.

In summary, the linear emulation plus machine learning paradigm presents a practical, scalable route toward more adaptive and efficient routing in heterogeneous networks. By combining rapid local estimations with learned global insights—and by embedding safety and operational pragmatism into the fusion layer—networks can better meet stringent QoS objectives while remaining flexible to changing demands. Continued refinement and real-world validation will further clarify the boundaries of applicability and unlock broader adoption in SDN, edge, and IoT environments.

REFERENCES

1. Al-Shammari, S., & Winston, E. (2020). Machine learning-based routing optimization in software-defined networks. *IEEE Access*, 8, 159241–159255. <https://doi.org/10.1109/ACCESS.2020.3024321>
2. Bedi, P., & Bhadoria, R. S. (2019). Deep learning techniques for routing optimization in wireless sensor networks: A review. *Journal of Network and Computer Applications*, 135, 51–99. <https://doi.org/10.1016/j.jnca.2019.03.021>
3. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
4. Chen, Y., Huang, Q., Yang, Y., & Li, Z. (2021). Reinforcement learning for dynamic routing in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 70(10), 9952–9964. <https://doi.org/10.1109/TVT.2021.3081959>
5. Cui, Y., Gu, D., & Lin, Z. (2018). Emulation and performance evaluation of optimized routing for IoT networks. *Computer Networks*, 135, 96–107. <https://doi.org/10.1016/j.comnet.2018.02.007>