# GNN-LSTM agent approach for Quality of Service routing*

Adrian Płatek[2,4][0009−0009−5463−7852],
Stanisław Kozdrowski[1][0000−0001−6647−5189],
Zbigniew Kopertowski[3][0000−0002−9471−6258], and
Sławomir Sujecki[2][0000−0003−4588−6741]

[1] Department of Computer Science,
Warsaw University of Technology, Nowowiejska 15/19,
[2] Faculty of Electronics, Military University of Technology,
S. Kaliskiego 2, 00-908 Warsaw, Poland,
[3] Orange Innovation Poland, Orange Polska,
[4] `adrian.platek@wat.edu.pl`

**Abstract.** This paper proposes a QoS focused routing algorithm, which relies on a GNN-LSTM agent approach. In order to reproduce main features characteristic of the modeled environment we embed analytical queuing models directly into a SimGrid-based flow-level state space. Thus, the agent accurately estimates native delay, jitter, and packet loss for distinct traffic profiles. Under high-load scenarios, the proposed routing algorithm significantly outperforms classical Shortest Path First algorithms and purely reactive dynamic routing strategies.

**Keywords:** Graph Neural Networks · LSTM · Quality of Service · Dynamic Routing · Network Simulation.

## 1 Introduction

The continuous evolution of global communication networks, driven by the emergence of 5G/6G architectures and the proliferation of bandwidth-hungry, delay-sensitive applications (e.g., 4K/8K video streaming, Voice over IP, autonomous systems, and IoT [6]), has exposed the limitations of traditional network management. The majority of network deployments still rely on variations of classical algorithms like Dijkstra's Shortest Path First (SPF), Open Shortest Path First (OSPF), or Equal-Cost Multi-Path (ECMP). These algorithms are inherently static or rely on simplistic link weights (such as hop count or fixed capacity). Consequently, they react poorly to sudden traffic bursts, often leading to suboptimal routing, micro-burst congestion, and severe degradation of Quality of Service (QoS).

---

* The work on this paper was partially done in the 6G-Cloud project, founded by the European Union's Horizon Europe/SNS JU research innovation program under Grant Agreement No. 101139073.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

The fundamental challenge in dynamic network optimization is managing the dual nature of traffic behavior: spatial dependencies and temporal dynamics. Spatially, congestion on one network link directly impacts the performance of adjacent links due to shared switching resources and cascading buffer overflows. Temporally, traffic patterns fluctuate continuously, exhibiting both short-term bursts and long-term trends.

Traditional routing protocols often rely on static metrics, failing to account for the dynamic, non-linear nature of network traffic. As modern networks transition towards 5G/6G, supporting real-time applications with stringent Quality of Service (QoS) requirements becomes paramount. While recent Machine Learning approaches integrating Graph Neural Networks (GNN) and Long Short-Term Memory (LSTM) effectively model spatio-temporal dynamics, they predominantly operate within a reactive paradigm, optimizing routes based on post-facto state snapshots.

To address these challenges, we propose a novel Spatio-Temporal predictive routing framework. The main contributions of this paper are summarized as follows:

- Proactive State Routing While recent surveys highlight the growing role of Machine Learning in network optimization [11], we shift the paradigm of ML-based routing from reactive forecasting to proactive "what-if" evaluation. We introduce a mechanism where a baseline Spatio-Temporal model (GNN-LSTM) is dynamically fed with counterfactually loaded link states. By simulating the non-linear queuing impact of a specific flow prior to path assignment, our agent actively avoids the “Thundering Herd” problem and prevents micro-burst congestion.
- Simulation-Analytical Hybrid Environment Prevailing ML routing studies typically rely either on static, pre-computed datasets (which fail to capture continuous network feedback) or on discrete-event packet simulators like *ns-3* (which suffer from severe scalability bottlenecks during thousands of training episodes). To bridge this gap, we propose a highly realistic, closed-loop training environment. By embedding analytical M/M/1/K queuing theory and TCP throughput models directly into the SimGrid flow-level state space, we achieve a highly practical balance. The agent continuously experiences native packet-level metrics—such as transient delay, jitter, and bufferbloat-induced packet loss—in real-time, while maintaining the computational efficiency required for dynamic ML training. This makes our framework significantly more representative to real-world network operation than purely theoretical models.
- Multi-Objective QoS Optimization and Stability We formulate a comprehensive reward function inspired by ITU-T E-model recommendations, allowing the routing agent to differentiate between UDP (delay-sensitive) and TCP (throughput-sensitive) flows. Evaluated on the realistic Germany50 [7] topology, our proactive framework successfully mitigates routing instability (route flapping) while substantially outperforming classical SPF and purely reactive dynamic routing strategies.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

## 2 Problem formulation

We model the physical topology as an undirected graph $G = (V, E)$, where $V$ represents the set of forwarding nodes (switches/routers) and $E$ represents the set of communication links. Let $N = |V|$ and $M = |E|$. Each link $e \in E$ is characterized by a static maximum capacity $C_e$ (in bits per second) and a physical propagation delay $D_{prop,e}$ (in seconds). At any given time interval $t$, a link experiences a variable traffic load $\lambda_{e,t}$.

The primary goal of the routing agent is to compute an optimal end-to-end path $P(s, d) = \{e_1, e_2, \ldots, e_k\}$ for a flow originating at source $s \in V$ and terminating at destination $d \in V$, such that the cumulative QoS degradation is minimized.

### 2.1 Analytical Queuing Model for QoS Estimation

A fundamental dilemma in training network routing agents lies in the choice of the simulation engine. Packet-level simulators offer high fidelity but are computationally prohibitive for the iterative training of complex GNN-LSTM architectures. Conversely, flow-level simulators (such as SimGrid) are highly scalable but abstract away individual packets, completely losing critical native QoS metrics such as microburst delay, instantaneous jitter, and probabilistic packet loss.

To resolve this dichotomy and create a highly practical training pipeline, we superimpose analytical queuing theory onto the continuous flow-level simulation. This mathematical fusion allows our agent to perceive a highly realistic network state without the overhead of tracking millions of discrete packets.

We model each link's egress port as an M/M/1/K queuing system. The arrival rate $\lambda$ is derived from the simulated bit load, and the service rate $\mu = C_e/\text{packet\_size}$. The load factor is defined as $\rho = \lambda/\mu$. For a finite buffer of size $K$ packets, the probability of packet loss ($P_{loss}$) is given by:

$$P_{loss} = P_K = \begin{cases} \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} & \text{if } \rho \neq 1 \\ \frac{1}{K+1} & \text{if } \rho = 1 \end{cases} \tag{1}$$

The buffer size $K$ is a critical hyperparameter that dictates the maximum queuing delay. It is derived using the Bandwidth-Delay Product (BDP) principle. To model realistic Active Queue Management (AQM) behavior, we target a queuing delay ($D_{target}$) between 50 ms and 250 ms. Thus, the buffer capacity in packets is approximated as $K = (C_e \times D_{target})/packet\_size$. To prevent TCP starvation while avoiding bufferbloat for real-time traffic, we establish a strict lower bound of $K = 32$ packets.

The average number of packets in the system is $L_{queue}$, from which we calculate the expected queuing delay using Little's law. The total one-way delay $D_e$ for link $e$ is:

$$D_e = D_{prop,e} + \min(D_{queue,e}, D_{cap}) \tag{2}$$

where $D_{cap}$ represents an upper-bound delay limit enforced by the buffer size.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

## 2.2 TCP Throughput Modeling

For TCP flows, we utilize a heuristic adaptation inspired by macroscopic TCP behavior models, such as the one developed by Padhye et al. [8], Since classical analytical models approach infinity when the packet loss probability drops to zero, we introduce a modified bounding equation to ensure that the effective rate smoothly converges to, but never exceeds, the physical link capacity $Rate_{nominal}$:

$$Rate_{effective} = \frac{Rate_{nominal}}{1 + \alpha \cdot RTT \cdot \sqrt{P_{loss}}} \tag{3}$$

where $\alpha \approx 2$, and RTT is double the sum of one-way delays along the path $P$.

## 2.3 QoS Reward Function Formulation

To calculate the optimal path, we formulate a composite reward function $R$. The routing agent minimizes the path cost, which is the sum of link costs along path $P$:

$$Cost(P) = \sum_{e \in P} \Big(\alpha(1 - L_e) + \beta D_e + \gamma J_e + \delta P_{loss,e} + \zeta B_e\Big) + \epsilon I_{reroute} \tag{4}$$

where $L_e$ is the normalized load, $D_e$ is delay, $J_e$ is jitter, $P_{loss,e}$ is the packet loss probability, and $B_e$ represents the normalized bottleneck factor. The indicator variable $I_{reroute} \in \{0, 1\}$ applies a penalty $\epsilon = -0.5$ for route flapping to enforce routing stability.

The weights ($\alpha$, $\beta$, $\gamma$, $\delta$, $\zeta$) are strictly tuned based on the traffic profile. For UDP flows (voice/video), the weights are grounded in the ITU-T E-model (G.107) [4] recommendations, which state that a 1% packet loss degrades voice quality equivalently to 35–40 ms of additional delay. Consequently, equating $\delta \cdot 0.01 \approx \beta \cdot 0.04$, we assign $\delta = -1.0$ and $\beta = -0.25$. Because jitter severely impacts video streaming perception, it is heavily penalized with $\gamma = -0.5$. The remaining UDP weights are $\alpha = 1.0$ and $\zeta = 0.5$.

Conversely, for TCP flows (bulk data transfers), the agent prioritizes throughput and bottleneck avoidance over strict delay constraints, since TCP inherently handles packet loss via retransmissions. The TCP profile weights are adjusted to: $\alpha = 1.5$, $\beta = -0.1$, $\gamma = -0.1$, $\delta = -0.5$, and $\zeta = 1.0$.

The formulation of our comprehensive reward function aligns with recent advancements, where Machine Learning techniques are increasingly leveraged to proactively detect degradation in QoS parameters, particularly for demanding multimedia services such as video streaming. Furthermore, robust ML-based prediction of the Quality of Transmission (QoT) has proven highly effective in core optical networks [5], underscoring the critical need for unified, multi-objective metrics in intelligent routing. By integrating delay, jitter, and packet loss into a single composite reward, our routing agent is explicitly trained to maintain these stringent, ML-predictable quality thresholds for diverse traffic profiles."

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

## 3 Methods

To predict the optimal weights for Equation 4, we propose an architecture that maps the network into a latent space, processes the topological structure, and analyzes time-series trends.

Standard GNNs [3] operate on nodes, but critical routing metrics reside on links, a challenge similarly addressed by foundational models like RouteNet [9]. We apply a Line Graph Transformation, creating $L(G) = (V_L, E_L)$, where $V_L = E$. An edge $(e_i, e_j) \in E_L$ exists if links $e_i$ and $e_j$ share a common physical switch. The input feature matrix for the GNN at time $t$ is $X_t \in \mathbb{R}^{M \times F}$.

To capture temporal dynamics and advance toward intent-based network automation [1], a sequence of spatial embeddings over a lookback window $W$ is fed into an LSTM network. The LSTM cell states are governed by standard gating mechanisms. The final hidden state $h_t$ encapsulates both the topological congestion context and the temporal trajectory of the traffic load on that specific link. The data processing pipeline underlying our predictive framework is depicted in Fig. 1a, illustrating the systematic transformation of raw network data into high-level Quality of Service (QoS) metrics.

Unlike traditional predictive models that compute routing paths based on a reactive snapshot of current or historical congestion, our framework employs a Prospective State mechanism. This approach leverages counterfactual reasoning — evaluating "what-if" scenarios before actual flow assignment.

When a new traffic flow $f$ with an estimated bandwidth demand $\lambda_f$ arrives, the agent computes a set of candidate $k$-shortest paths $\mathcal{P} = \{P_1, P_2, ..., P_k\}$ between the source and destination. For each candidate path $P_c \in \mathcal{P}$, the controller simulates a counterfactual state update. Specifically, the load feature of every link $e \in P_c$ is virtually incremented:

$$\lambda_{e,t}^{prospect}(P_c) = \lambda_{e,t} + \lambda_f \quad \forall e \in P_c \tag{5}$$

This temporary load injection allows the analytical M/M/1/K model to re-evaluate the prospective utilization, $\rho^{prospect}$, and the expected queuing delays. The modified input feature matrix $X_t^{prospect}$ is then passed through the GNN-LSTM pipeline.

The Readout layer, a Multi-Layer Perceptron (MLP), maps the resulting prospective hidden state $h_t^{prospect}$ to a predicted dynamic weight for each link:

$$\hat{w}_{e,t}(P_c) = MLP_{readout}(h_t^{prospect}) \tag{6}$$

These weights represent the model's estimate of the link's future congestion *if* the flow $f$ were routed through $P_c$. The agent evaluates the composite reward function $R$ (Equation 5) for each prospective path, directly incorporating the routing penalty $\epsilon I_{reroute}$ to prevent flow oscillation [10]. The optimal path $P^*$ is selected as the one that minimizes the prospective cumulative cost:

$$P^* = \arg\min_{P_c \in \mathcal{P}} \sum_{e \in P_c} \hat{w}_{e,t}(P_c) \tag{7}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

(a) GNN model training pipeline.

(b) Germany50 network.

Fig. 1: GNN model and analyzed network topology.

By predicting the localized impact of its own decisions before execution, the agent proactively avoids assigning flows to paths that are currently underutilized but would succumb to microburst congestion upon the flow's arrival.

## 4 Early Results and Discussion

This Section presents preliminary research results, describes the simulation environment, and compares the proposed algorithm with those known from the literature. The framework is instantiated using Python, integrating the SimGrid API [2] for network simulation and PyTorch for the GNN-LSTM model. We deployed the Ger50 topology from the SNDlib library (50 nodes, 89 bidirectional links) [7]. The structural complexity and spatial diversity of the evaluated environment are depicted in Fig. 1b. Link capacities were dynamically assigned between 1 Gbps and 10 Gbps. Background traffic was generated using randomized demand matrices (70% TCP flows, 30% UDP flows). The network state monitor operates at an interval of 20–50 ms to ensure rapid detection of traffic microbursts. The model was trained offline using a supervised learning paradigm, with a loss function combining Mean Squared Error (MSE) for metric predictions and a custom Routing Loss. Table 1 presents the detailed configurations.

### 4.1 Baseline Algorithms and Performance

We compared the GNN-LSTM agent against two standard routing strategies: OSPF (static link capacities) and Floyd-Warshall (FW) (dynamic recalculation based on instantaneous delay without memory). The models were evaluated under a high-load scenario where aggregate traffic demand exceeded 80% of the core network's capacity.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

Table 1: Model and Simulation Hyperparameters

| **Parameter** | **Value** |
|---|---|
| Monitor Interval | 20–50 ms |
| GNN Message Passing Steps ($K$) | 2 |
| LSTM Sequence Length ($W$) | 100 time steps |
| Optimizer | Adam ($1 \times 10^{-3}$ with Weight Decay) |
| M/M/1/K Buffer Size ($K$) | 32 packets (BDP 50–250 ms) |
| Convergence Penalty Time | 50 ms |

As observed in Table 2, static OSPF performs poorly under congestion, causing massive buffer overflows and resulting in 0.013% packet loss and severe jitter (165.69 ms). The reactive Floyd-Warshall approach mitigates some congestion. In contrast, the GNN-LSTM model leverages its spatial-temporal memory to anticipate congestion before buffers overflow. The network dynamics under high-load conditions are further detailed in Fig. 2, which presents the evolution of average jitter over the simulation time.

Table 2: Performance Comparison Under High-Load Scenarios

| **Metric** | **OSPF (Static)** | **FW (Dynamic)** | **GNN-LSTM** |
|---|---|---|---|
| Avg. UDP E2E Delay (ms) | 82.60 | 78.51 | **74.55** |
| Avg. UDP Jitter (ms) | 165.69 | 120.55 | **103.34** |
| UDP Packet Loss (%) | 0.013 | 0.012 | **0.0088** |
| Avg. TCP Goodput (Gbps) | 3.86 | 4.28 | **4.65** |

This chart visually confirms that while the baseline methods show progressive performance degradation, the proposed model, after the initial learning period, sustains significantly lower jitter levels. It successfully drops UDP packet loss to a highly acceptable 0.0088% and jitter to 103.34 ms. The LSTM's predictive nature enables the model to differentiate between transient microbursts and sustained traffic shifts, yielding the highest TCP throughput (4.65 Gbps) while protecting delay-sensitive traffic. In our implementation, we restrict the number of message-passing steps to K = 2. Empirical evaluations indicated that for the Germany50 line graph, a 2-hop receptive field is sufficient to capture regional congestion bottlenecks. Furthermore, K = 2 reduces the computational overhead, which is critical for real-time agent inference.

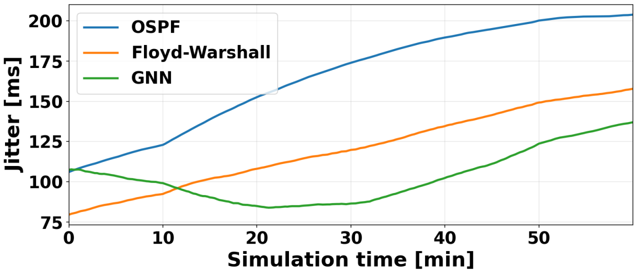

Fig. 2: Evolution of average jitter over simulation time.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10

## 5 Conclusions

In this paper, we propose a Spatio-Temporal predictive routing framework for IP networks that integrates GNN and LSTM models. The integration of analytical M/M/1/K queuing models into the SimGrid environment enabled accurate flow-level simulation of delay, jitter, and packet loss. For the Ger50 topology, the proposed GNN-LSTM agent routing algorithm demonstrated superior performance over classical OSPF and dynamic reactive algorithms. In particular, the proposed GNN-LSTM agent routing algorithm minimized QoS degradation for real-time UDP traffic, enhanced TCP throughput, and preserved routing stability.

## References

1. Ahmed Khan, T., Abbas, K., Diaz Rivera, J.J., Muhammad, A., Song, W.c.: Applying routenet and lstm to achieve network automation: An intent-based networking approach. In: 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS). pp. 254–257 (2021). https://doi.org/10.23919/APNOMS52696.2021.9562499
2. Casanova, H., et al.: Versatile, scalable, and accurate simulation of distributed applications and platforms. Journal of Parallel and Distributed Computing **74**(10), 2899–2917 (2014). https://doi.org/10.1016/j.jpdc.2014.06.008
3. Dhamala, B.K., Dawadi, B.R., Manzoni, P., Acharya, B.K.: Performance evaluation of graph neural network-based RouteNet model with attention mechanism. Future Internet **16**(4), 116 (2024). https://doi.org/10.3390/fi16040116
4. ITU-T: Recommendation G.107: The E-model: a computational model for use in transmission planning. Tech. rep., International Telecom. Union, Geneva (2015)
5. Kozdrowski, S., Cichosz, P., Paziewski, P., Sujecki, S.: Machine learning algorithms for prediction of the quality of transmission in optical networks. Entropy **23**(1), 7 (2021). https://doi.org/10.3390/e23010007
6. Kumar, L., Sharma, M., Singh, A.: Mitigating congestion in iot-driven sdn environments: An intelligent traffic-aware flow optimization approach. In: 2025 IEEE 6th India Council International Subsections Conference (INDISCON). pp. 1–6 (2025). https://doi.org/10.1109/INDISCON66021.2025.11253719
7. Orlowski, S., Wessäly, R., Pióro, M., Tomaszewski, A.: Survivable network design library. Networks **55**(3), 276–286 (2010). https://doi.org/10.1002/net.20371
8. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP Reno performance: a simple model and its empirical validation. IEEE/ACM Transactions on Networking **8**(2), 133–145 (2000). https://doi.org/10.1109/90.842137
9. Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., Cabellos-Aparicio, A.: RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN. IEEE Journal on Selected Areas in Communications **38**(10), 2260–2270 (2020). https://doi.org/10.1109/JSAC.2020.3000405
10. Song, Y., Qian, X., Zhang, N., Wang, W., Xiong, A.: QoS routing optimization based on deep reinforcement learning in SDN. Computers, Materials and Continua **79**(2), 3007–3021 (2024). https://doi.org/10.32604/cmc.2024.051217
11. Sun, M., He, B., Li, R., Li, J., Zhang, X.: A survey: Network feature measurement based on machine learning. Applied Sciences **13**(4) (2023). https://doi.org/10.3390/app13042551

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_10