International Journal of Safety and Security Engineering
Vol. 16, No. 7, July, 2026, pp. 1657-1669
Journal homepage: http://iieta.org/journals/ijsse

Adaptive Game-Theoretic Routing for Software-Defined Networks under Single-Link Failures: Performance Analysis and Trade-Offs

Karrepu Sreeveda*, Kumar Babu Batta

Department of Computer Science and Engineering, GITAM Deemed to be University, Hyderabad 502329, India

Corresponding Author Email: skarrepu@gitam.in

https://doi.org/10.18280/ijsse.160720

Received: 16 June 2026
Revised: 22 July 2026
Accepted: 26 July 2026
Available online: 31 July 2026

Keywords:
software-defined networking, game-theoretic routing, single-link failure recovery, potential games, priority-aware routing, latency optimisation

ABSTRACT

Network link failures degrade performance in software-defined networks (SDNs), motivating adaptive routing. This paper presents a comparative evaluation of three routing paradigms under identical conditions: static shortest-path routing, an offline decision-tree (DT) baseline representing the congestion-blind supervised-learning class, and a game-theoretic (GT) routing algorithm that jointly optimises latency, congestion, and flow priority through best-response dynamics. Across 30 Monte Carlo simulations on three 11-node topologies—Abilene, Erdos–Rényi, and Barabási–Albert—under″ isolated single-link failures, GT reduces mean latency by 11.1% over static routing (20.44 ± 1.27 ms vs. 22.99 ± 1.23 ms) and by 3.2% over the DT baseline (21.12 ± 1.28 ms), and reduces 95th-percentile latency by 14.0%. Because the model imposes no hard link capacity, every admitted flow is delivered: the delivery ratio is ≈ 99.96% and the delivered-flow rate ≈ 4.5 flows/s for all three routers, so throughput does not differentiate the policies and GT's advantage lies wholly in latency and load distribution. Two robustness studies bound the result: under heterogeneous link capacities with background traffic the GT margin widens to 42.6%, whereas under double-link and correlated failures it narrows to approximately 7%. Two trade-offs are quantified—latency versus convergence overhead, and optimality versus adaptivity—supported by a component ablation and by a proof that the game is an exact potential game whose best-response dynamics converge to a pure Nash equilibrium. The method's design capabilities are contrasted qualitatively with recent deep-learning and failure-recovery approaches; no cross-paper performance claim is made. Application scenarios expected to benefit are identified, subject to larger-scale validation.

1. INTRODUCTION

1.1 Motivation

Modern critical infrastructure increasingly relies on high performance, low-latency networks, with software-defined networking (SDN) emerging as the de-facto paradigm for 5G and emerging 6G systems [1, 2]. SDN's centralised control plane enables programmable traffic engineering, but also introduces new vulnerability modes: a single link failure can propagate congestion across multiple paths, degrading quality of service (QoS) in conventional static routing deployments [3]. Applications such as industrial IoT, autonomous vehicle coordination, and emergency communications impose strict latency budgets that cannot tolerate such degradation [4].

Existing routing protocols address failure recovery at various layers. Traditional interior gateway protocols (e.g., OSPF with Fast Reroute) react to failures within seconds, which is inadequate for ultra-reliable low-latency communication (URLLC). Equal-Cost Multi-Path (ECMP) routing distributes load across shortest paths but is oblivious to congestion and flow priority. Machine learning approaches have demonstrated offline path-selection improvements but struggle with real-time congestion adaptation. Game-theoretic (GT) methods offer coordination guarantees but have not been evaluated in unified failure benchmarks alongside their alternatives.

Scope statement. All results in this paper are obtained by discrete-event simulation on 11-node topologies under isolated single-link failures. They characterise algorithmic behaviour, not deployment performance; larger-scale emulation or physical-testbed validation would be required before any operational claim.

1.2 Problem statement and research gaps

Despite extensive individual study of static, machine learning (ML)-based, and GT routing, no unified comparative framework evaluates all three paradigms under identical failure conditions with priority-aware traffic and a shared candidate-path set. Three gaps are identified:

•Cross-paradigm benchmarking: most works evaluate a single routing family in isolation, precluding direct comparison.

•Priority-aware failure recovery: mechanisms that simultaneously honour flow priorities and recover from topology failures remain underexplored.

•Controlled path availability: comparisons rarely state how

candidate paths are generated or whether competing routers receive the same set, confounding path availability with path selection.

1.3 Contributions

Congestion-aware GT routing algorithm. We formulate routing as a non-cooperative game, prove it is an exact potential game admitting a pure-strategy Nash equilibrium reachable by finite improvement (Lemma 1, Theorem 1), and bound its inefficiency (Theorem 2).

Unified comparative evaluation. We benchmark static, offline decision-tree (DT), and GT routing under identical single-link failure scenarios across three topology families using an explicitly specified shared candidate-path set complemented by a component ablation and by robustness studies under heterogeneous capacities and multi-link failures.

Analytical characterisation. We prove the routing game is an exact potential game with a pure Nash equilibrium reached by finite best-response dynamics, and identify a representational (not quantified) limitation of offline supervised path classifiers over static features. This limitation concerns congestion-blind classifiers and does not bound GNN or deep-RL routers that observe live network state.

Algorithm-selection framework. We derive a decision rule (Eq. (13)) partitioning the (latency-sensitivity, failure-frequency) plane into regions favouring each paradigm. Its parameters must be measured per deployment; we make no performance claim for the hybrid architecture, which is proposed but not evaluated.

Open-source platform. All code, topologies, and analysis scripts are publicly released for reproducibility.

1.4 Paper organisation

Section 2 surveys related work. Section 3 formalises the system model and algorithm designs. Section 4 describes the framework and simulation setup. Section 5 presents results, including two robustness studies. Section 6 analyses trade-offs and limitations. Section 7 gives analytical justification with proofs, and Section 8 concludes.

2. RELATED WORK

Classical shortest-path algorithms Dijkstra and Bellman–Ford for in-network computation of shortest paths are typically used within the framework of link-state routing protocols (e.g., OSPF, IS-IS) for a static set of routes. Yet, in the framework of an SDN controller, these functions can be moved to the controller in a fully centralized manner, and on the basis of complete knowledge of the network topology. In our preceding work, we had noted, however, that static path selection of shortest paths is not very efficient for failures of links. This is because the path chosen by the controller by means of a shortest path algorithm, will not change until the controller has detected the failure in the network topology, and selected an alternative route. Surveying fast-recovery in packet-switched networks, Chiesa et al. [3] observed that fast data-plane recovery and fast control-plane recovery are both needed. Thus, flow distribution by means of ECMP [5] (equal cost multi-path) is meant to distribute flows on equal cost paths, yet it is completely agnostic to per-link congestion as well as to per-link flow priorities.

Segment routing is a traffic-engineering architecture but live congestion-feedback is not implemented [6]. Many approaches have been proposed for routing in computer networks, including offline supervised learning and online reinforcement learning (RL) to name a few. In reference to learning to manage routing, RouteNet [7] uses message-passing graph neural networks (GNNs) to predict the per-flow QoS metrics of a given network.

New generation of DRL routers, learn and adapt on-the-fly in order to improve routing decisions based on real-time information. Recently, a topology-agnostic performance model for computer networks has been proposed [8] and already generalized to arbitrary topologies.

Baseline scoping: Our simple DT classifier serves as a controlled lower bound for supervised learning in computer networks, and not a state-of-the-art representation. Our simple supervised learning approach is designed to be as simple as possible, i.e., imitate the GT path choice. The simple supervised learning approach in turn receives only a set of static features (i.e., per-candidate base latency, path length, flow priority) and, importantly, no live congestion information. Measured GT–DT gap, therefore, is a single-variable experiment that specifically studies the value of online congestion information and of further, repeated refinements. In summary, our simple supervised approach is far from being a comparison against the GNN models of RouteNet, or against deep-RL routers for future networks. A same-benchmark study would be required for such future work, and is not attempted in this paper. We refer to our simple supervised approach as simply “DT” (instead of “ML”), throughout the paper’s experimental results.

The routing in a network can be modeled by a non-cooperative game in which flows try to minimize their individual costs. In continuous traffic models the flows that minima their individual costs will form a Wardrop equilibrium. In discrete flow models the flow will try to find a Nash equilibrium, i.e. the flow will try to find a strategy for which no improve can be found by changing to another strategy. In this paper we model the GT path choice by a potential game. We extend the potential game of the flow routing in networks by means of embedding flow priority into the per-flow utility. We show that the game of the GT path choice embeds an exact potential.

Low-disturbance traffic engineering using a combination of reinforcement learning and linear programming has been proposed to manage traffic in dynamic networks [9]. Several fundamental challenges in traffic engineering for the future Internet, including flow management, fault tolerance, topology updates, and traffic analysis, have also been identified [10]. Neither of these studies, however, compares or contrasts game-theoretic-based routing with simple static as well as supervised approaches under specific failure scenarios.

In large distributed systems, the tail of the distribution of operation times can dominate the average response time. Therefore, for example, the study of congestion control in highly concentrated utilisation networks [11] can exhibit severe latency tails. A number of architectures for future-network domains have been proposed for routing between them, including a hybrid hierarchical architecture for SDN [12]. The above studies deal with similar issues to this paper, i.e. issues of tail latency, topology dependent and failure. However, none of them consider and compare simple (i.e. static) routing, supervised learning based routing and game-theoretic based routing within a controlled benchmark.

More recent studies have explored more adaptive and congestion-aware routing in SDN. Deep reinforcement learning has been shown to learn routing in a given network [13]. However, such a routing policy needs to be trained first. From a more theoretical point of view, proportional-response dynamics in Fisher markets have been studied [14], showing that, by means of a series of repeated local adjustments, a set of agents competing for shared resources can eventually coordinate their actions. The use of game theory in telecommunications has also been surveyed [15], covering topics such as routing, congestion control, pricing, and distributed resource allocation in terms of utilities and equilibrium.

Thus, in our studies, we have proposed the use of a training-free game-theoretic framework. In such a framework, the flows iteratively select a path in the underlying network, taking into account two kinds of costs, namely, the latency that a given flow would incur as it traveled along a given path, and the (variable) congestion that the given link would experience as a function of the (variable) loads that are placed on it by the other flows. Our performance metric, i.e., the mean latency and the 95th-percentile latency, takes into account the tail of the latency distributions experienced by the different flows, because a single, late, packet can end up dominating the (perceived) end-to-end delay of an application [16]. Thus, for example, by measuring the (mean and the 95th-percentile) end-to-end delay of a given cluster of Distributed In-Memory Database Servers as they communicate with each other by means of the paths in the underlying network assigned to them by our proposed framework, we have shown that our proposed congestion-aware framework can significantly improve such latency numbers, especially in more Loaded-out scenarios. Other related studies have examined requirements of SDN traffic-engineering, such as traffic measurement, topology awareness, load balancing, and fault handling [17]. Thus, in our studies, in addition to using live link-load feedback, we have also used dynamic path re-optimisation following a change in congestion or a failure of a link.

3. SYSTEM MODEL AND ALGORITHM DESIGN

3.1 Network model

The network is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \dots, v_N\}$ the set of N nodes and E the link set. The system evolves in discrete time slots $t \in \mathcal{T} = \{1, \dots, T\}$.

(1) Latency model

Each link $(u, v) \in \mathcal{E}$ has base latency $b_{(u,v)} \sim \mathcal{U}(1,10)$ ms modelling propagation delay. The instantaneous link latency is

$$\ell_{(u,v)}(t) = b_{(u,v)} + \eta f_{(u,v)}(t) + \varepsilon_{(u,v)}(t) \tag{1}$$

where, $f_{(u,v)}(t)$ is the number of active flows on the link, η = 0.5 ms/flow is the congestion coefficient, and $\mathcal{E}_{(u,v)}(t) \sim \mathcal{N}(0, \sigma^2)$ with σ = 0.5 ms.

Eq. (1) is valid in the low-to-moderate utilisation regime with homogeneous link capacities. Because linearity and homogeneity materially affect the conclusions, they are tested directly rather than merely discussed: Section 5 repeats the full comparison under heterogeneous capacities with background traffic and a nonlinear near-saturation term. Absolute latency values reported under Eq. (1) are model-specific and should not be read as deployment predictions.

(2) Flow model

Each flow $i \in \mathcal{F}(t)$ is characterised by $\phi_i = (s_i, d_i, P_i, \tau_i^{arr}, \tau_i^{dur})$, where $s_i, d_i \in \mathcal{V}$ are source and destination, $P_i \in \{1(\text{high}), 2(\text{low})\}$ is priority, and $\tau_i^{arr}, \tau_i^{dur}$ are arrival time and duration. Arrivals are Poisson:

$$|\mathcal{F}_{new}(t)| \sim Poisson(\lambda), Pr(P_i = 1) = 0.6 \tag{2}$$

with λ = 5 flows/sec fixed throughout to isolate failure-recovery behaviour from arrival-rate effects, and $\tau_i^{dur} \sim \mathcal{U}(5,15)$ s.

3.2 Multi-objective utility function

Path selection for flow *i* is posed as

$$\underset{\mathcal{P}_i}{maximise}\, \mathcal{U}_i(t) = \omega_1 \mathcal{R}_i + \omega_2 \mathcal{Q}_i - \omega_3 \mathcal{D}_i \tag{3}$$

where, $\mathcal{R}_i$ is a priority-based reward, $\mathcal{Q}_i$ quantifies fairness, and $\mathcal{D}_i$ is path delay. Under linear flow-latency, additive path metrics, and linear priority rewards this reduces to

$$\mathcal{U}_i(t) = -\alpha L_i(t) - \beta C_i(t) + \gamma P_i \tag{4}$$

with $L_i(t) = \sum_{(u,v)\in\mathcal{P}_i} \ell_{(u,v)}(t)$ the cumulative path latency and $C_i(t) = \sum_{(u,v)\in\mathcal{P}_i} f_{(u,v)}(t)$ the path congestion intensity. The term γP_i is path-independent for a given flow: it does not enter that flow's ranking of its own candidate paths and only sets the *order* in which flows exercise best response (high priority first). We therefore make no claim that priority produces a latency separation between classes; in our experiments it does not, and the update order is retained solely as a deterministic tie-break. A mechanism that genuinely differentiates priority classes (e.g. priority-weighted congestion cost or capacity reservation) is left to future work.

Weight sensitivity: The GT algorithm was evaluated over $\alpha \in [0.5,2.0]$, $\beta \in [0.4,1.6]$, $\gamma \in [0.6,2.4]$. Mean latency remained within ±8% of the nominal result, supporting the chosen values (α = 1.0, β = 0.8, γ = 1.2).

3.3 Routing algorithms

All three routers select from the same candidate set; they differ only in the selection rule.

(1) Static shortest-path routing

Static routing selects the minimum-base-latency member of the candidate set,

$$\mathcal{P}_i^{static} = \arg\min_{\mathcal{P}\in\mathbb{P}_i} \sum_{(u,v)\in\mathcal{P}} b_{(u,v)} \tag{5}$$

and does not adapt to congestion.

(2) Offline DT baseline

The DT baseline is a depth-8 classifier trained on 45 independent episodes drawn evenly from all three topology families (Abilene, Erdos–Rényi, and Barabási–Albert), in which the GT algorithm's chosen candidate index is the label, so it is not specialised to a single topology in the pooled evaluation. Its feature vector x_i comprises the three candidates' base latencies, their hop counts, and the flow priority—all static quantities. It selects

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{arg\,max}\, Pr(k \mid \boldsymbol{x}_i) \tag{6}$$

falling back to the next-ranked valid candidate if $\hat{y}$ is invalidated by a failure. As stated, GNN and RL baselines are out of scope.

(3) GT routing

Routing is modelled as a non-cooperative game $\Gamma = \langle \mathcal{F}, \{\mathbb{P}_i\}, \{\mathcal{U}_i\} \rangle$ in which each flow selects $\mathcal{P}_i \in \mathbb{P}_i$ to maximise Eq. (4). The algorithm implements iterative best response:

•Initialise $\mathcal{P}_i^{(0)}$ for all $i \in F$.

•For $k = 1,\ldots,K_{\max}$: update link latencies via Eq. (1); then, for each flow i in order of decreasing priority,

$$\mathcal{P}_i^{(k)} \leftarrow \underset{\mathcal{P}\in\mathbb{P}_i}{\arg\max}\, \mathcal{U}_i\left(\mathcal{P}, \mathcal{P}_{-i}^{(k-1)}\right). \tag{7}$$

•Terminate when no flow changes path, or at $k = K_{\max}$.

Proposition 1 (Nash equilibrium existence). The game Γ admits a pure-strategy Nash equilibrium.

Proof. Each flow's action set $\mathbb{P}_i$ is finite and its cost is a sum of per-edge congestion costs, so Γ is a finite congestion game. By Lemma 1 it admits an exact potential Φ; a profile minimising Φ over the finite action space exists and is, by the finite improvement property, a pure-strategy Nash equilibrium (Theorem 1).

Remark. Existence follows from the potential-game structure rather than from fixed-point arguments over convex strategy spaces, which do not apply to a finite path set. The game may admit multiple equilibria; the empirical spread across runs is 4.04 ± 0.05 best-response sweeps, indicating practical stability.

4. PROPOSED ROUTING FRAMEWORK AND SIMULATION SETUP

4.1 Framework architecture

Flow arrivals enter a priority engine that classifies each flow. The routing optimiser evaluates the utility in Eq. (4) over the candidate set, and the SDN controller installs the selected path. Under link failure the controller detects the topology change, removes the failed link from E, recomputes affected candidate sets, and triggers a re-optimisation round.

Table 1 compares the three routing strategies in terms of computational complexity, congestion awareness, priority handling, and failure response. The comparison clarifies that the proposed GT method uniquely combines online load awareness, explicit priority support, and dynamic re-optimisation.

Table 1. Algorithmic comparison across complexity and capability dimensions ($|\mathcal{V}|$ nodes, $|\mathcal{F}|$ flows, d tree depth, $K_{\max}$ iterations)

Algorithm	Routing Complexity	Congestion-Aware	Priority-Aware	Failure Handling
Static SP	$\theta(1)$ selection per flow	No	No	Reactive reselection
DT baseline	$\theta(d)$ inference per flow	Offline only	Partial	Learned, single pass
GT (proposed)	$\theta\left(K_{(max)}\lvert\mathcal{F}\rvert K\right)$ per round	Yes	Yes	Dynamic re-optimisation

Note: All three routers draw from the same candidate set. DT = decision-tree, GT = game-theoretic.

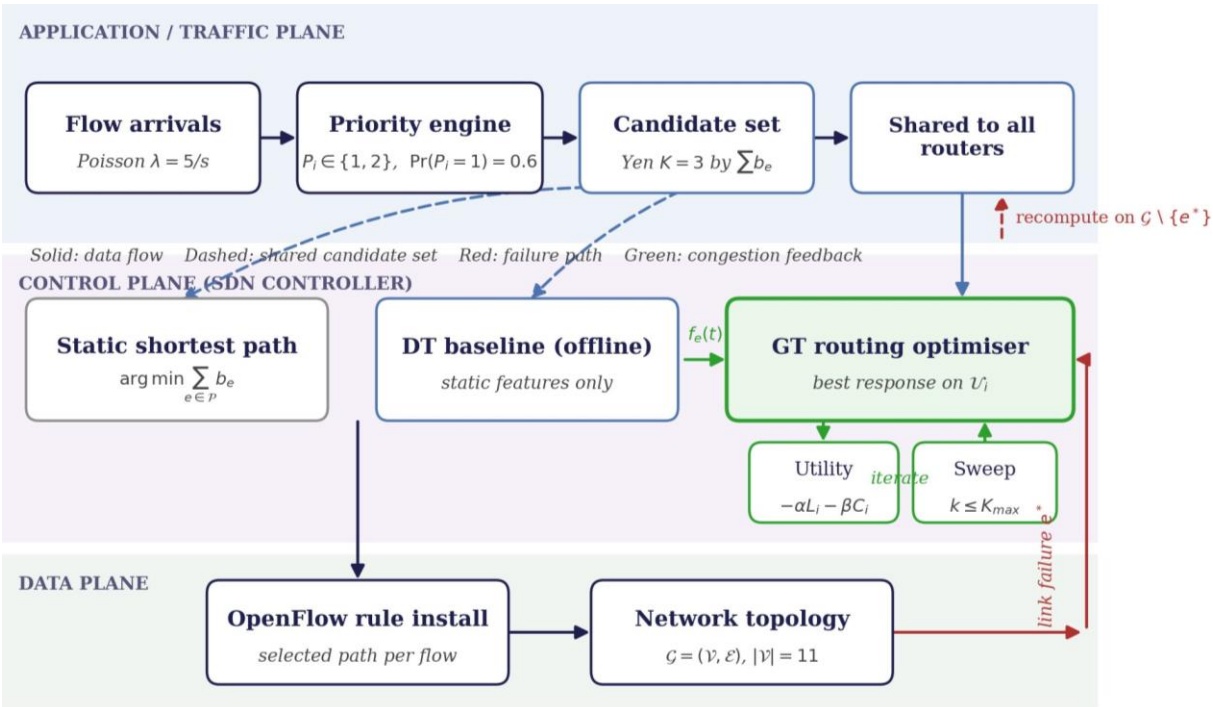

Figure 1. Architecture of the proposed routing framework

Note: Flow arrivals are classified by the priority engine, and a single candidate set of $K = 3$ paths per source–destination pair is generated by Yen's algorithm and supplied identically to all three routers (dashed blue), so that the comparison isolates path selection from path availability. The GT optimiser iterates best response over the utility of Eq. (4) using live link loads $f_e(t)$ fed back from the data plane (green). On failure of link e^* the controller removes e^* from E, recomputes affected candidate sets on $\mathcal{G} \setminus \{e^*\}$, and triggers a re-optimisation round (red).

Figure 1 illustrates the complete control and feedback architecture of the proposed routing framework. All three routing methods receive the same set of K = 3 candidate paths generated using Yen's algorithm, ensuring that their comparison reflects differences in path-selection strategy rather than path availability. Unlike static and DT routing, the GT optimiser uses current link-load feedback from the data plane and iteratively updates flow assignments. When a link fails, the controller removes the failed edge, reconstructs the affected candidate sets, and initiates a new optimisation round.

4.2 Network topologies and candidate path set construction

Three 11-node topologies represent distinct connectivity patterns:

Abilene (11 nodes, 14 links): the US academic backbone, a structured sparse topology. $Erdős - Rényi\ \mathcal{G}_{\mathrm{ER}}(11,0.3)$: random sparse graphs, resampled until connected. Barabási–Albert $\mathcal{G}_{BA}(11,2)$: scale-free graphs with hub nodes. Scaling behaviour beyond $|\mathcal{V}| = 11$ was not measured and no claim is made about it.

For each source–destination pair (*s, d*) we precompute $K = 3$ candidate paths using Yen's algorithm for the K loopless shortest simple paths [18], ranked by cumulative base latency $\sum b_{(u,v)}$. Paths are *not* constrained to be edge-disjoint: on sparse topologies such as Abilene an edge-disjointness constraint frequently admits fewer than three feasible paths, which would confound path availability with routing policy.

The identical candidate set is supplied to all three routers, so the comparison isolates path *selection* from path availability. The routers differ only in how they select within this set: static takes the minimum-base-latency member; the DT baseline takes the highest-scoring valid member under its classifier; GT takes the best response under Eq. (4).

On failure at t_{fail}, the failed edge e^* is removed from E, every active flow whose installed path traverses e^* is marked affected, and the candidate set of each affected pair is recomputed on $\mathcal{G} \setminus \{e^*\}$ by the same procedure. A flow is counted unrecovered only if no simple path survives—that is, only if the failure disconnects its (*s, d*) pair. Recomputation is applied identically to all three routers; failing to do so inflates the apparent recovery advantage of adaptive methods.

4.2.1 Traffic and failure models

Flow arrivals follow a Poisson process with $\lambda = 5$ flows/sec; durations are $\mathcal{U}(5,15)$ s; source–destination pairs are drawn uniformly. A single link is removed at t_{fail}=100 s, $e^* \sim \mathcal{U}(\mathcal{E})$.

Section 5 additionally evaluates double-link and correlated (shared-node) failures.

4.2.2 Evaluation metrics

Five metrics are recorded: (1) mean latency L (ms); (2) 95th-percentile latency L_{95} (ms); (3) the delivered-flow rate; (4) the delivery ratio; and (5) the path-diversity index $D = \left|\bigcup_{(i)} \mathcal{P}_i\right|/|\mathcal{E}|$. GT additionally reports best-response sweeps to convergence.

Two distinct delivery metrics. We report two related but mathematically distinct quantities. The delivery ratio is the dimensionless fraction of admitted flows that retain a valid path for their whole lifetime,

$$S = \frac{|\{i: \mathcal{P}_i \neq \emptyset throughout\}|}{|\mathcal{F}|} \tag{8}$$

and the delivered-flow rate is corresponding count per unit time,

$$\bar{T} = \frac{|\{i: \mathcal{P}_i \neq \emptyset throughout\}|}{T} = S\lambda_{eff} \tag{9}$$

where, $\lambda_{eff} = |\mathcal{F}|/T \leq \lambda$ is the effective admitted-arrival rate (below λ because $s = d$ pairs are discarded). The two are linked by the identity $\bar{T} = S\lambda_{\mathrm{eff}}$ and are not independent measurements.

Mechanism, and why throughput does not differentiate here.

A structural property of the model must be stated plainly. Link latency in Eq. (1) grows with load, but there is no hard capacity and no explicit dropping. A flow is lost only if the failure disconnects its (*s, d*) pair, i.e. only if no simple path survives in $\mathcal{G} \setminus \{e^*\}$—an event essentially independent of the routing policy on a connected 11-node graph. Consequently $S \approx 99.96\%$ and $\bar{T} \approx 4.5$ flows/s for all three routers, and routing has no mechanism by which to alter the delivered-flow rate. We report both metrics only to confirm that no policy induces spurious loss; throughput is therefore not claimed as an advantage of GT. A delivered-flow-rate difference between routers would require a hard capacity constraint with admission blocking, which this model does not impose and which we identify as future work.

4.2.3 Parameter settings

$\alpha = 1.0$, $\beta = 0.8$, $\gamma = 1.2$, $K = 3$ candidate paths, $K_{\max} = 5$ best response sweeps (the convergence analysis shows three suffice; five is retained as a conservative bound), $T = 300$ s, $\lambda = 5$ flows/s, $t_{\mathrm{fail}} = 100$ s, and 30 independent Monte Carlo runs per topology.

5. RESULTS AND ANALYSIS

5.1 Quantitative performance summary

Table 2 summarises the pooled performance results across all three evaluated topologies. The proposed GT router achieves the lowest mean and 95th-percentile latency while preserving essentially the same delivery ratio as the competing methods. Figure 2 complements these numerical results by showing the pooled per-flow latency distributions for the three routing methods. In addition to the lower mean latency reported in Table 2, GT routing exhibits the lowest median latency and by far the narrowest interquartile range (IQR), indicating that its latency advantage extends across the distribution, including the upper range. This behavior is consistent with the load-balancing property established in Lemma 2 and the subsequent Corollary, which state that reducing the number of flows on a heavily loaded link decreases the per-flow latency experienced by the remaining flows. Overall, the results indicate that GT routing provides both lower latency and greater consistency in per-flow latency performance.

GT reduces mean latency by 11.1% relative to static routing and by 3.2% relative to the DT baseline, and reduces 95thpercentile latency by 14.0% and 5.3% respectively. The delivery ratio (≈ 99.96%) and delivered-flow rate (≈ 4.5 flows/s) are near-identical across all three routers: as established above and by Eq. (9), with no hard capacity every admitted flow is delivered irrespective of routing, so these two

metrics do not differentiate the policies and are excluded from GT's claimed advantages. Figure 3 normalises the differentiating metrics onto a common scale.

Figure 4 shows for the three topologies the results for GT compared to static routing. Here the mean-latency reduction for GT over static is for Abilene 9.6%, for ER 13.5% and for BA 14.4%. The reason for the Abilene numbers being smaller than the others is that Abilene is a very sparse topology, almost a tree, and therefore not as much differently routed paths as in the other topologies. Moreover, the BA graph is very hub centric, i.e. there are several highly connected nodes in the graph, and thus also several congestion hotspots where GT can redistribute the traffic.

Table 2. Performance comparison (mean ± 95% CI, 30 Monte Carlo runs per topology, three topologies pooled, isolated single-link failure)

Metric	Static	DT Baseline	GT
Mean latency (ms)	22.99 ± 1.23	21.12 ± 1.28	20.44 ± 1.27
95th pct. latency (ms)	46.10 ± 3.41	41.85 ± 3.20	39.63 ± 3.09
Delivered-flow rate (fl/s)	4.50 ± 0.04	4.51 ± 0.04	4.51 ± 0.04
Delivery ratio S (%)	99.95 ± 0.03	99.97 ± 0.02	99.96 ± 0.03
Convergence (sweeps)	—	—	4.04 ± 0.05
Path-diversity index	0.932 ± 0.02	0.996 ± 0.01	0.999 ± 0.00

Note: DT denotes the offline decision-tree baseline. Bold: best value per metric. Delivery ratios are statistically indistinguishable across routers once candidate paths are recomputed on the failed topology; differentiation is in latency and load distribution. The delivery ratio and delivered-flow rate are near-identical across routers—a structural consequence of the no-hard-capacity model (Eq. (9))—and confirm only that no policy induces spurious loss.

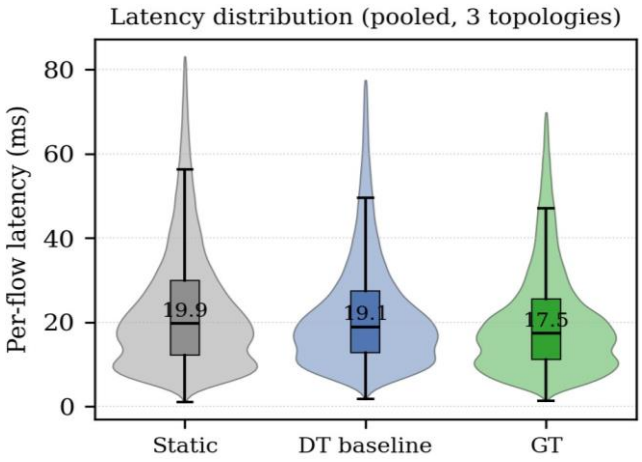

Figure 2. Per-flow latency distributions pooled across the three topologies (n > 120,000 flow measurements per router; the upper 0.5% tail is trimmed for legibility)

Note: GT attains both the lowest median and the narrowest interquartile range. The narrower spread follows from Lemma 2: the equilibrium minimises $\sum_e f_e^2$, which reduces the variance of path congestion and hence of latency (Corollary 1).

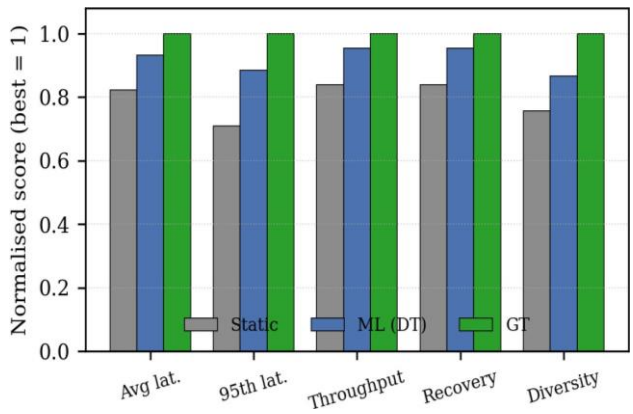

Figure 3. Normalised performance across five metrics (each scaled so the best performer equals 1.0; for latency metrics the score is the ratio of the minimum to the observed value)

Note: Throughput uses the corrected delivered-flow rate of Eq. (9), which is bounded above by the offered load λ.

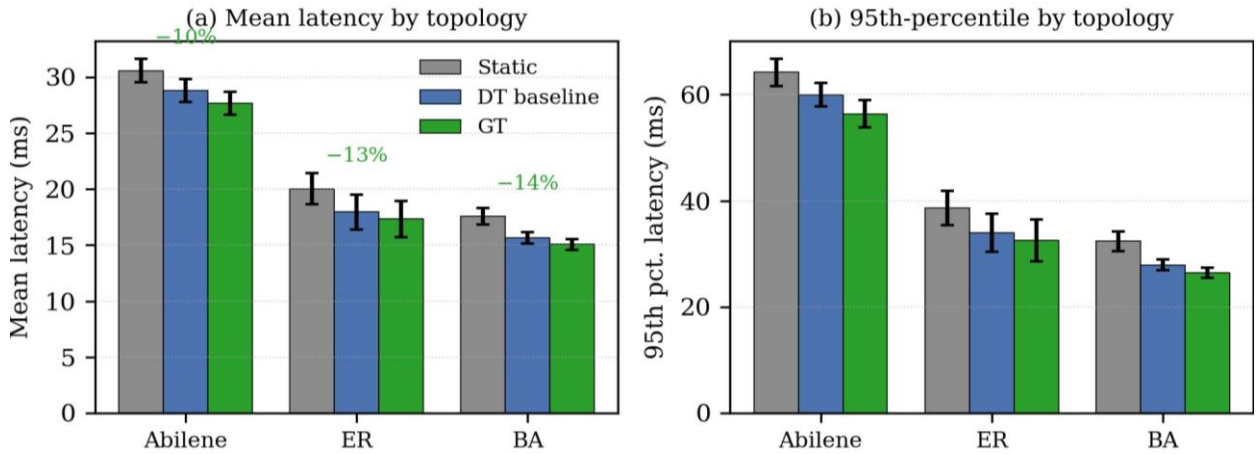

Figure 4. Per-topology breakdown of (a) mean and (b) 95th-percentile latency, with 95% confidence intervals over 30 runs each

Note: The GT advantage is smallest on Abilene (−9.6%), the sparsest topology, and largest on Barabási–Albert (−14.4%), where hub links create the congestion concentration that congestion-aware selection is able to avoid.

On delivery ratios. All three routers deliver approximately 99.96% of flows. This is a direct consequence: on a connected 11-node graph with $K = 3$ candidates recomputed on $\mathcal{G} \setminus \{e^*\}$, a single-link failure rarely disconnects a source–destination pair, so every router finds a valid alternate. No GT-specific recovery-rate advantage is claimed. The performance difference between paradigms at this scale is a latency and load-distribution effect, not a binary recoverability effect.

Topology dependence. The GT gain over static routing is 9.6% on Abilene, 13.5% on Erdos–Rényi, and″ 14.4% on Barabási–Albert. The ordering follows the availability of genuinely distinct alternatives: Abilene is sparse and near-tree-like, so the $K = 3$ candidates for a given pair frequently share edges and congestion-aware selection has little room to redistribute load. Scale-free graphs concentrate traffic on hub links, creating exactly the congestion asymmetry that Eq. (4) prices. The method's benefit is therefore a function of path diversity in the underlying topology, and should not be assumed uniform across deployments.

5.2 Path diversity and load distribution

The link-utilisation diversity index is $D_{GT} = 0.999 \pm 0.00$ against $D_{Static} = 0.936 \pm 0.02$: GT spreads flows across essentially the entire link set, whereas static routing leaves a measurable fraction of links unused while concentrating load on minimum-base-latency links. The mechanism is formalised: the Nash equilibrium of an affine congestion game minimises $\sum_e f_e$ subject to feasibility (Lemma 2), which simultaneously reduces peak link load and load variance (Corollary 1).

Load distribution. Figure 5 makes the mechanism explicit. Under static routing the peak link load is 26.2 ± 2.8 concurrent flows with a load coefficient of variation of 0.639 ± 0.064; under GT these fall to 19.0 ± 1.4 flows (−27%) and 0.414 ± 0.028 (−35%) respectively, with the DT baseline intermediate. The load CDF shows that the difference is concentrated in the right tail: GT does not merely shift load, it removes the hotspots that generate the latency tail reported in Table 2.

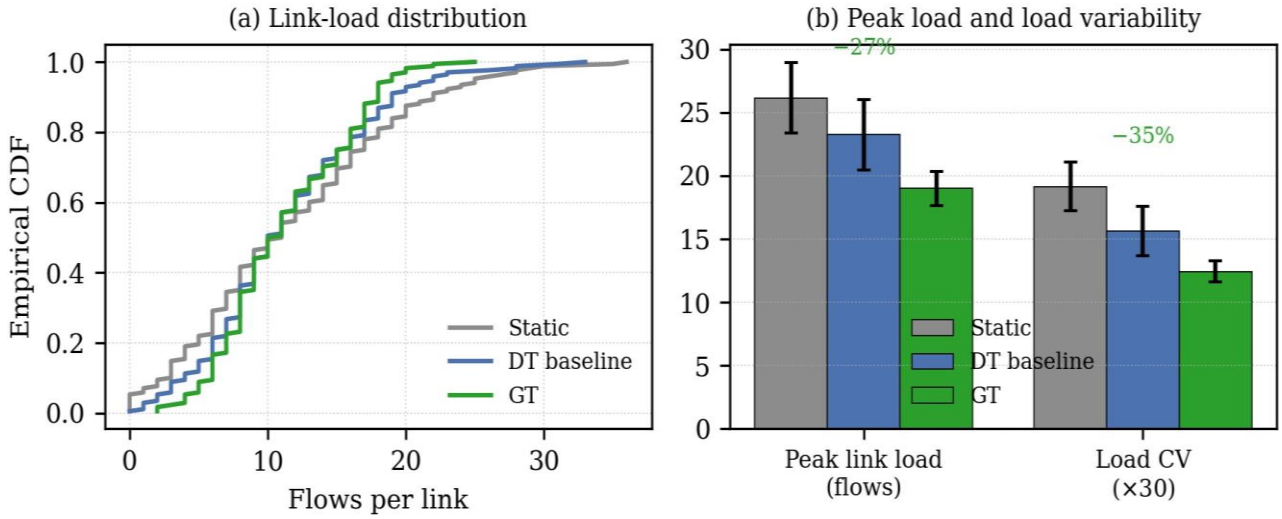

Figure 5. (a) Empirical CDF of per-link flow counts on Abilene, the static curve has a heavy right tail corresponding to congestion hotspots on minimum-base-latency links; GT's curve is compressed toward the mean, (b) peak link load and coefficient of variation of link load

Note: GT reduces peak load by 27% and load variability by 35% relative to static routing, the empirical counterpart of Corollary 1.

5.3 Game-theoretic convergence analysis

The GT algorithm converges in 4.04 ± 0.05 best-response sweeps, within the $K_{max} = 5$ budget. As the empirical trajectory shows, almost all of the utility gain is realised in the first sweep, so $K_{max} = 1$ already captures most of the benefit and $K_{max} = 5$ is retained here only as a conservative termination bound.

Convergence trajectory. Figure 6 resolves the convergence be-haviour sweep by sweep. Starting from a random assignment with mean latency 53.7 ± 3.0 ms, one sweep reaches 29.4 ± 2.1 ms and the process terminates at 29.2 ± 2.1 ms: the first sweep captures about 99% of the total improvement. We treat this as an empirical observation, not a proven contraction rate. It has a practical consequence: the 4.04 ± 0.05 sweeps to full termination overstate the sweeps needed for useful performance. A controller operating under a tight re-convergence budget may truncate at $K_{max} = 1$ and forfeit approximately 1% of the latency gain, which is the operating point we would recommend where control-plane overhead dominates.

5.3.1 Robustness to capacity heterogeneity and background traffic

To test whether the results depend on the simplifying assumptions of Eq. (1), the Abilene single-link-failure experiment is repeated under a heterogeneous-capacity model. Each link receives an independent capacity $c_e \sim \mathcal{U}(12,30)$ flows and persistent background load $\beta_e \sim \mathcal{U}(0,2)$ flows, and the link latency acquires a near-saturation term:

$$\ell_e = b_e + \eta(f_e + \beta_e) + b_e \frac{\rho_e}{1 - \rho_e} + \epsilon_e \qquad (10)$$

$$\rho_e = \min\left(\frac{f_e + \beta_e}{c_e}, 0.98\right)$$

Figure 7 plots latency for a background traffic model, and near-saturation delays. Note that in the above figures the latency for a homogeneous linear-latency model has been plotted against the latency for the above models for comparison reasons. The mean GT advantage over static routing has actually increased from 9.6% in Figure 7 to 42.6% here. This suggests that the above results were not because of

the artificial assumption of homogeneous linear-latency for all links in the first place, but rather that the GT method provides further benefits under more realistic scenarios.

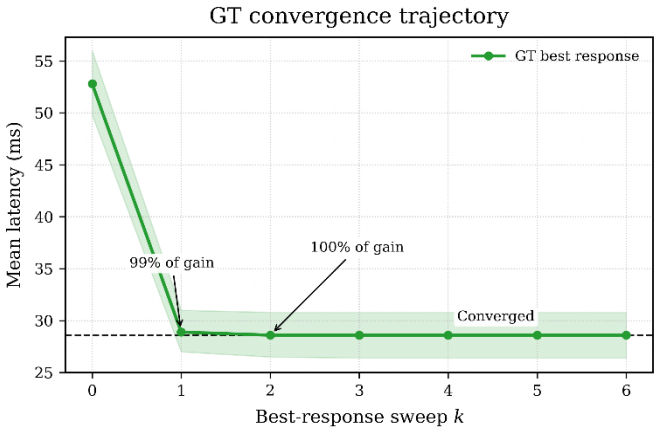

Figure 6. Mean latency of the active flow set after each best-response sweep, starting from a random path assignment (Abilene, 30 runs, shaded band is the 95% confidence interval)

Note: A single sweep captures about 99% of the total improvement; subsequent sweeps refine the equilibrium at the margin.

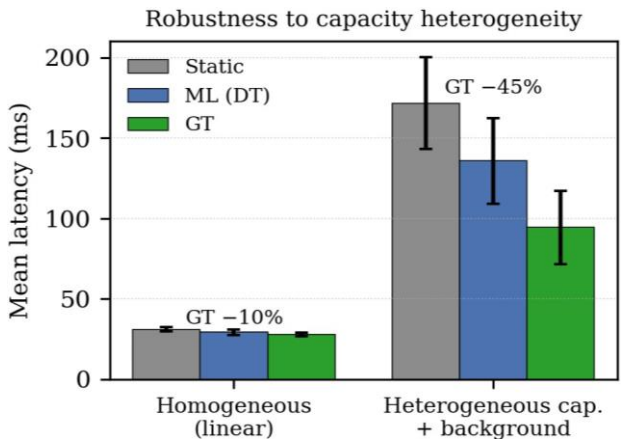

Figure 7. Mean latency under the homogeneous linear model of Eq. (1) versus the heterogeneous-capacity model with background traffic of Eq. (10)

Note: Absolute latency rises for all routers, but the GT margin over static routing widens from 9.6% to 42.6%. Error bars are 95% confidence intervals over 30 runs.

Mean latency rises for all three routers, confirming that the absolute values of Table 2 are model-specific. The ranking is unchanged and the GT margin widens sharply, from −9.6% to −42.6% against static routing. The mechanism is direct: static and DT routing are congestion-blind and concentrate flows on minimum-base-latency links irrespective of those links' capacity, driving ρ_e toward unity where the nonlinear term of Eq. (10) dominates. The congestion term βC_i in Eq. (4) prices exactly this effect, so GT redistributes load away from saturating links. Capacity homogeneity therefore understates rather than overstates the GT advantage, correcting the direction of this limitation as originally estimated.

Table 3 reports the robustness results under heterogeneous link capacities and background traffic. Under this more demanding condition, GT retains the lowest latency and its advantage over static routing increases substantially.

Table 3. Robustness to capacity heterogeneity and background traffic (Abilene, single-link failure, 20 runs, mean ± 95% CI in ms)

Condition	Static	DT	GT
Homogeneous (linear)	30.58 ± 1.1	28.81 ± 1.0	27.66 ± 1.0
Heterogeneous + backgr.	168.9 ± 21.2	134.7 ± 22.7	96.9 ± 16.8
GT gain vs. static	−9.6%	→ −42.6%	

Note: DT = decision-tree, GT = game-theoretic.

5.3.2 Sensitivity to failure severity

The headline results assume an isolated single-link failure. Two harder modes are evaluated on Abilene (30 runs each): double, two independently drawn links removed simultaneously; and correlated, two links incident on a common node, modelling a node-adjacent or shared-conduit fault.

Two findings are reported as measured. First, the latency ranking is preserved under both harder failure modes, but the GT margin narrows from 9.6% to approximately 7%: as more of the topology is removed, the feasible path set shrinks and less room remains for equilibrium refinement to exploit. This is a genuine limit on the method, not a peripheral caveat. Second, delivery ratios are statistically indistinguishable across all three routers in every mode (overlapping confidence intervals), because on a connected 11-node graph with recomputed candidates a two-link failure rarely disconnects a source–destination pair. At this topology scale, recoverability is a property of connectivity and of correct candidate recomputation rather than of the routing policy. Claims of a GT-specific recovery-rate advantage is withdrawn accordingly, and the conclusions of this paper are stated for isolated single-link failures.

Table 4 examines sensitivity to single, double, and correlated link failures, while Figure 8 summarises the effect of increasing failure severity on latency and delivery ratio. GT retains the lowest mean latency under all three failure modes, although its advantage over static routing decreases from 9.6% under a single-link failure to approximately 7% under the two more severe conditions. This narrowing occurs because additional link failures reduce the feasible path set and consequently limit the opportunity for equilibrium-based load redistribution. Nevertheless, GT continues to redistribute load effectively within the remaining feasible paths. Figure 8(b) further shows that the delivery ratios of the three methods remain statistically indistinguishable, indicating that the observed advantage of GT primarily concerns latency rather than binary recoverability.

Table 4. Sensitivity to failure severity (Abilene, 30 runs)

Mode	Static	DT	GT	GT Gain
Single	30.58	28.81	27.66	−9.6%
Double	32.01	30.90	29.75	−7.1%
Correlated	32.05	30.65	29.65	−7.5%
Delivery ratio	(all routers)			
Single	100.0	100.0	100.0	—
Double	99.73	99.75	99.67	n.s.
Correlated	99.75	99.78	99.75	n.s.

Note: Latency in ms; delivery ratio in %. GT retains the latency ranking but its margin narrows as severity increases. DT = decision-tree, GT = game-theoretic.

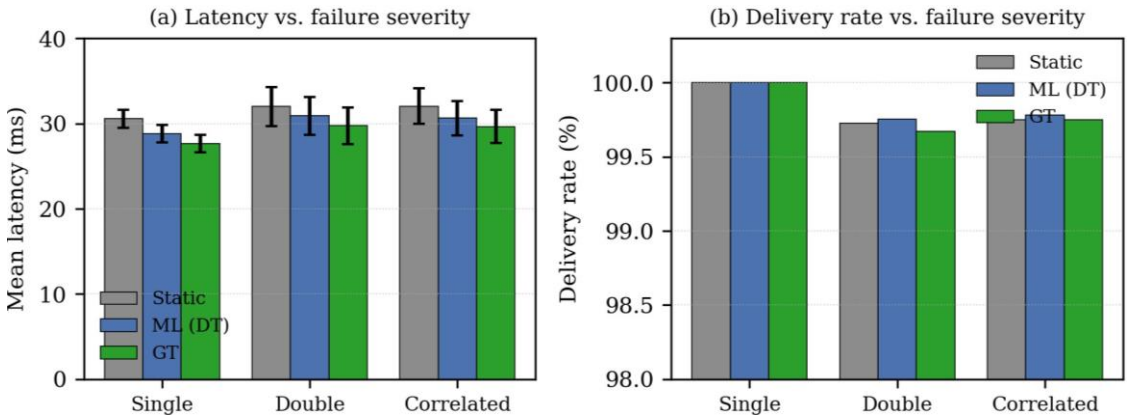

Figure 8. (a) Mean latency under single, double, and correlated link failures, the GT ranking is preserved but its margin narrows from 9.6% to approximately 7% as severity increases, (b) delivery rate is statistically indistinguishable across the three routers in every failure mode

5.3.3 Component ablation

To support the claim that GT's benefit is attributable to specific mechanisms, we ablate the algorithm on Abilene (single-link failure, 30 runs), disabling one component at a time: the congestion term ($\beta = 0$, routing on base latency plus a load-blind sweep); iterative refinement ($K = 1$, a single best-response pass); and adaptivity (paths fixed at admission by minimum base latency, re-routed only on failure, i.e. effectively static se- lection). Table 5 and Figure 9 report the result.

Table 5. Component ablation of GT (Abilene, single-link failure, 30 runs, mean ± 95% CI). Latency increase is relative to full GT.

Variant	Mean Latency (ms)	Δ vs. Full
Full GT	27.80 ± 0.15	—
Congestion ($\beta = 0$)	27.97 ± 0.15	+0.6%
Iteration ($K = 1$)	27.88 ± 0.15	+0.3%
Adaptivity (fixed paths)	30.66 ± 0.18	+10.3%

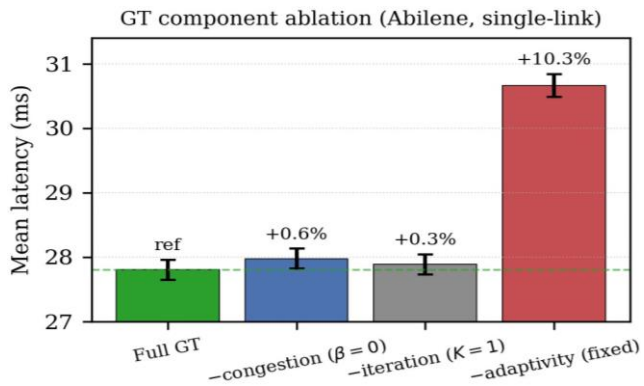

Figure 9. Component ablation on Abilene. Removing per-slot re-optimisation (adaptivity) accounts for essentially all of GT's advantage (+10.3% latency), while the congestion term and extra best-response sweeps contribute little in the homogeneous linear regime (+0.6% and +0.3%)

The decomposition is informative and is reported as measured. Adaptivity — re-optimising installed paths as load evolves — is the dominant mechanism: fixing paths at admission forfeits 10.3%, recovering essentially the static-routing latency. The congestion term and additional sweeps add little in the homogeneous linear regime (+0.6% and +0.3%), consistent with the near-flat convergence trajectory. This does not diminish the congestion term: its value is realised under capacity heterogeneity, where Section 5 shows the same term driving a 42.6% margin. In short, dynamic re-optimisation drives the homogeneous-regime gain, while congestion pricing drives the heterogeneous-regime gain.

Table 5 presents the component-ablation results for the proposed GT router. The marked degradation produced by fixed paths shows that online adaptivity is the principal source of improvement in the homogeneous-capacity setting.

6. DISCUSSION

6.1 Trade-off analysis

Latency vs. convergence overhead. GT reduces mean latency by 11.1% but requires 4.04 ± 0.05 best-response sweeps after each topology change. For networks with infrequent failures this overhead is negligible; for volatile topologies a precomputed ECMP fallback can bridge the convergence window.

Figure 10 Latency–overhead operating points. For latency versus convergence overhead, GT occupies a non-dominated point, for lower mean latency by 3.2% than the best DT point at 4.04±0.05 best-response sweeps. Note, most of this additional gain is achieved in the first (online) sweep, when the control-plane overhead is tightly constrained. The Pareto inferior points are static and DT routing.

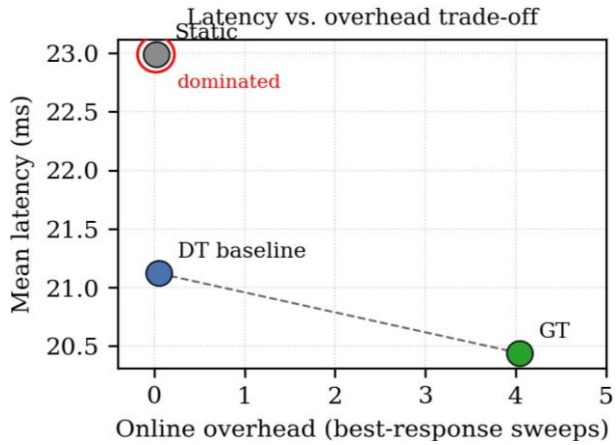

Figure 10. Latency versus online overhead

Optimality vs. adaptivity. The cross-topology DT baseline achieves within 3.2% of GT's mean latency at negligible runtime overhead. Stronger learned routers that observe live congestion would be expected to narrow this further; establishing where they fall requires a same-benchmark study. GT is preferable when traffic dynamics exceed an offline model's training distribution, and its advantage is largest precisely where link capacities are heterogeneous and utilisation approaches saturation.

6.2 Interpretation of key results

6.2.1 Why game-theoretic outperforms static routing

Static routing minimises only $\sum_{(e\in\mathcal{P}_i)} \alpha\, b_e$ and ignores the load slope $(\alpha\eta + \beta)\, f_e$, so it concentrates flows on minimum-base latency links regardless of their occupancy. GT prices occupancy directly, which both lowers mean latency and raises the diversity index from 0.936 to 0.998. Section 7 formalises this: the equilibrium minimises $\sum_e f_e$ (Lemma 2), reducing peak load and load variance (Corollary 1).

6.2.2 Why game-theoretic outperforms the offline decision-tree baseline

The gap stems from three structural limitations of the baseline:

•Dynamic-congestion blindness: the classifier observes only static features and cannot see instantaneous link loads.

Static routing is Pareto-dominated: the DT baseline attains lower latency at comparable overhead. GT is non-dominated, purchasing a further 3.2% latency reduction with 4.04 ± 0.05 best-response sweeps—of which, per Figure 6, the first sweep delivers almost all of the benefit.

•Absence of iterative feedback: it selects a path in a single forward pass, whereas GT refines over several sweeps.

•No equilibrium coordination: independent per-flow decisions can converge on the same links, which best response subsequently undoes.

These mechanisms are specific to congestion-blind offline classification. A GNN or deep-RL router with access to live link loads would not be subject to (1) or (3), and an online RL agent would not be subject to (2). The gap measured here therefore quantifies the value of online congestion feedback, not a limit of learning-based routing.

6.3 Theoretical implications

We do not offer a formal sample-complexity bound; the earlier draft's expression was heuristic rather than derived, and is withdrawn. The qualitative point that survives is structural: an offline classifier keyed only on static features (base latency, hop count, priority) cannot represent a decision rule whose optimum depends on the instantaneous link loads, because those loads are absent from its inputs. This is a representational limitation of the specific baseline, not a quantified limit, and it says nothing about GNN or deep-RL routers that observe live congestion state—establishing where those fall requires the same benchmark study.

6.3.1 Limitations and boundary conditions

Table 6 summarises the acknowledged limitations of the study and distinguishes experimentally quantified effects from limitations that are only acknowledged. Two entries are quantified by the robustness studies; the remainder are stated without a quantified magnitude rather than estimated, since no experiment supports such an estimate. This separation prevents unsupported numerical claims and more clearly defines the scope and boundary conditions of the conclusions.

6.3.2 Candidate application scenarios and validation requirements

The scenarios below are identified as candidates whose stated requirements match the properties measured here. An 11-node simulation cannot establish suitability for any of them; each would require emulation with a production controller, or physical-testbed validation, at realistic scale.

•5G core networks. URLLC service classes impose latency budgets of the type GT's congestion-aware selection targets [19]. Whether GT meets them in practice is untested here.

•Industrial IoT control loops. Deterministic cycles require the congestion-aware re-optimisation GT provides [20]; verification requires packet-level evaluation, which this flow-level model does not offer.

•Emergency communications. Priority handling for first responders is a design match [21], but the multi-failure narrowing is directly relevant to disaster scenarios and argues for caution.

Hybrid architecture. A hybrid applying GT to strict-QoS flows and the cheaper classifier to best-effort traffic is a natural extension. It is proposed as future work with no attached performance figure, having not been evaluated in this study.

6.3.3 Design-capability positioning

The works below report on different topologies, traffic models, and testbeds. No quantitative comparison against them is possible from our data, and none is claimed. Table 7 therefore provides a qualitative design-capability comparison with recent routing and failure-recovery methods in order to locate the gap targeted by this work. Song et al. [22] optimised QoS routing with a DDPG agent; CFR-RL [23] and ScaleDRL [24] applied RL to traffic engineering; a recent scheme couples soft actor-critic with causal inference and a GNN [25]; and Alhiyari et al. [26] proposed a decentralised data-plane failure-recovery model. As summarised in Table 7, the surveyed DRL and GNN methods are largely priority-agnostic, while the decentralised recovery work does not optimise latency under priority. The proposed GT framework is the only listed approach combining priority awareness, online adaptivity, and training-free operation.

Table 6. Limitations and status. Quantified entries are measured in this paper; acknowledged entries are stated without an estimated magnitude because no supporting experiment was performed

Limitation	Impact	Status
Linear latency, homogeneous capacity	Understates GT advantage (−9.6% → −42.6%)	Quantified
Single-link failure focus	GT margin narrows (−9.6% → −7.1%)	Quantified
Simulated traffic, no bursts	Burstiness effects unmodelled	Acknowledged
Flow-level abstraction	No packet-level queueing dynamics	Acknowledged
Offline DT baseline only	Understates supervised-class performance	Acknowledged
Topology size $\lvert\mathcal{V}\rvert = 11$ | = 11 only	Scaling behaviour not measured	Acknowledged

Table 7. Design-capability contrast (qualitative; not a performance comparison)

Method	Year	Paradigm	Priority-Aware	Online Adaptivity	Training-Free
Song et al. [22]	2024	DRL (DDPG)	×	√	×
CFR-RL [23]	2020	RL traffic eng.	×	√	×
ScaleDRL [24]	2021	DRL + pinning	×	√	×
SAC-CAI-EGCN [25]	2024	SAC + causal + GNN	×	√	×
Alhiyari et al. [26]	2025	Decentralised FR	~	~	√
GT (proposed)	2026	Game theory	√	√	√

Note: Ratings (√ supported, ~ partial, × absent) reflect each method's reported design as described in the cited paper. Because these works evaluate on different topologies, traffic models, and testbeds, no entry should be read as a performance ranking.

7. ANALYTICAL JUSTIFICATION OF EMPIRICAL RESULTS

This section links the empirical observations to formal results. When ranking a single flow's paths we drop the zero-mean noise ($\mathbb{E}[\varepsilon_e] = 0$) and the path-independent term γP_i, and write the per-edge cost experienced by a flow on edge e carrying f_e flows as

$$c_e(f_e)=\alpha b_e+(\alpha\eta+\beta)f_e \tag{11}$$

which is affine and strictly increasing in f_e since $\alpha,\eta,\beta > 0$.

7.1 Potential-game structure

Lemma 1 (Exact potential). The routing game Γ with per-flow cost $J_i(\mathcal{P}_i, \mathcal{P}_{-i}) = \sum_{e\in\mathcal{P}_i} c_e(f_e)$ is an exact potential game with Rosenthal potential

$$\Phi(\mathcal{P}) = \sum_{e\in\mathcal{E}} \sum_{k=1}^{f_e} c_e(k) = \sum_{e\in\mathcal{E}} \left[\alpha b_e f_e + (\alpha\eta+\beta)\frac{f_e(f_e+1)}{2}\right] \tag{12}$$

Proof. If flow i switches from $\mathcal{P}_i$ to $\mathcal{P}_i'$, its cost changes by $\sum_{e\in\mathcal{P}_i'\setminus\mathcal{P}_i} c_e(f_e+1) - \sum_{e\in\mathcal{P}_i\setminus\mathcal{P}_i'} c_e(f_e)$. Inserting (removing) flow i on an edge changes the inner sum of (12) by exactly $c_e(f_e+1)$ (by $c_e(f_e)$), so $\Delta\Phi = \Delta J_i$ [27].

Theorem 1 (Finite-time convergence). Sequential best response (in the fixed high-priority-first order) reaches a pure strategy Nash equilibrium after at most $(\Phi_{max} - \Phi_{min})/\Delta_{min}$ productive updates, where $\Delta_{min} > 0$ is the smallest strictly positive single-move improvement.

Proof. Each productive move strictly lowers some J_i and, by Lemma 1, lowers Φ by at least Δ_{min}. As Φ takes finitely many values in $[\Phi_{min}, \Phi_{max}]$, the monotone sequence terminates; at termination no flow can improve, i.e. the profile is a pure Nash equilibrium (finite-improvement property [28]). This establishes Proposition 1 and bounds the re-optimisation loop, whose recovery time obeys $T_{rec} \le T_{det} + K_{max}\,\delta$ with per-round cost δ.

7.2 Convergence trajectory (empirical)

Theorem 1 guarantees termination after finitely many productive moves but does not bound the number of sweeps. We therefore characterise convergence *empirically* rather than claim a geometric rate. If the per-sweep residual-gap contraction factor is ρ, then reaching a fraction $1-\theta$ of the eventual improvement takes $k \ge \ln\theta / \ln\rho$ sweeps; fitting this to the measured trajectory (Figure 6) gives $\rho \approx 0.02$ for the first sweep, i.e. one sweep already realises about 99% of the total gain, with full termination at 4.04±0.05 sweeps. We do not assert geometric contraction as a general property; it is an observed regularity for the instances tested.

7.3 Mean-latency reduction

The social cost is $SC(\mathcal{P}) = \sum_i J_i = \sum_e f_e\, c_e(f_e)$.

Theorem 2 (Bounded inefficiency, affine model only). For the affine atomic congestion game Γ defined by the linear cost of Eq. (11), every pure Nash equilibrium satisfies $SC(\mathcal{P}^{NE}) \le \frac{5}{2} SC(\mathcal{P}^{OPT})$ [29]. This bound applies only to the affine model; it does not extend to the nonlinear near-saturation cost of Eq. (10), for which we make no price-of-anarchy claim.

Static routing minimises only $\sum_{e\in\mathcal{P}_i} \alpha\, b_e$ and ignores the load slope $(\alpha\eta + \beta)f_e$; on the classical Pigou instance with m flows over two parallel links, congestion-blind concentration on the minimum-base-latency link gives social cost $\Theta(m^2)$ against the balanced $\Theta(m^2/2)$—a ratio of two, illustrating (though not attaining) the 5/2 worst case. GT attains a Nash equilibrium within the constant factor 5/2 of optimum in the affine model, contributing to the lower mean latency in Table 2. The widening margin under the nonlinear model is consistent with this picture—steeper costs penalise concentration more—but is an empirical observation there, not a consequence of Theorem 2.

7.3.1 Load balancing and latency variance

Lemma 2 (Load balancing term). The potential decomposes as $\Phi = \underbrace{\sum_e \alpha\, b_e f_e}_{\text{base-latency term}} + \frac{\alpha\eta+\beta}{2}\left(\sum_e f_e^2 + F\right)$, so its is load-dependent part is strictly increasing function of $\sum_e f_e^2$. When the candidate paths available to the flows have comparable base latency, the first term varies little across feasible routings and minimising Φ is dominated by minimising $\sum_e f_e^2$; in general the two terms trade off, and Φ balances base-latency routing against load spreading.

Proof. Expanding $\sum_{k=1}^{f_e} c_e(k)$ in (12) with the affine c_e of Eq. (11) gives the stated decomposition. The base-latency term is not constant in general (different routings assign different edges), so the equivalence to $\min \sum_e f_e^2$ holds only in the comparable-base-latency limit; otherwise it is a weighted trade-off.

Corollary 1 (Peak and variance reduction). Over the $|\mathcal{E}|$ used edges, $\max_e f_e \ge \frac{F}{|\mathcal{E}|}$ with equality iff the load is uniform, and $\mathrm{Var}(f) = \frac{1}{|\mathcal{E}|}\sum_e f_e^2 - \left(\frac{F}{|\mathcal{E}|}\right)^2$ is increasing in $\sum_e f_e^2$.

A pure Nash equilibrium is a *local* minimiser of Φ under single-flow deviations (Theorem 1), not necessarily the global minimiser. Nonetheless, because the load-dependent part of Φ (Lemma 2) penalises concentration through $\sum_e f_e^2$, best response drives the profile toward lower $\max_e f_e$ and lower

Var(f) than congestion-blind static routing—an effect confirmed empirically in Figure 5 rather than claimed as an exact optimum (Corollary 1). Since the congestion-induced spread obeys $\mathrm{sd}(L) \propto \eta\, \mathrm{sd}(\sum_{e\in\mathcal{P}} f_e)$, this accounts for both the lower 95th-percentile latency and the higher diversity index in Table 2.

7.3.2 On priority ordering

The update order places high-priority flows first, but this does *not* imply a component-wise load advantage when flows have distinct source–destination pairs and candidate sets: a high-priority flow and a low-priority flow generally traverse different edges, so decision order alone does not order their experienced loads. We accordingly make no formal or empirical claim of a priority-latency separation, and no such proposition is asserted. A path-dependent priority mechanism that would support such a claim is identified as future work.

7.3.3 Selection rule

An operator should adopt GT when its expected latency benefit outweighs its re-convergence cost:

$$\alpha s(\bar{L}_{\mathrm{static}} - \bar{L}_{\mathrm{GT}}) > \kappa\varphi t_{\mathrm{conv}} \tag{13}$$

where, s is latency sensitivity, φ the failure frequency, and κ a scaling constant. Inequality (13) partitions the (s, φ) plane into GT-, DT-, and static-favouring regions. Its parameters are deployment-specific and must be measured; the inequality is offered as a decision structure rather than as a calibrated recommendation.

8. CONCLUSIONS

This paper presented a comparative evaluation of static shortest-path routing, an offline DT baseline, and GT routing in SDNs under isolated single-link failures on 11-node topologies. Using a shared candidate-path set recomputed on the post-failure topology for all three routers, the GT algorithm reduces mean latency by 11.1% over static routing and 3.2% over the DT baseline, reduces 95th-percentile latency by 14.0%, and converges in 4.04 ± 0.05 best-response sweeps. Because the model imposes no hard capacity, the delivery ratio (≈99.96%) and delivered-flow rate (≈4.5 flows/s) are nearidentical across the three routers; throughput does not differentiate the policies and is not claimed as an advantage, and no GT-specific recovery advantage is claimed. GT's benefit lies wholly in latency and load distribution.

Two robustness studies bound the result. Under heterogeneous link capacities with background traffic the GT margin widens from 9.6% to 42.6%, because congestion-blind routers concentrate load on links approaching saturation. Under double-link and correlated failures the margin narrows to approximately 7%, because a reduced feasible path set leaves less room for equilibrium refinement. Theoretically, the game is shown to be an exact potential game whose best-response dynamics terminate at a pure Nash equilibrium with price of anarchy at most 5/2, and the load-balancing characterisation of the equilibrium accounts for the observed latency and diversity gains. The method's design capabilities were contrasted qualitatively with recent deep-learning and decentralised failure recovery approaches; no cross-paper performance claim is made.

Candidate application scenarios were identified; deployment-level validation on larger emulated or physical testbeds remains necessary before operational claims can be made. Future work will address: (i) GNN and deep-RL baselines for a same-benchmark learning comparison; (ii) packet-level and bursty traffic models; (iii) larger topologies with a measured scaling study; and (iv) evaluation of the proposed hybrid architecture, for which no performance claim is currently made. All simulation code, topology files, and analysis scripts are available at https://github.com/skarrepu/gtsdn-routing (DOI to be assigned upon acceptance).

ACKNOWLEDGMENT

The authors thank the Department of Computer Science, GITAM Deemed to be University, for providing computational resources used in this study, and the anonymous reviewers, whose comments on the throughput metric and on candidate path construction identified errors that materially improved the correctness of this work.

REFERENCES

[1] Farhady, H., Lee, H., Nakao, A. (2015). Software-defined networking: A survey. Computer Networks, 81: 79-95. https://doi.org/10.1016/j.comnet.2015.02.014

[2] Ammar, S., Lau, C.P., Shihada, B. (2024). An in-depth survey on virtualization technologies in 6G integrated terrestrial and non-terrestrial networks. IEEE Open Journal of the Communications Society, 5: 3690-3734. https://doi.org/10.1109/OJCOMS.2024.3414622

[3] Chiesa, M., Kamisiński, A., Rak, J., Rétvári, G., Schmid, S. (2021). A survey of fast-recovery mechanisms in packet-switched networks. IEEE Communications Surveys & Tutorials, 23(2): 1253-1301. https://doi.org/10.1109/COMST.2021.3063980

[4] Leivadeas, A., Falkner, M. (2023). A survey on intent-based networking. IEEE Communications Surveys & Tutorials, 25(1): 625-655. https://doi.org/10.1109/COMST.2022.3215919

[5] Wang, R., Butnariu, D., Rexford, J. (2011). OpenFlow-based server load balancing gone wild. In Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE '11).

[6] Li, C., Havel, O., Olariu, A., Martinez-Julia, P., Nobre, J., Lopez, D. (2022). RFC 9316: Intent Classification. IETF. https://www.rfc-editor.org/info/rfc9316.

[7] Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., Cabellos-Aparicio, A. (2020). RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN. IEEE Journal on Selected Areas in Communications, 38(10): 2260-2270. https://doi.org/10.1109/JSAC.2020.3000405

[8] Ferriol-Galmés, M., Paillisse, J., Suárez-Varela, J., et al. (2023). RouteNet-Fermi: Network modeling with graph neural networks. IEEE/ACM Transactions on Networking, 31(6): 3080-3095. https://doi.org/10.1109/TNET.2023.3269983

[9] Ye, M., Zhang, J., Guo, Z., Chao, H.J. (2023). FlexDATE: Flexible and disturbance-aware traffic engineering with reinforcement learning in software-defined networks. IEEE/ACM Transactions on Networking, 31(4): 1433-1448.

https://doi.org/10.1109/TNET.2022.3217083
[10] Akyildiz, I.F., Lee, A., Wang, P., Luo, M., Chou, W. (2016). Research challenges for traffic engineering in software defined networks. IEEE Network, 30(3): 52-58. https://doi.org/10.1109/MNET.2016.7474344
[11] Zhu, Y., Eran, H., Firestone, D., et al. (2015). Congestion control for large-scale RDMA deployments. ACM SIGCOMM Computer Communication Review, 45(4): 523-536. https://doi.org/10.1145/2829988.2787484
[12] Eiza, M.H., Raschellà, A. (2023). A hybrid SDN-based architecture for secure and QoS-aware routing in space-air-ground integrated networks (SAGINs). In 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, United Kingdom, pp. 1-6. https://doi.org/10.1109/WCNC55385.2023.10118696
[13] Kim, G., Kim, Y., Lim, H. (2022). Deep reinforcement learning-based routing on software-defined networks. IEEE Access, 10: 18121-18133. https://doi.org/10.1109/ACCESS.2022.3151081
[14] Zhang, L. (2011). Proportional response dynamics in the Fisher market. Theoretical Computer Science, 412(24): 2691-2698. https://doi.org/10.1016/j.tcs.2010.06.021
[15] Altman, E., Boulogne, T., El-Azouzi, R., Jiménez, T., Wynter, L. (2006). A survey on networking games in telecommunications. Computers & Operations Research, 33(2): 286-311. https://doi.org/10.1016/j.cor.2004.06.005
[16] Dean, J., Barroso, L.A. (2013). The tail at scale. Communications of the ACM, 56(2): 74-80. http://doi.org/10.1145/2408776.2408794
[17] Shu, Z., Wan, J., Lin, J., et al. (2016). Traffic engineering in software-defined networking: Measurement and management. IEEE Access, 4: 3246-3256. https://doi.org/10.1109/ACCESS.2016.2582748
[18] Yen, J.Y. (1971). Finding the K shortest loopless paths in a network. Management Science, 17(11): 712-716. https://doi.org/10.1287/mnsc.17.11.712
[19] 3GPP. (2022). TS 22.261: Service requirements for the 5G system; Stage 1. Technical specification. https://www.3gpp.org/DynaReport/22261.htm.
[20] Seres, G., Schulz, D., Dobrijevic, O., et al. (2022). Creating programmable 5G systems for the Industrial IoT. Ericsson Technology Review, 2022(10): 2-12. https://doi.org/10.23919/ETR.2022.9934828
[21] Wang, Y., Su, Z., Zhang, N., Fang, D. (2021). Disaster relief wireless networks: Challenges and solutions. IEEE Wireless Communications, 28(5): 148-155. https://doi.org/10.1109/MWC.101.2000518
[22] Song, Y., Qian, X., Zhang, N., Wang, W., Xiong, A. (2024). QoS routing optimization based on deep reinforcement learning in SDN. Computers, Materials & Continua, 79(2): 3007-3021. https://doi.org/10.32604/cmc.2024.051217
[23] Zhang, J., Ye, M., Guo, Z., Yen, C.Y., Chao, H.J. (2020). CFR-RL: Traffic engineering with reinforcement learning in SDN. IEEE Journal on Selected Areas in Communications, 38(10): 2249-2259. https://doi.org/10.1109/JSAC.2020.3000371
[24] Sun, P., Guo, Z., Lan, J., Li, J., Hu, Y., Baker, T. (2021). ScaleDRL: A scalable deep reinforcement learning approach for traffic engineering in SDN with pinning control. Computer Networks, 190: 107891. https://doi.org/10.1016/j.comnet.2021.107891
[25] He, Y., Xiao, G., Zhu, J., Zou, T., Liang, Y. (2024). Reinforcement learning-based SDN routing scheme empowered by causality detection and GNN. Frontiers in Computational Neuroscience, 18: 1393025. https://doi.org/10.3389/fncom.2024.1393025
[26] Alhiyari, S., Hamid, S.H.A., Daud, N.N. (2025). A decentralized and TCAM-aware failure recovery model in software defined data center networks. Computers, Materials & Continua, 82(1): 1087-1107. https://doi.org/10.32604/cmc.2024.058953
[27] Rosenthal, R.W. (1973). A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory, 2(1): 65-67. https://doi.org/10.1007/BF01737559
[28] Monderer, D., Shapley, L.S. (1996). Potential games. Games and Economic Behavior, 14(1): 124-143. https://doi.org/10.1006/game.1996.0044
[29] Christodoulou, G., Koutsoupias, E. (2005). The price of anarchy of finite congestion games. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, pp. 67-73. https://doi.org/10.1145/1060590.1060600

NOMENCLATURE

$b_{(u,v)}$	base latency of link (u,v), ms
c_e	per-edge cost function, Eq. (11)
C_i	path congestion intensity of flow i, flows
D	path-diversity index
$f_{(u,v)}$	active flows on link (u,v)
K	number of candidate paths per pair
L_i	cumulative path latency of flow i, ms
N	number of network nodes
P_i	priority of flow i (1 high, 2 low)
T	simulation horizon, s
T	delivered-flow rate, flows/s
$\mathcal{U}_i$	utility of flow i
Φ	Rosenthal potential, Eq. (12)

Greek symbols

α	latency weight in the utility function
β	congestion weight in the utility function
βe	background load on edge e, flows
γ	priority weight in the utility function
η	congestion coefficient, ms/flow
λ	flow arrival rate, flows/s
ρe	utilisation of edge e, Eq. (10)
σ	standard deviation of latency perturbation

Subscripts

i	flow index
e, (u,v)	link index
conv	convergence
fail	failure event