

System Relaxation for Interpretable and Adaptive Network Control

Zhiyuan Ren, *Member, IEEE*, Zhiliang Shuai, and Wenchi Cheng, *Senior Member, IEEE*,

Abstract—Prevailing network control strategies, which rely on static shortest-path logic, suffer from catastrophic “stress concentration” on critical nodes. This paper introduces the System Relaxation Algorithm (SRA), a new control paradigm inspired by physical relaxation that guides a network toward an emergent equilibrium of load balance. SRA is an interpretable, ‘white-box’ dynamical system whose behavior is profoundly topology-dependent: in heterogeneous networks, it acts as a proactive performance optimizer, reducing peak centrality by over 80% and increasing high-load throughput by more than 45%; in homogeneous topologies, its objective intelligently shifts to resilience enhancement. We rigorously prove its global convergence and practical stability using the theory of non-smooth dynamical systems, establishing a predictable paradigm for network governance that intelligently trades off performance and resilience.

Index Terms—Network control, stress-aware governance, network resilience, self-organizing systems, load balancing, non-smooth dynamical systems, stability analysis.

I. INTRODUCTION

PREVAILING network control paradigms, rooted in static shortest-path logic, inherently lack a mechanism for managing structural stress. This oversight leads to “stress concentration,” a severe imbalance of load on a few critical nodes. We formalize this “stress” using the network science metric of **Betweenness Centrality (BC)**, where a high BC value identifies a node as a topological bottleneck and a point of structural vulnerability. This endemic issue manifests as two critical symptoms: structurally, it creates “stress singularities” that make the network fragile to cascading failures; performatively, it causes premature capacity saturation, where bottlenecks lead to massive packet loss and soaring latency long before the network’s physical capacity is reached.

The root of this fragility lies in a flawed focus. Existing methods, including adaptive routing that reacts to transient metrics like queue length, concentrate on *pathfinding* within a given cost landscape. They fail to address the underlying structural cause because they do not attempt to *control the landscape itself*. This paper proposes a paradigm shift from reactive pathfinding to proactive stress control, inspired by the concept of “relaxation” in physical systems. We introduce the System Relaxation Algorithm (SRA), a model-driven control system designed to proactively disperse this stress, guiding the network toward a more robust and “relaxed” state of balanced load, as conceptually illustrated in Fig. 1. This

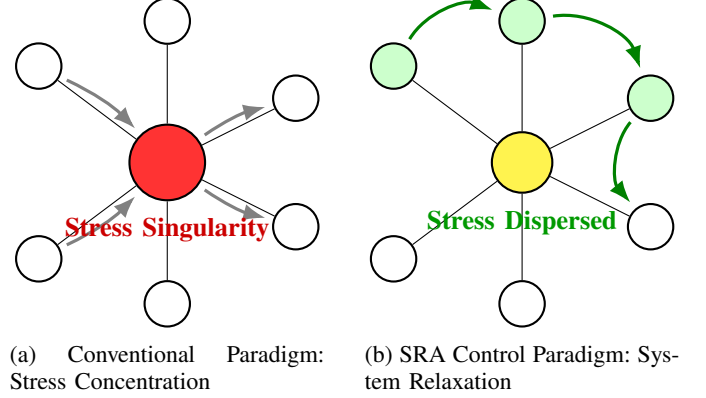


Fig. 1: Conceptual illustration of the core problem and the proposed SRA paradigm. SRA mitigates the “stress concentration” inherent in conventional network control by guiding the system to a “relaxed” state of balanced load.

control paradigm is perfectly aligned with modern architectures like Software-Defined Networking (SDN), where a central controller possesses the global view required for stress calculation and the direct mechanisms to enforce the resulting cost policies.

Our main contributions are:

- 1) **A Novel Control-Theoretic Model:** We formalize structural stress, quantified via betweenness centrality, as a controllable state variable within an interpretable, “white-box” dynamical system, introducing “system relaxation” as a new paradigm for network governance.
- 2) **Rigorous Analysis of Discontinuous Dynamics:** We are the first to model the discrete path switching in network control as a non-smooth dynamical system, providing formal proofs of stability using specialized tools like piecewise Lyapunov functions.
- 3) **Demonstrable Enhancement of Structural Robustness:** We demonstrate through simulations that SRA guides the network to self-organize into a verifiably more robust state, achieving an intelligent, topology-aware trade-off between performance and resilience.

The remainder of this paper is organized as follows. Section II reviews related work. Section III details our theoretical model. Section IV presents simulation results, and Section V concludes the paper.

II. RELATED WORK

The challenge of routing in dynamic networks has spurred research beyond traditional protocols, which struggle to handle

This work was supported by the National Key Research and Development Program of China under Grant No. 2024YFE0200300.

Zhiyuan Ren, Zhiliang Shuai, and Wenchi Cheng are with the School of Telecommunications Engineering, Xidian University, Xi’an 710071, China.

Corresponding author: Zhiyuan Ren (zyren@xidian.edu.cn).

the "stress concentration" problem in complex topologies. This has led to two primary advanced methodologies: reactive, load-aware routing and adaptive, AI-driven routing. In contrast, the System Relaxation Algorithm (SRA) proposed here represents a third, model-driven paradigm that maintains full interpretability. Table I provides a comparative positioning of SRA against these approaches.

TABLE I: Comparative Positioning of SRA against Existing Routing Paradigms

Feature	Traditional Load-Aware	AI-Driven	SRA (This Work)
Sensing Object	Real-time traffic state (e.g., queue lengths)	Traffic patterns & Topology features	Accumulated structural stress
Timescale	Instantaneous / Short-term	Short to Medium-term	Long-term equilibrium
Primary Goal	Mitigate local, transient congestion	Maximize performance metrics (e.g., throughput)	Disperse endemic structural stress
Core Method	Heuristic cost functions	Black-box learning (e.g., RL, GNN)	Model-driven dynamical system
Interpretability	High	Low	High

A. Traditional Load-Aware Routing

Load-aware routing protocols dynamically adjust paths based on real-time network state metrics. These metrics often include local information such as node queue lengths [1], channel busyness [2], or composite path load metrics [3]. For instance, the ALR protocol proposed by S and M [2] uses a composite metric of queue length and packet transmission success rate to select paths. Similarly, the CRADLE protocol by Macabale et al. [1] leverages queue occupancy and channel state information for path selection in wireless mesh networks. The LIRP protocol by Tao et al. [3] employs a cost function based on link quality and queue length. While effective at mitigating immediate, localized congestion, these methods often rely on heuristic rules and lack a global perspective on network structure. They primarily focus on state-aware path selection rather than fundamentally reshaping the underlying "cost landscape" of the network based on its structural properties, a key differentiator of our SRA framework.

B. Static Centrality-Based Routing Optimization

Centrality metrics, which quantify the structural importance of nodes, have been widely used to inform routing decisions. While some studies have explored physical layering [4] or closeness centrality [5], betweenness centrality (BC) [6], [7] is the most common metric for identifying structural bottlenecks. Several studies have proposed static or quasi-static methods that leverage BC. For example, Ahmed et al. [8] and Ma and Yang [9] used pre-calculated BC to optimize routing paths

and flow table storage, respectively. Upadhyay and Singh [10] attempted to use a genetic algorithm to find a single, static network configuration that minimizes the variance of BC. However, calculating BC is computationally intensive [11], and recent efforts have focused on distributed computation methods [12], [13]. The primary limitation of these approaches is their static or infrequently updated nature, which struggles to adapt to the rapid, dynamic changes in network topology and traffic that our SRA framework is designed to handle.

C. AI-Driven Adaptive Routing

Modern AI techniques, particularly machine learning and reinforcement learning, have been applied to create highly adaptive routing protocols. These methods can be broadly categorized as either heuristic optimization or reinforcement learning-based. Heuristic approaches like Genetic Algorithms (GA) [14] and Particle Swarm Optimization (PSO) [15], [16] can search for optimal routing policies but often require significant computational resources.

Reinforcement learning (RL) offers a more online, adaptive approach. Classical Q-learning algorithms [17] have been used to allow nodes to learn optimal next-hops based on local rewards. More advanced methods now combine Graph Neural Networks (GNNs) with Deep Reinforcement Learning (DRL) [18], [19] to learn complex relationships between network topology and traffic state for more sophisticated path selection. While powerful, these AI-driven methods often function as "black boxes," making their behavior difficult to predict, interpret, or formally verify. This lack of interpretability is a significant barrier to their deployment in mission-critical systems. In contrast, SRA is a fully model-driven, "white-box" framework whose behavior is governed by a clear, mathematically tractable dynamical system, a key advantage for mission-critical applications.

D. Positioning of Our Work

The novelty of the SRA framework lies in its unique synthesis of ideas from prior routes to create a system that is simultaneously: 1) **structural and proactive**, unlike symptomatic and reactive load-aware methods; 2) **dynamic and closed-loop**, unlike static centrality optimization; and 3) **mechanistically interpretable**, unlike "black-box" AI models. These fundamental distinctions are summarized in Table I. SRA's core contribution, therefore, is its systematic, cross-disciplinary integration of three domains: employing centrality measures from **network science** as the sensing signal, utilizing an integrator from **control theory** as the actuation engine, and applying the theory of **non-smooth dynamical systems** to provide rigorous mathematical guarantees for its stable operation. It is this integration that establishes a new, model-driven paradigm for resilient network design.

III. THEORETICAL MODEL AND METHODS

The primary challenge in designing any adaptive routing protocol is ensuring provable stability, as naive reactive approaches often collapse into destructive oscillations. To

overcome this, we move beyond simple heuristics and formally construct SRA as a **discrete-time, non-smooth dynamical system**. This control-theoretic approach is essential for modeling the complex interplay between the continuous evolution of node costs and the discrete "switching" of routing paths. This section provides a systematic exposition of this system: we first define the network model and the centrality-based state variables, then detail the core leaky-integrator evolution rule. The culmination is a rigorous proof of SRA's global convergence and practical stability using the powerful tools of piecewise Lyapunov functions and discrete-time inclusion theory.

A. Network Model and Load Metric

We model the network as an undirected graph $G = (V, E)$, where $N = |V|$. Throughout SRA's iterative process, path lengths are calculated based on a dynamically evolving node cost vector $C \in \mathbb{R}_+^N$. The cost of any path P is the sum of the transit costs of its constituent nodes, i.e., $d(P) = \sum_{v \in P} C_v$. Standard shortest-path algorithms are used to minimize this cumulative cost. We assume G is connected and $N \geq 3$.

To quantify the load concentration induced by the topology and cost distribution, we employ **Betweenness Centrality (BC)** as the core metric. The (unnormalized) BC of a node i , denoted $\kappa_i(S)$, is the total number of all-pairs shortest paths that pass through it:

$$\kappa_i(S) = \sum_{s \neq i \neq t \in V} \frac{\sigma_{st}^{C(S)}(i)}{\sigma_{st}^{C(S)}} \quad (1)$$

where $\sigma_{st}^{C(S)}$ is the number of shortest paths from s to t , and $\sigma_{st}^{C(S)}(i)$ is the subset of those paths traversing i . As $\kappa_i(S)$ sums proportions over all node pairs, it is an unbounded real number reflecting a node's absolute bridging role. To ensure the boundedness of state variables in our dynamical system, we use a max-normalized centrality vector in the algorithm's update steps:

$$\text{norm}(\kappa(S)) := \begin{cases} \frac{\kappa(S)}{\max_j \kappa_j(S)}, & \text{if } \max_j \kappa_j(S) > 0, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \quad (2)$$

Consequently, the vector $\text{norm}(\kappa(S))$ is always confined within the unit hypercube $[0, 1]^N$. Crucially, for a finite graph with a deterministic tie-breaking rule, the set of all possible shortest-path configurations is finite. This implies that the set of all possible values for $\text{norm}(\kappa(S))$ is also a **finite set of constant vectors**. This property is a cornerstone of the subsequent convergence analysis.

B. SRA: Core Mechanism and Algorithm Implementation

Having defined the network model and load metric, this section details SRA's dynamic cost evolution rule. We formally design SRA as a closed-loop control system, as depicted in Fig. 2, where the SRA controller governs the underlying network plant.

A naive feedback mechanism making node cost proportional to its current centrality is inherently unstable and leads to

persistent oscillations. To ensure stability, SRA's design must incorporate memory. We achieve this using an Integral (I) control structure, implemented as a **leaky integrator**. A core state vector, which we term the "accumulated pressure" $S \in \mathbb{R}_+^N$, represents the historically accumulated stress on each node. This state evolves according to the discrete-time transition equation:

$$S_i^{(k+1)} = (1 - \alpha)S_i^{(k)} + \alpha \cdot \text{norm}(\kappa_i^{(k)}) \quad (3)$$

where k is the iteration step and the learning rate $\alpha \in (0, 1)$ controls the system's memory. The $(1 - \alpha)$ term provides a critical **damping** effect, ensuring the stability of the integration process. The node transit cost C —the system's control output—is then defined as an affine function of this historical state:

$$C_i^{(k+1)} = 1 + \beta_I \cdot S_i^{(k+1)} \quad (4)$$

where 1 is a base cost and $\beta_I > 0$ is the "pressure-to-cost" conversion factor. By basing costs on the system's history (S), SRA makes stable, proactive decisions.

The complete iterative process is described in Algorithm 1, and its cyclical, four-phase workflow is visualized in Fig. 3. The algorithm introduces a key tunable parameter, the minimum dwell time T_d , which allows network operators to trade off path stability against system responsiveness. The core principle of SRA is to reshape the network's "cost landscape" based on accumulated structural stress. As shown conceptually in Fig. 4, this forces the routing protocol to intelligently trade path efficiency for bottleneck avoidance, rerouting traffic from a physically short but high-stress path to a longer but more stable peripheral route.

The computational complexity of a single SRA iteration is dominated by the all-pairs shortest-path and BC calculation, which is $O(N(E + N \log N))$ using Brandes' algorithm. While this cost can be substantial for large-scale networks, it is computationally feasible in the intended semi-dynamic or SDN-based deployment scenarios where a central entity performs the calculation. Furthermore, the centrality update is highly parallelizable, and future work can integrate the SRA framework with high-performance approximation algorithms, creating an effective trade-off between optimization quality and overhead for fully dynamic environments.

C. Theoretical Analysis of Model Convergence

This section provides a rigorous analysis of SRA's dynamical behavior. The inherent, discontinuous switching of routing paths makes the system **non-smooth**, requiring specialized analytical tools. Our analysis is structured as follows: first, we prove the existence of a global attractor for the system's discrete-time inclusion model, guaranteeing its trajectories are ultimately bounded. Second, we use a piecewise Lyapunov function to quantitatively characterize its oscillatory behavior. Finally, we introduce a minimum dwell time as an engineering mechanism to achieve practical stability and derive an explicit convergence rate.

1) *Global Attractiveness and Existence Guarantee for Convergence*: The convergence analysis of SRA is challenged by the piecewise constant, non-smooth nature of the centrality map $\kappa(S)$. To establish a rigorous guarantee, we reformulate the

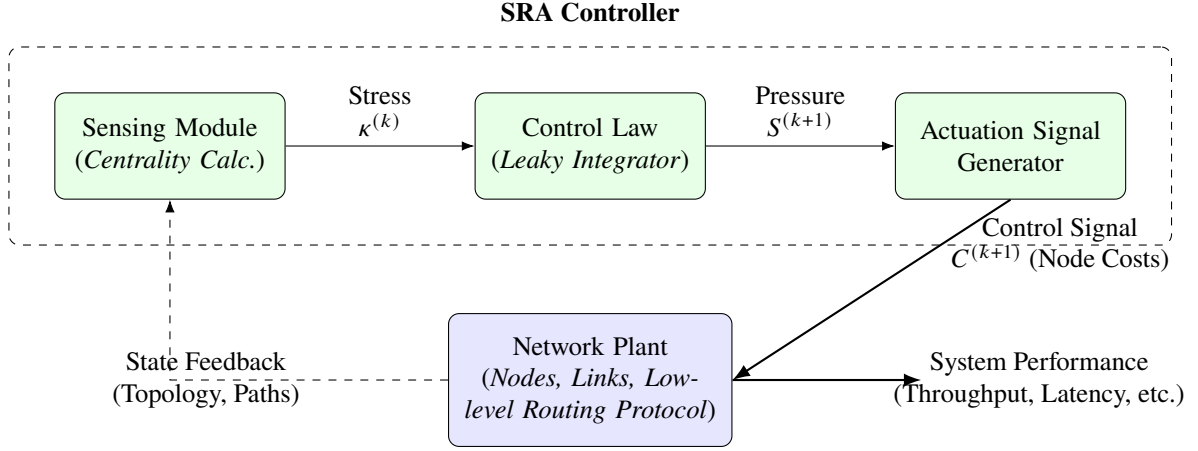


Fig. 2: The architecture of the SRA as a closed-loop control system. The SRA controller acts as a supervisory layer that governs the underlying network plant by dynamically adjusting the cost landscape.

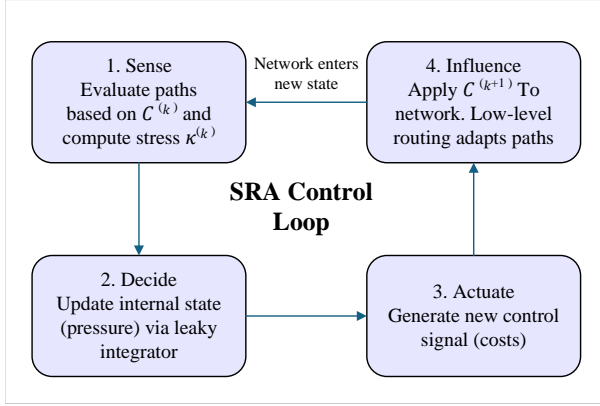


Fig. 3: The iterative workflow of the SRA control loop, illustrating the four key phases of sensing, decision-making, actuation, and system influence that drive the network toward a state of dispersed structural stress.

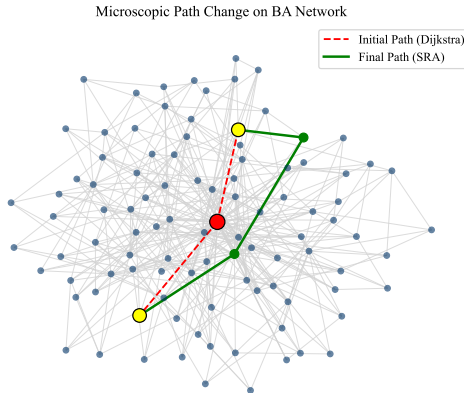


Fig. 4: A conceptual visualization of SRA's path reshaping mechanism. The algorithm diverts traffic from the efficient but congested 2-hop path (red dashed) through the central hub to a longer but more stable 4-hop peripheral path (green solid).

Algorithm 1: System Relaxation Algorithm (SRA)

Input: Graph $G = (V, E)$; learning rate $\alpha \in (0, 1)$; pressure-cost factor $\beta_I > 0$; stopping threshold $\varepsilon > 0$; max iterations K_{\max} ; (optional) min dwell time $T_d \geq 1$.

Output: Final node cost vector $C^{(k+1)}$ and the corresponding routing policy.

```

1 Initialize:  $S^{(0)} \leftarrow 0$ ;  $C^{(0)} \leftarrow 1$ 
2 for  $k = 0, 1, 2, \dots, K_{\max} - 1$  do
    // Step 1: Path Evaluation
3   Compute all-pairs shortest paths using node costs  $C^{(k)}$ 
    // Step 2: Centrality Calculation
4   Compute unnormalized centrality  $\kappa^{(k)}$  based on new paths
    // Step 3: Pressure Update
5    $S^{(k+1)} \leftarrow (1 - \alpha)S^{(k)} + \alpha \text{norm}(\kappa^{(k)})$ 
    // Step 4: Cost Update
6    $C^{(k+1)} \leftarrow 1 + \beta_I S^{(k+1)}$ 
    // Step 5: Anti-Chattering Strategy
7   if a path set switch occurred at  $k$  and  $T_d > 1$  then
8     Forcibly use current path set for the next  $T_d - 1$ 
      steps, executing only steps 2-6
    // Step 6: Convergence Check
9   if  $\|S^{(k+1)} - S^{(k)}\|_{\infty} \leq \varepsilon$  and path set is stable then
10    break

```

algorithm as a **discrete-time inclusion**, a standard tool from non-smooth dynamical systems. The update rule is regularized into the set-valued map:

$$S^{(k+1)} \in F(S^{(k)}) := (1 - \alpha)S^{(k)} + \alpha \mathcal{N}(S^{(k)}) \quad (5)$$

where $\mathcal{N}(S)$ is a Krasovskii regularization of the discontinuous mapping $\text{norm}(\kappa(\cdot))$, which takes the convex hull of all limit points in a vanishing neighborhood of S . This yields a mathematically well-behaved map, allowing us to prove convergence.

Lemma 1 (Positive Invariance and Absorbing Set): Let the state space be $X = [0, 1]^N$. The discrete-time inclusion for SRA, given by $S^{(k+1)} \in F(S^{(k)})$, evolves on a compact and positively invariant set X , which also serves as a compact absorbing set for the system.

Proof Sketch: The proof shows that if a state $S^{(k)}$ is within the unit hypercube X , its successor $S^{(k+1)}$ —being a convex combination of elements in X —must also lie within X . This establishes positive invariance, from which the absorbing property follows. *The full formal proof is provided in Appendix A.* ■

Lemma 1 provides a foundational safety guarantee against divergence. Next, we verify the map’s analytical properties.

Proposition 1 (Upper Semi-Continuity and Compact, Convex Values): The set-valued map $F(\cdot)$ defined in (5) is an **upper semi-continuous (USC)** map with **non-empty, compact, and convex** values.

Proof Sketch: The proof relies on the piecewise constant structure of $\mathcal{N}(S)$, which stems from the finite nature of the network’s path configurations. This structure ensures that the graph of the map is closed, which in turn guarantees the USC property. *The full formal proof is provided in Appendix B.* ■

With these properties established, we can prove the system’s global convergence to a stable equilibrium set.

Theorem 1 (Existence of a Global Attractor): For the discrete-time inclusion (5), there exists a non-empty, compact, and positively invariant set $\Omega \subset X := [0, 1]^N$ such that for any initial value $S^{(0)} \in X$,

$$\text{dist}(S^{(k)}, \Omega) \rightarrow 0 \quad (k \rightarrow \infty). \quad (6)$$

Proof Sketch: The proof is based on standard arguments for non-smooth dynamical systems. Since the system evolves on a compact, positively invariant set (Lemma 1) and is governed by a USC map (Proposition 1), any trajectory is pre-compact. The ω -limit set of any such trajectory is then shown to be non-empty and invariant, allowing for the construction of a global attractor Ω . *The full formal proof is provided in Appendix C.* ■

This theorem definitively guarantees that the SRA algorithm will always converge to a bounded, stable region Ω and will never diverge.

2) **Piecewise Lyapunov Analysis and Ultimate Boundedness:** The existence of a global attractor guarantees non-divergence, but does not quantitatively characterize the system’s final dynamics, such as persistent oscillations. To rigorously answer whether the algorithm will “oscillate indefinitely” and to bound this behavior, this section introduces a powerful tool from the analysis of non-smooth switched systems: the **piecewise Lyapunov function**.

Our goal is to construct a function $V(S)$ that quantifies the system’s “energy”—defined as its squared Euclidean distance to an ideal equilibrium set. We define it as:

$$V(S) := \min_{c \in \mathcal{N}(S)} \|S - c\|_2^2. \quad (7)$$

The existence of this minimum is guaranteed as $\mathcal{N}(S)$ is non-empty and compact. Concurrently, we define $\bar{\Delta} := \max_{i,j} \|v_i - v_j\|$ as the diameter of the finite set of all possible constant

centrality vectors $\{v_\ell\}$. The following theorem reveals the fundamental evolution law for this system “energy.”

Theorem 2 (Drop-Jump Inequality and Ultimate Boundedness): For any step k , the Lyapunov function satisfies the inequality:

$$V(S^{(k+1)}) \leq (1 - \alpha) V(S^{(k)}) + \frac{1}{\alpha} \bar{\Delta}^2. \quad (8)$$

Consequently, the system is ultimately bounded, with the limit superior of $V(S^{(k)})$ satisfying:

$$\limsup_{k \rightarrow \infty} V(S^{(k)}) \leq \frac{\bar{\Delta}^2}{\alpha^2}. \quad (9)$$

Proof Sketch: The proof hinges on bounding the one-step evolution of $V(S)$. By applying the Weighted Young’s Inequality, we establish a linear recurrence for $V(S^{(k)})$ that explicitly reveals its ‘drop’ and ‘jump’ terms, from which ultimate boundedness directly follows. *The full formal proof is provided in Appendix D.* ■

This “drop-jump” inequality is key to understanding SRA’s stability. It shows that the system’s energy $V(S)$ is governed by two competing forces: an energy **decay term** (the “drop”) with a contraction factor of $(1 - \alpha)$, representing the algorithm’s intrinsic damping; and an energy **injection term** (the “jump”), upper-bounded by $\frac{1}{\alpha} \bar{\Delta}^2$ and caused by path switching. This theorem proves that SRA’s oscillatory behavior is fully controllable. To better understand the damping effect, the following corollary analyzes the ideal case where no path switching occurs.

Corollary 1 (Local Exponential Convergence in a Fixed Partition): If the set of shortest paths does not switch, the update rule becomes a standard linear contraction map:

$$S^{(k+1)} = (1 - \alpha) S^{(k)} + \alpha c, \quad (10)$$

which converges exponentially to the unique fixed point $S^* = c$ at a rate of $(1 - \alpha)$.

Proof Sketch: The proof first solves the fixed-point equation $S = (1 - \alpha)S + \alpha c$ to show uniqueness of $S^* = c$. It then subtracts the fixed point from the update rule to derive the linear contraction relationship $\|S^{(k+1)} - S^*\|_2 = (1 - \alpha) \|S^{(k)} - S^*\|_2$, from which exponential convergence directly follows. *The full formal proof is provided in Appendix E.* ■

3) **Convergence Rate and Practical Stability with Dwell Time:** The “drop-jump” inequality guarantees that the system is ultimately bounded. However, for practical engineering applications, we must answer two further quantitative questions: How fast does the system converge to this stable neighborhood, and can we eliminate the residual oscillations (chattering)? This section shows that introducing a **minimum dwell time**, T_d , provides a definitive positive answer to both. By decomposing the system’s evolution into “dwell blocks,” we can derive an explicit convergence rate and formally prove that this strategy ensures practical stability.

To analyze the system under this policy, we decompose its evolution into “dwell blocks.” Let the minimum dwell time be T_d . We define the start of each block recursively: $\tau_0 = 0$ and $\tau_{m+1} = \tau_m + T_d$. The m -th dwell block is the time interval $[\tau_m, \tau_{m+1} - 1]$. We denote the state and Lyapunov energy at

the start of block m as $S_m := S^{(\tau_m)}$ and $V_m := V(S^{(\tau_m)})$, respectively.

Proposition 2 (Intra-Block Contraction and Inter-Block Recurrence): Under the minimum dwell time T_d policy, the set of shortest paths remains constant within a block. Let $\gamma := 1 - \alpha$, $\tilde{C} := \bar{\Delta}^2/\alpha$. The state and energy at the block endpoints are S_m and V_m respectively. Then,

- (i) (Intra-block contraction) The energy contracts exponentially within each block:

$$V(S^{(\tau_m+t)}) \leq \gamma^t V_m, \quad \text{for } t \in [0, T_d - 1]. \quad (11)$$

- (ii) (Inter-block recurrence) The energies at the start of adjacent blocks are related by the linear recurrence:

$$V_{m+1} \leq \gamma^{T_d} V_m + \tilde{C}. \quad (12)$$

Proof Sketch: The proof establishes two points. (i) For intra-block contraction, since the path set is fixed, the "jump" term in the Lyapunov inequality is zero, leading to simple exponential decay. (ii) For the inter-block recurrence, we chain the result from (i) over $T_d - 1$ steps of decay with a single "drop-jump" step at the block boundary to derive the stated linear recurrence for V_m . *The full formal proof is provided in Appendix F.* ■

This recurrence relation is the key to the global rate analysis. By solving it, we obtain explicit upper bounds for the steady-state radius and the number of steps required to converge.

Theorem 3 (Steady-State Radius and Step Bound to an ε -Tube): Let $\gamma = 1 - \alpha \in (0, 1)$ and $\tilde{C} = \bar{\Delta}^2/\alpha$. Then,

- (a) (Steady-state radius) The block-endpoint energy is ultimately bounded by:

$$\limsup_{m \rightarrow \infty} V_m \leq \frac{\tilde{C}}{1 - \gamma^{T_d}}. \quad (13)$$

- (b) (Steps to ε -tube) For a target radius $\varepsilon^2 > \frac{\tilde{C}}{1 - \gamma^{T_d}}$, the number of steps k to reach the set $\{S : V(S) \leq \varepsilon^2\}$ is bounded.

Proof Sketch: The proof is based on solving the linear recurrence relation derived in Proposition 2. By iteratively expanding the recurrence, we obtain a closed-form upper bound on V_m as a geometric series, from which the limit superior (steady-state radius) and the number of steps to reach an ε -tube can be directly calculated. *The full formal proof is provided in Appendix G.* ■

Finally, the following theorem consolidates all benefits of the dwell-time policy into a single, powerful guarantee of practical stability.

Theorem 4 (Dwell Time \Rightarrow Practical Stability): With a minimum dwell time $T_d \geq 1$, the following conclusions hold:

- (a) **Finite Switching / No Chattering:** The number of path set switches $N_s(T)$ within any time window of length T is bounded by $N_s(T) \leq 1 + \lfloor T/T_d \rfloor$.
- (b) **Practical Stability:** The system trajectories are ultimately bounded within a neighborhood whose size is explicitly determined by T_d , as given in Theorem 3.
- (c) **Convergence Time:** An explicit upper bound on the time to reach this neighborhood can be calculated as shown in Theorem 3.

Proof Sketch: The proof consolidates the consequences of the dwell-time policy. (a) Finite switching is a direct result of the policy's definition. (b) Practical stability is a direct consequence of the steady-state radius calculated in Theorem 3. (c) The convergence time bound is obtained by inverting the inequality derived in Theorem 3. *The full formal proof is provided in Appendix H.* ■

This theoretical result provides a valuable design principle. Given a desired steady-state energy radius R_{target}^2 , the minimum required dwell time T_d to guarantee it can be determined:

$$\frac{\tilde{C}}{1 - \gamma^{T_d}} \leq R_{\text{target}}^2 \implies T_d \geq \left\lceil \frac{\ln(1 - \tilde{C}/R_{\text{target}}^2)}{\ln \gamma} \right\rceil. \quad (14)$$

This clearly demonstrates how to quantitatively trade off system convergence speed against its final stability precision by tuning the single parameter T_d .

D. Betweenness Peaks and Network Performance

The convergence guarantees established previously do not formally imply that the resulting equilibrium state offers improved network performance. This section provides this theoretical link, building a rigorous bridge from the algorithm's proven peak-shaving dynamics to concrete performance gains. We first prove that SRA guarantees a net reduction of the peak centrality.

Lemma 2 (Blockwise Peak Drop): Within a dwell-time block that includes a path-switching event, if the network has no structurally unavoidable bottlenecks, then the peak unnormalized centrality is guaranteed to decrease, i.e., there exists some $\delta > 0$ such that $\kappa_{\text{max}}^{(m+1)} \leq \kappa_{\text{max}}^{(m)} - \delta$.

Proof Sketch: The existence of a net peak decrease is a direct consequence of the system's guaranteed evolution within a stable attractor (Theorems 1, 4). The state update rule ensures that a node with peak centrality experiences the strongest pressure increase, forcing an eventual traffic diversion that strictly reduces its unnormalized centrality. *The full formal proof is provided in Appendix I.* ■

This guaranteed reduction in peak centrality, a direct result of SRA's control law, translates directly into performance improvements. The following proposition quantifies this gain without relying on any idealized traffic redistribution assumptions.

Proposition 3 (Quantifying the Performance Gain): Consider a network where arrivals are proportional to centrality, $\Lambda_i \propto \kappa_i$. Let κ' be the centrality vector after applying SRA, and suppose the peak is reduced, $\kappa'_{\text{max}} < \kappa_{\text{max}}$. Then: (i)

- 1) **Unconditional robust lower bound.** The robust lower bound on network capacity, defined as $L(\kappa) := \mu_{\min}/\kappa_{\text{max}}$, strictly increases, i.e., $L(\kappa') > L(\kappa)$.
- 2) **Aligned-bottleneck capacity gain.** If the original bottleneck was at the peak node, the actual network capacity $\rho_{\text{max}}(\kappa) := \min_i \mu_i/\kappa_i$ strictly increases.

Proof Sketch: The proof leverages the guaranteed peak reduction from Lemma 2. (i) The increase of the robust lower bound $L(\kappa)$ follows directly from its definition, as κ'_{max} strictly decreases while μ_{\min} is constant. (ii) For the aligned-bottleneck case, the new capacity $\rho_{\text{max}}(\kappa')$ is lower-bounded by the post-SRA capacity of the original peak node, which is strictly greater

than the original network capacity. *The full formal proof is provided in Appendix J.* ■

Remark 1 (Generality and Limitations): A key strength of these performance guarantees is their generality; the proofs do not rely on simplifying assumptions regarding traffic redistribution and thus hold for the arbitrary, non-proportional shifts that occur in practice. However, if the topology contains unavoidable cuts or bridges, the ultimate capacity is limited by these structural features. In such cases, SRA still suppresses all *non-structural* peaks, and the robust lower bound guarantee of Proposition 3(i) continues to hold.

IV. SIMULATION ANALYSIS

The theoretical framework established in Section III guarantees SRA’s stability and links its peak-shaving mechanism to performance enhancement. This section provides comprehensive empirical evidence to validate these claims across diverse network conditions. We are guided by three objectives: first, to directly validate the core theoretical link between structural optimization and capacity gains; second, to analyze the macroscopic, topology-dependent dynamics of the “system relaxation” process; and finally, to benchmark SRA’s performance in terms of throughput, reliability, and robustness against standard and adaptive routing protocols.

A. Simulation Environment Setup

TABLE II: Summary of Simulation Parameters. Note on capacity: “Node Service Rate” (μ_i) is the steady-state rate perceived by flows, set from the “Node Proc. Capacity” (hardware-level budget) after accounting for overheads.

Parameter	Value / Description
A. Network Topology Parameters	
Network Types	Barabási-Albert (BA), Watts-Strogatz (WS), Erdős-Rényi (ER)
Default Size (N)	100 nodes
BA Attachment (m)	4
WS Neighbors (k)	8
WS Rewiring (p)	0.1
ER Connection (p)	0.1
B. SRA Algorithm Parameters	
Learning Rate (α)	0.1
Integral Gain (β_I)	10.0 (Default)
Max Iterations (K_{\max})	200
Convergence Tol. (ε)	1×10^{-5}
C. Dynamic Traffic Simulation Parameters	
Node Service Rate (μ)	1.0 packet/timestep
Total Timesteps (T)	2000
Warmup Period	300 timesteps
Node Proc. Capacity	10 packets/timestep
Node Queue Capacity	50 packets

To evaluate SRA’s universality, we conduct experiments on three canonical network models with distinct structural properties: Barabási-Albert (BA), Watts-Strogatz (WS), and Erdős-Rényi (ER). The default parameters for network generation,

the SRA algorithm, and the discrete-event traffic simulator are summarized in Table II. For benchmark protocols, Dijkstra and LIRP-D are deterministic; Q-Routing was configured with a learning rate of $\alpha = 0.1$, discount factor $\gamma = 0.9$, and exploration rate $\epsilon = 0.3$. All presented results are the mean of 100 independent trials with distinct random seeds; error bands, where shown, depict ± 1 standard deviation. The SRA controller’s cost landscape was updated at a fixed interval of 50 timesteps in dynamic simulations.

B. Empirical Validation of Core Theoretical Claims

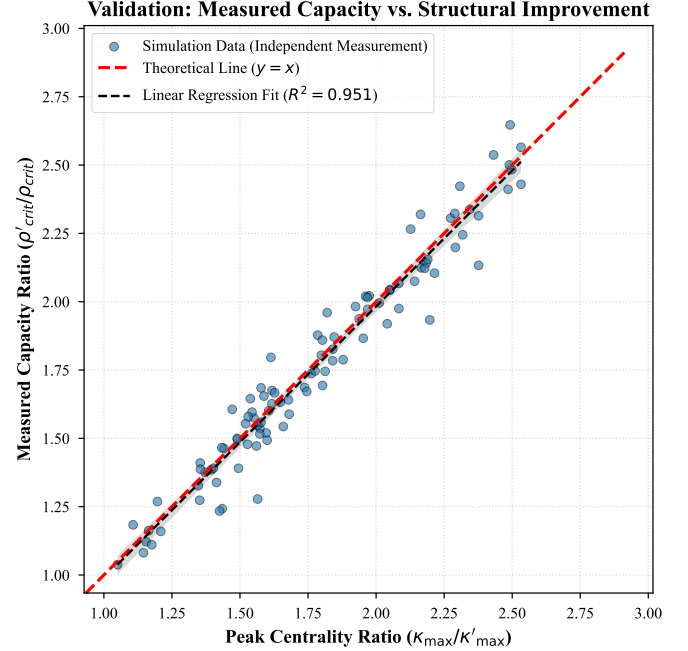


Fig. 5: Independently Measured Network Capacity Gain vs. Structural Optimization Ratio. The tight clustering of empirical data around the theoretical $y = x$ line validates our core theory that performance gains are driven by structural stress dispersion. Sample details are in Section IV-A.

This section empirically validates the two central claims from our theoretical analysis: first, that SRA’s structural optimization directly translates into quantifiable performance gains, and second, that the algorithm’s non-smooth dynamics are stable and controllable as predicted.

To validate the structure-performance link, we conducted an experiment on 100 randomized networks (BA and WS topologies, $N \in [40, 60]$). We performed two independent measurements on each sample: a static analysis of the peak centrality reduction ratio ($\kappa_{\max}/\kappa'_{\max}$) achieved by SRA, and a dynamic simulation to measure the actual network capacity gain ($\rho'_{\text{crit}}/\rho_{\text{crit}}$). The results are shown in Fig. 5. Under the experimental condition of homogeneous node service rates ($\mu_i = \text{const}$), our theory (Proposition 3(ii)) predicts a linear relation $\rho'_{\max}/\rho_{\max} = \kappa_{\max}/\kappa'_{\max}$. The tight clustering of empirical data around this $y = x$ line ($R^2 = 0.951$) provides powerful validation, confirming that SRA’s capacity enhancement is directly driven by its peak-shaving mechanism.

Furthermore, we validated the stability analysis presented in Section III-C. Numerical experiments confirm that the evolution of the piecewise Lyapunov function $V(S)$ perfectly reproduces the “drop-jump” dynamics predicted by Theorem 2. Moreover, the introduction of a minimum dwell time ($T_d > 1$) is shown to be a highly effective mechanism for eliminating the high-frequency “chattering” of the baseline algorithm, achieving the predictable, block-wise convergence guaranteed by Theorem 4. The detailed plots for these stability validations are provided in the Supplementary Material.

C. Dynamic Relaxation of Structural Stress and Analysis of the Equilibrium State

Having validated the underlying theory, we now analyze the structural properties of the equilibrium state that SRA achieves. To provide a statistically robust “before-and-after” comparison, we analyze the final network structure averaged over 100 independent runs. The results, summarized in Table III, provide precise quantitative evidence of SRA’s structural optimization benefits across different topologies.

The data highlight the algorithm’s profound and topology-dependent impact. SRA’s intervention is most aggressive in the heterogeneous BA network, which is prone to severe bottlenecks. Here, a minor cost—an average path length increase of only **5.63%**—yields massive structural gains: a remarkable **83.76%** reduction in peak centrality and a **75.38%** decrease in the standard deviation of centrality. This is powerful evidence of “stress dispersion,” where the burden on a few critical hubs is effectively redistributed across the network. In contrast, SRA’s behavior in the more homogeneous WS and ER networks is more conservative. It still achieves significant structural improvements (e.g., 49% peak centrality reduction in WS networks) but with a negligible efficiency cost (less than 2% for WS and none for ER).

These results offer strong quantitative support for our core thesis: SRA is a universal and highly effective control system for dispersing network structural stress, capable of applying a “targeted remedy” that intelligently trades a minor efficiency cost for a fundamental enhancement in structural stability.

TABLE III: Analysis of Structural Optimization Benefits of SRA (Mean \pm Std Dev over 100 runs)

Network	Metric	Initial (Dijkstra)	Final (SRA)	Change (%)
ER	Peak Centrality	589.93 (± 120.59)	209.40 (± 26.87)	-64.50%
	Std Dev of Centrality	106.93 (± 10.62)	33.55 (± 9.26)	-68.62%
	Avg Path Length	2.25 (± 0.04)	2.25 (± 0.04)	+0.00%
BA	Peak Centrality	2657.37 (± 619.87)	431.52 (± 45.86)	-83.76%
	Std Dev of Centrality	370.10 (± 43.87)	91.12 (± 6.86)	-75.38%
	Avg Path Length	2.37 (± 0.02)	2.51 (± 0.01)	+5.63%
WS	Peak Centrality	856.61 (± 169.69)	439.76 (± 60.07)	-48.66%
	Std Dev of Centrality	175.10 (± 22.43)	86.94 (± 14.21)	-50.35%
	Avg Path Length	3.05 (± 0.09)	3.10 (± 0.10)	+1.78%

D. Performance Validation under Dynamic Traffic Load

The preceding sections have confirmed SRA’s ability to optimize network structure. This section quantifies the impact of this structural optimization on key performance indicators (KPIs) under dynamic load via discrete-event simulations.

The results in Fig. 6 reveal that SRA’s behavior is highly adaptive, transforming its role based on the network’s underlying topology.

a) *Performance on BA Networks (Heterogeneous Topology)*: In the bottleneck-prone BA network, SRA acts as a **proactive performance optimizer**. As shown in Fig. 6(a), while benchmark protocols saturate quickly, SRA’s throughput continues to scale with the load, effectively doubling the network’s capacity in high-load scenarios. This gain is a direct result of superior congestion management, evidenced by a packet loss rate that remains below 10%, whereas other protocols suffer catastrophic losses exceeding 40%.

b) *Performance on WS Networks (Hybrid Topology)*: The WS network simulation (Fig. 6(b)) highlights SRA’s crucial role in **guaranteeing system stability**. While all protocols perform similarly at low loads, their behaviors diverge dramatically as the load surpasses the critical “congestion knee.” The learning-based Q-Routing protocol suffers a catastrophic performance collapse. In contrast, SRA is the only protocol that smoothly navigates this transition, maintaining stable throughput and controlled packet loss, ensuring “graceful degradation” under high stress.

c) *Performance on ER Networks (Homogeneous Topology)*: In the homogeneous ER network, which lacks dominant bottlenecks, SRA intelligently shifts its role to a **resilience enhancement mechanism**. As seen in Fig. 6(c), while all protocols exhibit nearly identical throughput, SRA consistently suppresses the packet loss rate to approximately 2%, a significant improvement over the 7% loss of other methods. This reliability enhancement comes at the cost of a predictably higher average latency, showcasing SRA’s adaptive “latency-for-reliability” trade-off.

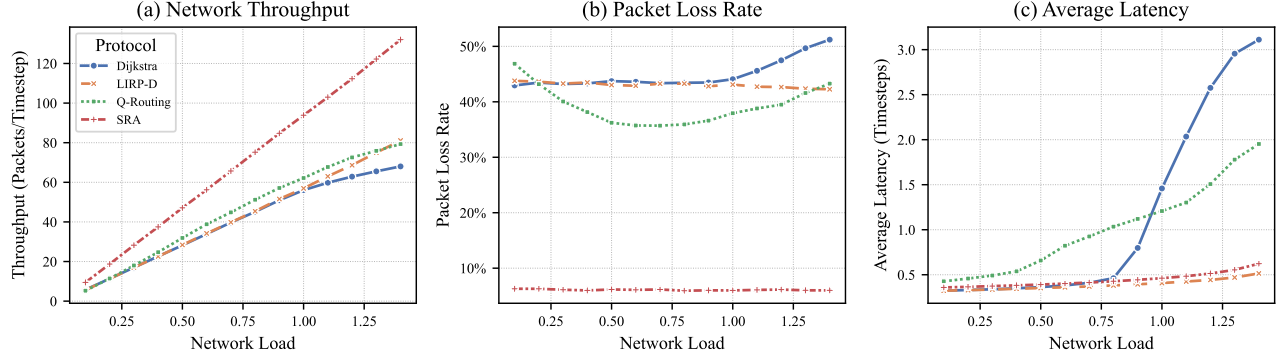
E. Parameter Sensitivity and Robustness Analysis

Before concluding the simulation analysis, we briefly discuss two final aspects: parameter sensitivity and robustness against targeted attacks. A detailed parameter sensitivity analysis, provided in the Supplementary Material, investigates the impact of the learning rate α and the pressure-cost factor β_I . The results reveal a classic performance trade-off: higher β_I values favor throughput and reliability at the cost of increased latency. This confirms that β_I acts as the primary lever for network operators to tune the system’s behavior between maximizing capacity and minimizing delay.

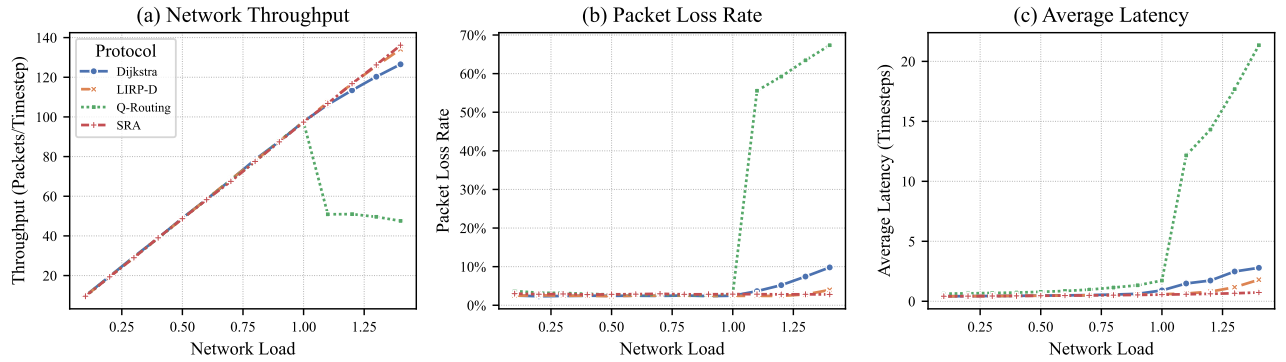
To quantify the robustness gains provided by SRA, we conducted a rigorous test under targeted attacks, where nodes were sequentially removed in descending order of their initial BC. After each removal, we measured the Packet Delivery Rate (PDR) for both SRA and the baseline Dijkstra policy. We define the **SRA PDR Advantage** as our primary metric:

$$\text{PDR Advantage (\%)} = \frac{\text{PDR}_{\text{SRA}} - \text{PDR}_{\text{Dijkstra}}}{\text{PDR}_{\text{Dijkstra}}} \times 100\% \quad (15)$$

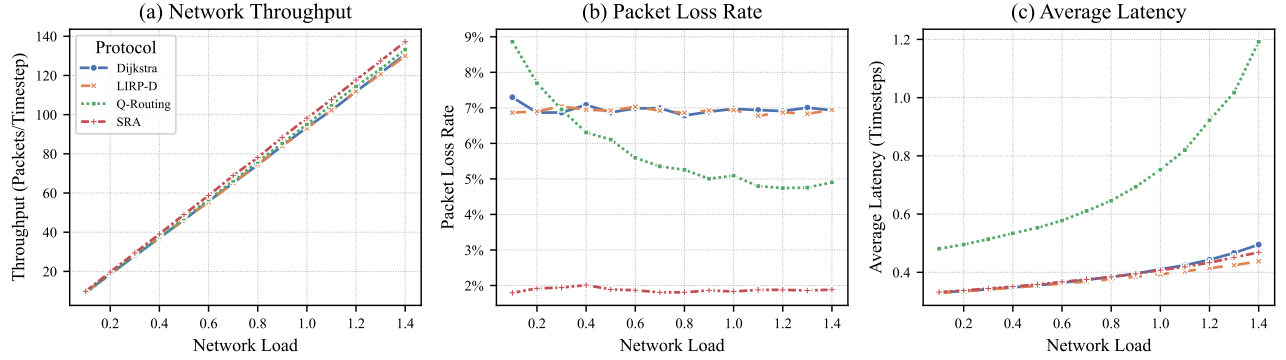
The results, averaged over 50 independent runs, are shown in Fig. 7. The findings provide strong statistical validation for SRA’s topology-dependent intelligence. In the vulnerable BA network, SRA provides a massive and consistent PDR advantage



(a) Performance on BA Network. SRA functions as a **proactive performance optimizer**, achieving significantly higher throughput and lower packet loss in the bottleneck-prone topology.



(b) Performance on WS Network. SRA **showcases its role in guaranteeing stability**, uniquely maintaining performance after the network reaches its critical "congestion knee" (load ≥ 1.0), a point where Q-Routing collapses.



(c) Performance on ER Network. SRA functions as a **resilience enhancement mechanism**, trading a predictably higher latency for a significant reduction in packet loss, thereby enhancing overall reliability.

Fig. 6: Dynamic traffic performance comparison of SRA against benchmark protocols across three distinct network topologies. The sub-captions highlight SRA's adaptive, topology-dependent roles.

of over 60%, confirming its role as a "performance engineer" that fundamentally enhances reliability. In the WS network, the advantage is "emergent," peaking around 20% after the network sustains critical damage. Finally, in the already robust ER network, SRA correctly provides a neutral (0%) advantage, avoiding unnecessary interventions. This validates that SRA's resilience-enhancing benefits are significant and reliable where needed, and correctly neutral otherwise.

V. DISCUSSION AND CONCLUSION

In response to the "stress concentration" problem endemic to conventional network control, this paper introduced, analyzed, and validated the System Relaxation Algorithm (SRA). We established the SRA framework on a rigorous theoretical foundation, uniquely applying the theory of non-smooth dynamical systems to formally prove the stability of a system managing

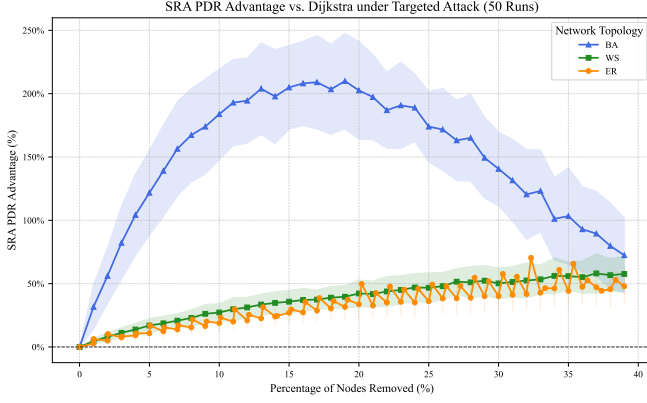


Fig. 7: The SRA PDR Advantage over Dijkstra under a targeted attack (mean \pm std dev over 50 runs). The results confirm SRA’s ability to significantly enhance resilience in vulnerable topologies (BA) and under critical damage (WS), while correctly remaining neutral in inherently robust networks (ER).

discontinuous path-switching dynamics. Comprehensive simulations then decisively demonstrated SRA’s core value: its profound, topology-dependent intelligence.

The central contribution of SRA lies not in optimizing a single metric, but in its intelligent, topology-aware navigation of the fundamental trade-off between **performance and resilience**. Our results reveal SRA acts as an adaptive controller whose strategy is dictated by the network’s structure. In **heterogeneous networks** (e.g., BA model) constrained by severe bottlenecks, SRA functions as a **proactive performance optimizer**, trading a minor increase in path length for a massive gain in capacity and reliability, evidenced by an over 80% reduction in peak centrality. Conversely, in **homogeneous networks** (e.g., ER/WS models), its objective intelligently shifts to that of a **resilience enhancement mechanism**, making finer adjustments to proactively manage risks from both dynamic congestion and structural damage.

This adaptive behavior converges not to a fragile, static fixed point, but to a **robust dynamic equilibrium**. This is a profound feature, not a failure of convergence, stemming from the non-smoothness of path selection. SRA’s core mechanism, the “leaky integrator,” is precisely designed to manage this dynamic, providing the memory and damping necessary to ensure predictable, bounded behavior that is far more resilient to real-world perturbations than a static optimum. Positioned within modern architectures, SRA represents a powerful, model-driven control application for an SDN controller, which naturally provides the global view needed for stress analysis and the means to enforce policies. This work lays a theoretical foundation for such model-driven resilient networks and opens several future research directions, including integration with mainstream SDN controllers, validation on real-world Internet topologies, and exploring adaptive parameter tuning mechanisms.

APPENDIX A PROOF OF LEMMA 1

Proof: (a) Positive Invariance: Take any $S^{(k)} \in X$. By max-value normalization, $\mathcal{N}(S^{(k)}) \subset [0, 1]^N = X$. For any choice of $c^{(k)} \in \mathcal{N}(S^{(k)})$ and any coordinate i , we have

$$S_i^{(k+1)} = (1 - \alpha)S_i^{(k)} + \alpha c_i^{(k)}, \quad \text{where } S_i^{(k)}, c_i^{(k)} \in [0, 1]. \quad (16)$$

Thus,

$$0 \cdot (1 - \alpha) + 0 \cdot \alpha \leq S_i^{(k+1)} \leq 1 \cdot (1 - \alpha) + 1 \cdot \alpha, \quad (17)$$

which implies $S_i^{(k+1)} \in [0, 1]$. As this holds for each coordinate, $S^{(k+1)} \in X$. This shows that $F(X) \subset X$: **if the state is in X at the current step, it remains in X at the next step**. Therefore, X is a positively invariant set.

(b) *Compactness:* The set $X = [0, 1]^N$ is a closed and bounded hypercube, and is therefore compact.

(c) *Absorbing Property on the “Effective State Space”:* The algorithm and analysis in this paper are initialized within X ($S^{(0)} \in X$), and from (a), we know $F(X) \subset X$. Therefore, for the discrete-time semi-flow on the state space X ,

$$\Phi(k, \cdot) : X \rightarrow 2^X, \quad \Phi(1, \cdot) = F, \quad \Phi(k + 1, \cdot) = F \circ \Phi(k, \cdot), \quad (18)$$

the set X **absorbs itself**: for any bounded set $B \subset X$, choosing $n_B = 0$ yields $\Phi(n, B) \subset X$ for all $n \geq 0$. By standard definitions, this means X is a **compact absorbing set**.

This leads to

$$\delta_{k+1,i}^+ \leq (1 - \alpha) \delta_{k,i}^+, \quad \delta_{k+1,i}^- \leq (1 - \alpha) \delta_{k,i}^-. \quad (19)$$

This means that $\max\{\delta_{k,i}^+, \delta_{k,i}^-\}$ converges geometrically to 0 at a rate of $1 - \alpha$, and thus $\text{dist}_\infty(S^{(k)}, X) \leq (1 - \alpha)^k \text{dist}_\infty(S^{(0)}, X) \rightarrow 0$. This shows **exponential attraction** to X . The main procedure of this paper is initialized within X , so the absorbing set formulation in (c) is sufficient.

In conclusion, $X = [0, 1]^N$ is positively invariant and compact for the system, and it is a compact absorbing set on the discrete-time semi-flow with state space X . ■

APPENDIX B PROOF OF PROPOSITION 1

Proof: (a) Non-empty, Compact, and Convex Values: By the definitions in Sections III-A, $\text{norm}(\kappa(S)) \in [0, 1]^N$. Thus, for any S ,

$$\mathcal{N}(S) = \text{co}(\text{norm}(\kappa(S))) \subset [0, 1]^N. \quad (20)$$

The closed convex hull, $\text{co}(\cdot)$, of a non-empty set is **non-empty**. It is also a **closed and bounded** convex subset of $[0, 1]^N$, and is therefore **compact** and **convex**.

(b) *Piecewise Constant Structure and Finiteness:* The graph G and the set of source-destination (OD) pairs are finite. The node cost $C(S)$ is an affine function of S . For each OD pair (s, t) , the set of candidate simple paths is **finite**. The cost of each path is the **pointwise minimum** of a family of affine functions of S . Thus, there exists a **finite partition** of the state space of S such that within each partition, the set of shortest paths for every OD pair remains **constant**. Therefore, within each partition, $\kappa(S)$

and $\text{norm}(\kappa(S))$ are **constant vectors**. On the boundaries of these partitions, the limit of $\text{norm}(\kappa(S))$ can only be one of the constant values from the **finite set of adjacent partitions**. Let these constant vectors be $\{v_\ell\}_{\ell \in I(S)}$, where $I(S)$ is the finite index set of partitions adjacent to S . Then,

$$\mathcal{N}(S) = \text{co}(\{v_\ell : \ell \in I(S)\}). \quad (21)$$

This establishes the structure of $\mathcal{N}(\cdot)$ as being **piecewise constant with the closed convex hull taken at the boundaries**.

(c) *Upper Semi-Continuity (USC) via Sequential Criterion / Closed Graph*: Take an arbitrary sequence $S_n \rightarrow S$, and let $c_n \in \mathcal{N}(S_n)$ such that $c_n \rightarrow c$. We must show that $c \in \mathcal{N}(S)$. From (b), for each n , there exists a finite set $\{v_\ell : \ell \in I(S_n)\}$ such that $c_n \in \text{co}(\{v_\ell : \ell \in I(S_n)\})$. By Carathéodory's theorem, we can write

$$c_n = \sum_{j=0}^N \lambda_{n,j} v_{\ell_{n,j}}, \quad \lambda_{n,j} \geq 0, \quad \sum_{j=0}^N \lambda_{n,j} = 1, \quad \ell_{n,j} \in I(S_n). \quad (22)$$

Since the vectors v_ℓ come from a **globally finite** set $\{v_1, \dots, v_M\}$, we can pass to a subsequence to ensure $\ell_{n,j} \equiv \bar{\ell}_j$ is **fixed**, and (by compactness of $[0, 1]$) $\lambda_{n,j} \rightarrow \bar{\lambda}_j$. Then,

$$c = \lim_{n \rightarrow \infty} c_n = \sum_{j=0}^N \bar{\lambda}_j v_{\bar{\ell}_j}, \quad \sum_{j=0}^N \bar{\lambda}_j = 1, \quad \bar{\lambda}_j \geq 0. \quad (23)$$

Since $S_n \rightarrow S$, for n large enough, any limit index $\bar{\ell}_j$ must belong to $I(S)$. Thus, each limit vector $v_{\bar{\ell}_j}$ is one of the constant vectors from a partition adjacent to S . Therefore,

$$c = \sum_{j=0}^N \bar{\lambda}_j v_{\bar{\ell}_j} \in \text{co}(\{v_\ell : \ell \in I(S)\}) = \mathcal{N}(S). \quad (24)$$

This means the graph of \mathcal{N} , $\text{gph } \mathcal{N} := \{(S, c) : c \in \mathcal{N}(S)\}$, is closed. A closed-graph map with compact values is **USC**.

(d) *Properties of $F(\cdot)$* : Scaling and translation preserve the USC property and non-empty, compact, convex values. The map $S \mapsto (1 - \alpha)S$ is continuous (single-valued). The Minkowski sum $F(S) = (1 - \alpha)S + \alpha \mathcal{N}(S)$ of a continuous map and a USC map is also USC with non-empty, compact, and convex values. ■

APPENDIX C

PROOF OF THEOREM 1

Proof: (a) Pre-compactness: From Lemma 1, $X = [0, 1]^N$ is a compact, positively invariant set. Thus, any trajectory starting in X remains in X . Since X is compact, any trajectory is **pre-compact** (i.e., has a convergent subsequence).

(b) *Existence of ω -limit sets*: For any $S^{(0)} \in X$, its ω -limit set is defined as

$$\omega(S^{(0)}) := \{x \in X : \exists k_n \uparrow \infty \text{ s.t. } S^{(k_n)} \rightarrow x\}. \quad (25)$$

By pre-compactness, $\omega(S^{(0)})$ is **non-empty** and **compact**.

(c) *Positive Invariance of ω -limit sets*: From Proposition 1, $F(\cdot)$ is USC with compact values, which implies its graph is closed. Take any $x \in \omega(S^{(0)})$. By definition, there is a subsequence $S^{(k_n)} \rightarrow x$. The subsequent points $S^{(k_n+1)} \in F(S^{(k_n)})$ are in the compact set X , so a further subsequence converges,

$S^{(k_n+1)} \rightarrow y$. By the closed graph property, $y \in F(x)$. Since $\{k_n + 1\}$ also tends to infinity, y is also in $\omega(S^{(0)})$. Thus,

$$F(\omega(S^{(0)})) \subset \omega(S^{(0)}). \quad (26)$$

(d) *Attraction to ω -limit sets*: It can be shown by contradiction that the trajectory must approach its ω -limit set. Thus, $\text{dist}(S^{(k)}, \omega(S^{(0)})) \rightarrow 0$.

(e) *Global Attractor*: Let Ω be the union of all ω -limit sets for all starting points in X :

$$\Omega := \bigcup_{S^{(0)} \in X} \omega(S^{(0)}). \quad (27)$$

This set Ω is non-empty, compact, and positively invariant. For any $S^{(0)} \in X$, since $\text{dist}(S^{(k)}, \omega(S^{(0)})) \rightarrow 0$ and $\omega(S^{(0)}) \subset \Omega$, it follows that

$$\text{dist}(S^{(k)}, \Omega) \rightarrow 0. \quad (28)$$

Thus, Ω is a global attractor for the system. ■

APPENDIX D

PROOF OF THEOREM 2

Proof: Let $c_k \in \arg \min_{c \in \mathcal{N}(S^{(k)})} \|S^{(k)} - c\|_2^2$, so $V(S^{(k)}) = \|S^{(k)} - c_k\|_2^2$. From the discrete-time inclusion, let $S^{(k+1)} = (1 - \alpha)S^{(k)} + \alpha c_k$. Let $c_{k+1} \in \mathcal{N}(S^{(k+1)})$ be a minimizer for $V(S^{(k+1)})$. We can write

$$S^{(k+1)} - c_{k+1} = (1 - \alpha)(S^{(k)} - c_k) + (c_k - c_{k+1}). \quad (29)$$

Using the Weighted Young's Inequality $\|x + y\|_2^2 \leq (1 + \lambda)\|x\|_2^2 + (1 + 1/\lambda)\|y\|_2^2$ with $\lambda = \alpha/(1 - \alpha)$, we get

$$\|S^{(k+1)} - c_{k+1}\|_2^2 \leq (1 - \alpha)\|S^{(k)} - c_k\|_2^2 + \frac{1}{\alpha}\|c_k - c_{k+1}\|_2^2. \quad (30)$$

Since c_k and c_{k+1} belong to the convex hull of a finite set of vectors, the distance between them is bounded, $\|c_k - c_{k+1}\| \leq \bar{\Delta}$. Substituting this in gives the desired inequality. The limit superior result follows from solving the linear recurrence inequality. ■

APPENDIX E

PROOF OF COROLLARY 1

Proof: (1) **Uniqueness of Fixed Point**: Solving $S = (1 - \alpha)S + \alpha c$ gives $\alpha S = \alpha c$, so $S = c$. The unique fixed point is $S^* = c$. (2) **Contraction and Convergence Rate**:

$$S^{(k+1)} - c = (1 - \alpha)S^{(k)} + \alpha c - c = (1 - \alpha)(S^{(k)} - c). \quad (31)$$

Taking the 2-norm and iterating gives the result:

$$\|S^{(k)} - S^*\|_2 = (1 - \alpha)^k \|S^{(0)} - S^*\|_2. \quad (32)$$

■

APPENDIX F

PROOF OF PROPOSITION 2

Proof: (i) Within a block, $\mathcal{N}(S) \equiv \{c_m\}$ is constant. The "jump" term in the inequality of Theorem 2 is zero. So, $V_{k+1} \leq \gamma V_k$, and iterating this t times gives the result. (ii) At the boundary between block m and $m+1$, a switch is allowed. We apply the "drop-jump" inequality for this single step:

$$\begin{aligned} V_{m+1} &= V(S^{(\tau_{m+1})}) \leq \gamma V(S^{(\tau_{m+1}-1)}) + \tilde{C} \\ &= \gamma V(S^{(\tau_m + T_d - 1)}) + \tilde{C} \\ &\leq \gamma \cdot (\gamma^{T_d-1} V_m) + \tilde{C} = \gamma^{T_d} V_m + \tilde{C}, \end{aligned} \quad (33)$$

where the last inequality uses the result from (i). ■

APPENDIX G

PROOF OF THEOREM 3

Proof: (a) Iteratively expanding the linear recurrence from Proposition 2(ii), $V_{m+1} \leq \gamma^{T_d} V_m + \tilde{C}$, gives:

$$\begin{aligned} V_m &\leq (\gamma^{T_d})^m V_0 + \tilde{C} \sum_{j=0}^{m-1} (\gamma^{T_d})^j \\ &= \gamma^{mT_d} V_0 + \tilde{C} \frac{1 - \gamma^{mT_d}}{1 - \gamma^{T_d}} \\ &\leq \gamma^{mT_d} V_0 + \frac{\tilde{C}}{1 - \gamma^{T_d}}. \end{aligned} \quad (34)$$

Taking the limit superior as $m \rightarrow \infty$ gives the steady-state radius. (b) Setting $V_m \leq \varepsilon^2$ and solving for m using the upper bound from (a) gives the result. ■

APPENDIX H

PROOF OF THEOREM 4

Proof: (a) By definition of the minimum dwell time, after a switch, no other switch is permitted for at least T_d steps. Thus, the number of switches in a window of length T is bounded. (b) This result is a direct consequence of solving the inter-block recurrence, as shown in the proof of Theorem 3(a). (c) This result follows from inverting the inequality in Theorem 3(b). ■

APPENDIX I

PROOF OF LEMMA 2

Proof: The proof consists of two parts: bounding the non-peak increase and guaranteeing the peak decrease.

1) *Bounding Non-Peak Increase:* Let $c^{(m)}$ and $c^{(m+1)}$ be the normalized centrality vectors at the start and end of the block. By definition of $\bar{\Delta}$, we have $\|c^{(m+1)} - c^{(m)}\|_\infty \leq \bar{\Delta}$, which means $|c_j^{(m+1)} - c_j^{(m)}| \leq \bar{\Delta}$ for all j . The unnormalized centralities are $\kappa_j^{(m)} = c_j^{(m)} \kappa_{\max}^{(m)}$ and $\kappa_j^{(m+1)} = c_j^{(m+1)} \kappa_{\max}^{(m+1)}$. Since the peak centrality does not increase (and in fact decreases, as shown next), we have $\kappa_{\max}^{(m+1)} \leq \kappa_{\max}^{(m)}$. Therefore, for any non-peak node $j \neq u$:

$$\begin{aligned} \kappa_j^{(m+1)} - \kappa_j^{(m)} &= c_j^{(m+1)} \kappa_{\max}^{(m+1)} - c_j^{(m)} \kappa_{\max}^{(m)} \\ &\leq c_j^{(m+1)} \kappa_{\max}^{(m)} - c_j^{(m)} \kappa_{\max}^{(m)} \\ &= (c_j^{(m+1)} - c_j^{(m)}) \kappa_{\max}^{(m)} \\ &\leq |c_j^{(m+1)} - c_j^{(m)}| \kappa_{\max}^{(m)} \leq \bar{\Delta} \cdot \kappa_{\max}^{(m)} =: B^{(m)}. \end{aligned}$$

2) *Guaranteeing Peak Decrease* ($\delta > 0$): The existence of a net peak decrease is a direct consequence of the system dynamics proven in Section III-C. The drop-jump inequality (Theorem 2) and the dwell-time policy (Theorem 4) guarantee that the system does not reach a fixed point but evolves within a stable attractor, forcing state transitions. A state transition (path switch) is triggered when the cost of a high-centrality node u , $C_u = 1 + \beta_I S_u$, increases sufficiently to make an alternative path cheaper. The state update rule $S_u^{(k+1)} = (1 - \alpha) S_u^{(k)} + \alpha c_u^{(k)}$ ensures that a node with peak normalized centrality ($c_u = 1$) experiences the strongest pressure increase. This mechanism guarantees that traffic is eventually diverted from the peak node u . As the network is assumed to have no structurally unavoidable bottlenecks, this diversion must lead to a strict reduction in its unnormalized centrality, κ_u . Therefore, there must exist a net decrease $\delta > 0$ over any block that contains such a path-switching event. The magnitude of δ depends on the system parameters (α, β_I, T_d) and the network topology, but its existence is guaranteed by the convergence dynamics. ■

APPENDIX J

PROOF OF PROPOSITION 3

Proof: The proof leverages the guaranteed peak reduction $\kappa'_{\max} < \kappa_{\max}$ established in Lemma 2.

(i) *Unconditional robust lower bound:* The robust lower bound on capacity is defined as $L(\kappa) := \mu_{\min}/\kappa_{\max}$. Since SRA does not change the physical node service rates, μ_{\min} is a constant. Given that $\kappa'_{\max} < \kappa_{\max}$, it follows directly that:

$$L(\kappa') = \frac{\mu_{\min}}{\kappa'_{\max}} > \frac{\mu_{\min}}{\kappa_{\max}} = L(\kappa). \quad (35)$$

The lower bound strictly increases.

(ii) *Aligned-bottleneck capacity gain:* The network capacity is defined as $\rho_{\max}(\kappa) = \min_{i \in V} \frac{\mu_i}{\kappa_i}$. The new capacity after SRA is $\rho_{\max}(\kappa') = \min_{i \in V} \frac{\mu_i}{\kappa'_i}$. By definition, the minimum of a set is less than or equal to any individual element. Therefore, the new capacity is lower-bounded by the capacity of the original peak node, u :

$$\rho_{\max}(\kappa') \geq \frac{\mu_u}{\kappa'_u}. \quad (36)$$

Furthermore, since u may not be the peak node in the new state κ' , its centrality κ'_u must be less than or equal to the new maximum centrality, κ'_{\max} . Thus:

$$\rho_{\max}(\kappa') \geq \frac{\mu_u}{\kappa'_u} \geq \frac{\mu_u}{\kappa'_{\max}}. \quad (37)$$

The premise of this case is that the original bottleneck was at the peak node, so $\rho_{\max}(\kappa) = \mu_u/\kappa_{\max}$. Combining these gives:

$$\rho_{\max}(\kappa') \geq \frac{\mu_u}{\kappa'_{\max}} > \frac{\mu_u}{\kappa_{\max}} = \rho_{\max}(\kappa). \quad (38)$$

This proves a strict increase in the actual network capacity. ■

REFERENCES

- [1] N. A. Macabale, A. N. Villasoto, J. D. Rivera, A. C. Ventanilla, M. P. P. Talplacido, A. L. Sevilla, and J. O. Garcia, "Cradle: Cross-layer design for load-aware routing in IEEE 802.11-based wireless mesh and sensor networks," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0970–0974.

- [2] D. S and G. M, "A novel adaptive load-aware routing protocol to enhance network lifetime in energy-constrained ad hoc networks," in *2025 6th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2025, pp. 374–377.
- [3] L. Tao, Y. Ji, and H. Wenxiao, "The uhf radio-based self-organizing routing protocol design," in *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE, 2011, pp. 2975–2978.
- [4] L. Guo, X. Wang, Y. Peng, J. Wu, X. Zhang, W. Hou, and J. Zhao, "Design on routing protocol in hybrid wireless self-organizing networks," in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 2010, pp. 569–573.
- [5] V. Aditya, S. Dhuli, P. Sashrika, K. Shivani, and T. Jayanth, "Closeness centrality based cluster head selection algorithm for large scale wsns," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2020, pp. 107–111.
- [6] L. Charekaoui, D. Khadija *et al.*, "A dynamic multi-criteria routing algorithm based in centrality measures for wsns," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*. IEEE, 2014, pp. 390–395.
- [7] E. M. Oliveira, H. S. Ramos, and A. A. Loureiro, "Centrality-based routing for wireless sensor networks," in *2010 IFIP Wireless Days*. IEEE, 2010, pp. 1–5.
- [8] M. Ahmed, K. Fuger, K. Kuladinithi, and A. Timm-Giel, "Poster: Can betweenness centrality mitigate greedy routing's sparse network problem?" in *2025 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2025, pp. 1–2.
- [9] Z. Ma and Y. Yang, "Optimization strategy of flow table storage based on "betweenness centrality"," in *2019 IEEE International Conference on Power Data Science (ICPDS)*. IEEE, 2019, pp. 76–79.
- [10] P. Upadhyay and Y. N. Singh, "Optimizing variance of link betweenness centrality distributing among links in elastic optical networks," in *2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2024, pp. 386–389.
- [11] P. Suppa and E. Zimeo, "A clustered approach for fast computation of betweenness centrality in social networks," in *2015 IEEE International Congress on Big Data*. IEEE, 2015, pp. 47–54.
- [12] L. Maccari, L. Ghiro, A. Guerrieri, A. Montresor, and R. L. Cigno, "Exact distributed load centrality computation: Algorithms, convergence, and applications to distance vector routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1693–1706, 2020.
- [13] —, "On the distributed computation of load centrality and its application to dv routing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2582–2590.
- [14] H. Mu, "High-speed carrier self-organizing network routing strategy based on genetic algorithm," in *2024 6th International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT)*. IEEE, 2024, pp. 344–349.
- [15] S. Wang, J. Zhang, Z. Zhang, and X. Yu, "A discrete particle swarm optimization algorithm for solving tsp under dynamic topology," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 2019, pp. 165–170.
- [16] X. Xia, Y. Tang, B. Wei, and L. Gui, "Dynamic multi-swarm particle swarm optimization based on elite learning," *IEEE Access*, vol. 7, pp. 184 849–184 865, 2019.
- [17] Y. Dou, H. Liu, L. Wei, and S. Chen, "Design and simulation of self-organizing network routing algorithm based on q-learning," in *2020 21st Asia-pacific network operations and management symposium (APNOMS)*. IEEE, 2020, pp. 357–360.
- [18] W. Wei, L. Fu, H. Gu, Y. Zhang, T. Zou, C. Wang, and N. Wang, "Grl-ps: Graph embedding-based drl approach for adaptive path selection," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 2639–2651, 2023.
- [19] H. Feng, W. Zhang, Y. Liu, Y. Li, J. Yin, and J. Kang, "Gnn-enabled multi-agent drl for adaptive path selection in multi-network domains." IEEE, 2025.