# RLRM: Reinforcement Learning-Based Routing for Ring-Augmented Mesh NoC

Juan Fang, *Member, IEEE*, Jingming Guo, *Student Member, IEEE*, Yubin He, *Student Member, IEEE*, and Qi Ming

***Abstract*—Modern multicore processors increasingly rely on network-on-chip (NoC) architectures to support high-bandwidth and low-latency communication. Traditional mesh-based NoC topologies suffer from uneven load distribution, with central router nodes frequently becoming congestion hot spots. The existing approaches mainly focus on introducing bypass links or independent ring interconnects. However, bypass links can only alleviate congestion between fixed routers, and routers not included in the ring still rely on conventional mesh forwarding, limiting the overall performance improvement. To address these challenges, this article proposes a reinforcement learning-based routing for ring-augmented mesh NoC (RLRM) to improve communication efficiency in multicore processors. First, we design a ring-augmented 8 × 8 mesh topology that integrates horizontal and vertical ring interconnects to achieve balanced load distribution and mitigate congestion hotspots. Second, we equip each router with a topology-aware reinforcement learning (RL) agent that dynamically selects paths based on congestion conditions, thereby proactively avoiding performance bottlenecks. Extensive experiments demonstrate that the proposed solution provides a scalable and efficient architecture for high-performance computing and multicore NoC systems. Experimental results under realistic benchmark traffic patterns show that RLRM reduces average latency by 22.78% compared to conventional mesh and static hybrid routing schemes.**

***Index Terms*—High-performance computing, mesh, network-on-chip (NoC), reinforcement learning (RL), routing.**

## I. Introduction

With the continuous increase in the number of processor cores in modern chips, the complexity and integration level of chip design have risen sharply. The network-on-chip (NoC), which organizes processing cores in a structured grid, provides a low-latency, high-bandwidth, and scalable interconnect, and has become the mainstream architecture for multicore processors [1], [2], [3]. For NoC architectures, performance and cost are the key evaluation metrics that determine communication efficiency and hardware overhead [4], [5]. The 2-D mesh topology, where routers are arranged regularly and communicate with neighbors in four cardinal directions [6], [7], offers scalability, layout regularity, and high concurrence and is therefore widely adopted in modern multicore systems [8], [9].

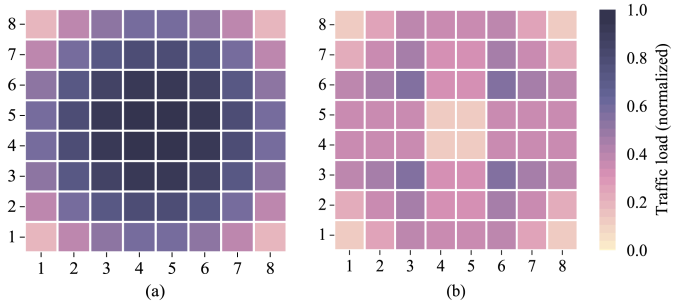


Fig. 1. Workload of (a) mesh and (b) ring-mesh our method. Under ideal ring operation, where the data volume and transmission probability between any two core nodes are identical, the visualization in (b) shows that our proposed ring-mesh design achieves a more balanced workload and reduces the load on the original mesh links by 46.7%.

However, previous work has demonstrated that the mesh architecture lacks edge symmetry, as the degree of nodes at the network boundary is smaller than that of nodes at the center. Consequently, the mesh topology places heavier load demands on interconnect links located in the network center compared to those at the edges. As reported in [10], even with techniques, such as the LAMP algorithm, central hot spots persist under real workloads. To further support this observation, a theoretical experiment is conducted. In an 88 × mesh network, assuming that the amount of transmitted data and the transmission probability between any two core nodes are identical, the router load distribution is shown in Fig. 1(a). It can be observed that the central nodes experience approximately three times higher load than the edge nodes.

Although prior studies have optimized mesh NoCs through bypass links, high-speed rings, and adaptive routing, none of these approaches can simultaneously address structural load imbalance and enable globally adaptive routing across heterogeneous interconnects. Traditional static routing strategies are inherently inflexible and cannot adapt to dynamic network conditions [11], [12]. Existing bypass-based methods mitigate congestion only between fixed node pairs [13], [14], while ring-based techniques accelerate traffic only for nodes located directly on the rings, leaving off-ring routers still constrained by multihop mesh forwarding [15]. Furthermore, prior reinforcement-learning-based routing schemes model

Received 26 September 2025; revised 9 December 2025 and 8 February 2026; accepted 9 February 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 61202076 and in part by Beijing Natural Science Foundation under Grant 4192007. *(Corresponding author: Qi Ming.)*

The authors are with the College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: fangjuan@bjut.edu.cn; guojingming@emails.bjut.edu.cn; heyubin@emails.bjut.edu.cn; chaser.ming@gmail.com).

Digital Object Identifier 10.1109/TVLSI.2026.3664149

only mesh networks and do not incorporate high-speed ring structures into the decision process [10], [16], resulting in limited adaptability under hybrid topologies. Therefore, a clear research gap exists: there is no unified ring-augmented topology together with a topology-aware RL routing framework capable of jointly leveraging mesh and ring paths for congestion mitigation and latency reduction.

Motivated by this gap, our work aims to design a hybrid topology in which ring interconnects are natively embedded into the mesh fabric, enabling packets to seamlessly traverse both interconnect types within a single routing path. This hybrid design not only alleviates central congestion but also exposes new routing opportunities that conventional mesh-only RL schemes cannot exploit. By incorporating ring load, ring-mesh transition behavior, and path diversity into the RL state and reward definitions, the proposed system enables topology-aware, congestion-adaptive routing that overcomes the limitations of previous approaches.

To address these challenges, this article proposes a hybrid NoC topology and a reinforcement learning (RL)-based routing algorithm, called RL-based routing for ring-augmented mesh NoC (RLRM). RLRM is inspired by urban traffic road design, which motivates the design of its hybrid network structure. By augmenting a conventional mesh network with high-speed ring interconnects, RLRM achieves a more balanced router load distribution. Under ideal ring operation, where the amount of data and the transmission probability between any two core nodes are identical, the resulting router load distribution is shown in Fig. 1(b), demonstrating a more uniform workload and a 46.7% reduction in the load on the original mesh links. The high-speed ring network enhances communication efficiency, while the RL-based routing algorithm leverages the structural characteristics of the topology to enable dynamic and adaptive routing decisions, aiming to balance network load, reduce transmission latency, and improve overall system performance.

Note that Fig. 1(b) is an upper bound visualization under an idealized setting rather than the steady-state behavior of our runtime routing. It assumes contention-free ring access. As a result, a large portion of medium- and long-distance traffic is absorbed by the ring layer and the mesh center naturally appears as a low-load region. In practical operation, however, the center does not remain underutilized because ring resources are limited and ring usage is regulated by probabilistic selection and RLRA's congestion-aware reward, which shift traffic back to mesh when ring ports become busy.

Based on the above observations, this work presents a unified hybrid architecture and an RL-driven routing mechanism that together provide structural enhancement at the topology level and dynamic intelligence at the algorithm level. The main contributions of this work can be summarized as follows.

1) We design a ring-mesh network topology. Distinct from previous ring-enhanced mesh network topologies [15], the ring–mesh allows a routing path to traverse both mesh and ring structures, thereby fully leveraging the advantages of both. This enhances the flexibility of communication between network nodes while shortening the critical-path communication distance.
2) A topology-aware RL routing scheme is proposed that differs from prior mesh-only Q-learning approaches. RLRA introduces ring-specific state encoding, a unified ring-mesh action space, and a compact multiobjective reward, forming the first RL routing framework tailored to hybrid ring–mesh NoCs.
3) We establish a paradigm that enables more flexible topology and routing algorithm design within a unified framework. This paradigm can be broadly applied to scenarios that demand high performance but are less sensitive to power consumption.

The remainder of this article is organized as follows. Section II reviews the related work on NoC topologies and routing algorithms. Section III presents the theoretical foundation. Section IV details the design philosophy and system architecture of RLRM. Section V reports the experimental evaluation and result analysis. Finally, Section VI concludes this article and discusses future research directions.

## II. Related Work

### A. Mesh Topology Optimization

The mesh structure is one of the most representative NoC topologies due to its regularity and scalability and has therefore attracted extensive optimization efforts [9].

Bypass-based enhancements add point-to-point links that allow packets to skip intermediate routers and relieve localized congestion. Farrokhbakht et al. [14] propose a low-overhead multipacket bypass mechanism using time-division multiplexing, reducing both latency and buffer usage. Ejaz and Sourdis [17] introduce a router architecture that applies bypass switching during traversal to lower latency. The SMART NoC family [13], [18], [19] further supports single-cycle multihop bypassing, enabling packets to skip several routers in one cycle and dynamically adapting to network conditions to reduce transmission latency. However, bypass links alleviate congestion only for fixed source–destination pairs and incur significant wiring overhead, making them unsuitable for scalable NoC deployments.

High-speed ring interconnects share a similar principle with bypass techniques. By introducing rings that operate independently of the mesh structure, packets among on-ring nodes can be transmitted at high speed, reducing the load on central mesh links [15]. Nevertheless, ring-based enhancements benefit on-ring nodes, and off-ring traffic still relies on multihop mesh routing, limiting system-wide performance gains.

Dynamic reconfigurable techniques adjust network connectivity based on runtime traffic, workload, and faults, improving resource utilization, load balance, and fault tolerance. RMC_NoC [20] introduces multifunctional reconfigurable channels that remap links under congestion or faults, enabling local reconstruction. Adapt-NoC [21] employs an RL controller to partition the mesh into subnetworks and dynamically select subtopologies according to application communication demands. Sparse Hamming graph [22] further extends reconfigurability by adding customized high-dimensional links, providing scalability, path diversity, and layout friendliness.

However, such dynamic schemes incur notable overhead. Frequent reconfigurations require additional hardware and

control logic, increasing wiring complexity and power consumption. Moreover, runtime adjustments depend on accurate traffic monitoring and decision-making, which may introduce delays or consistency issues in large multicore systems. Balancing flexibility with manageable system cost remains a major challenge for reconfigurable NoCs.

### B. Routing and RL in NoC

In NoC design, routing algorithms determine packet paths and directly affect latency, throughput, energy consumption, and load balance. Traditional static schemes, such as *XY* routing [23] and odd–even routing [24], are widely used due to their simplicity, but their lack of adaptability under dynamic and skewed traffic often leads to hot spots, path conflicts, and resource underutilization [12]. To address these limitations, researchers have explored both path-selection optimization and congestion/load-aware routing.

Flexible path selection has been used to enhance transmission efficiency. A thermal-aware predictive method in [25] constructs "hot zones" to avoid congested regions. An RL-based multipath mechanism in [10] performs intelligent message splitting to maximize bandwidth and reduce power. HaPM [11] improves multicast routing by leveraging hotspot prediction to shorten paths and alleviate congestion.

Congestion-aware mechanisms further improve performance under high load. An adaptive secure routing scheme for 3D-NoCs [26] employs real-time fault and security awareness to maintain reliability and reduce congestion. Other load-aware approaches [25], [27] adjust routing based on thermal or buffer utilization information to bypass hotspot routers.

RL has also been introduced to address complex design-space challenges [28]. A deep RL exploration framework [16] optimizes routerless NoC ring layouts by combining Monte Carlo tree search with neural policy learning. A dual-RL mapping strategy [29] reduces latency and congestion for neural network workloads. These studies, along with further evidence in [30], highlight RL's adaptability and effectiveness for routing, motivating its integration into the optimized mesh topology considered in this work. Moreover, the existing RL-based routing algorithms are designed for conventional mesh topologies and do not incorporate ring structures, preventing them from leveraging hybrid interconnects.

In summary, prior works have optimized either the topology or the routing mechanism, but none provides a unified framework that enables joint utilization of mesh and ring resources. This article addresses this gap by proposing a ring-augmented topology together with a topology-aware RL routing algorithm.

## III. Preliminaries

### A. Structural Analogy Between Mesh and Traffic Rings

The mesh topology is widely used in NoC design for its regularity, scalability, and compatibility with *XY* routing. Conceptually, a mesh resembles an urban traffic system in which routers act as intersections, links as roads, and packets as vehicles. As in real traffic, long-distance or burst flows can overload central intersections, producing hot spots when *XY* routing faces uneven communication demands. To alleviate such congestion and accelerate long-distance travel, urban planning introduces ring highways that bypass city centers [31], offering an analogy for improving communication efficiency in mesh-based NoCs.

Inspired by urban traffic roads, introducing a "global bypass" in NoCs has become a focus of research. Recent studies have proposed integrating reconfigurable ring interconnects on top of mesh networks, serving as low-latency "fast lanes" to support long-distance communication. A representative work [15] constructs a set of dynamically activatable ring interconnects on top of a mesh, enabling direct communication between hotspot nodes through preconfigured paths. These ring interfaces are simple, support single-hop single-cycle transmission, significantly reduce cumulative latency caused by router cascades, and consume far fewer hardware resources than conventional routers, demonstrating high integration adaptability.

The core idea is to intelligently select a subset of rings that connect frequently communicating node pairs, forming fast lanes analogous to urban ring highways. Packets can bypass conventional routing nodes and reach their destinations more efficiently. Compared with a fully router-based structure, the hybrid ring-mesh architecture exhibits good scalability and dynamic adaptability in terms of latency, throughput, and power consumption.

However, using ring structures alone still faces bandwidth bottlenecks and scalability limitations. As network size increases, the number of overlapping rings grows rapidly, making management complex. Therefore, combining mesh and ring to form a unified routing system becomes an effective compromise. This work builds upon this concept by further investigating methods to introduce structured ring interconnects into conventional mesh networks and exploring corresponding adaptive routing strategies to dynamically schedule multipath resources and enhance NoC performance.

### B. RL's Applicability to NoC Routing

In recent years, RL has demonstrated remarkable performance in architectural design, finding wide applications in task scheduling, resource allocation, and system adaptation [1], [15], [16], [18], [29]. We suggest that RL is naturally suited for routing decision-making in NoC systems, mainly for the following three reasons.

1) Decision learning in complex state spaces. NoC systems face high-dimensional state spaces, including node-to-node load, link congestion, and routing path status, all of which vary significantly over time and across communication patterns. RL can adaptively learn optimal strategies based on state-action feedback, dynamically selecting communication paths to reduce congestion and improve throughput.
2) Online learning capability. RL does not require offline training and can adapt in real time to changes in workload and network conditions. For heterogeneous multicore NoCs, an RL controller can continuously learn and adjust online, ensuring efficient data transmission under various topologies and communication pressures.

3) Convenient for integration and deployment. Unlike traditional machine learning models, RL methods are lightweight and require little hardware cost, which enables their deployment in resource-constrained on-chip systems. When combined with Q-learning-based policy networks, RL can be embedded within router microarchitecture to achieve intelligent routing control with low latency and power consumption.

Although more advanced RL algorithms, such as DQN or actor–critic methods, were also considered, they require neural-network inference hardware, resulting in significantly higher area, power, and decision latency. For on-chip router designs with strict timing and resource constraints, Q-learning offers the best balance between adaptability and hardware overhead. Therefore, this work adopts Q-learning as the core learning mechanism, enabling single-cycle lookup, small memory footprint, and seamless integration with the proposed ring-mesh-aware routing architecture.

### C. Theoretical Properties of the Proposed Ring-Augmented Mesh

To position the proposed topology relative to the conventional 2-D mesh and existing ring-augmented meshes, we briefly analyze several essential structural metrics.

*1) Average Hop Count:* For an $8 \times 8$ mesh, the expected Manhattan distance between two distributed nodes is

$$E\left[H_{\text{mesh}}\right] = \frac{2\left(N-1\right)}{3} = 6.67. \tag{1}$$

In the ring-mesh topology, traffic can be divided into two categories. For *local* communication occurring within a $4 \times 4$ subnet, packets are forwarded entirely through mesh links and retain the baseline behavior, yielding an average hop count of

$$E\left[H_{\text{local}}\right] \approx 2.67. \tag{2}$$

For *nonlocal* communication benefits from the express ring, a hybrid route consists of mesh segment to reach the nearest ring node, a single-hop traversal through the ring, and another mesh segment to reach the destination. Exhaustive enumeration shows that the average hop count for hybrid routes satisfies

$$E\left[H_{\text{hybrid}}\right] \approx 2.7\text{–}2.8 \tag{3}$$

which is significantly shorter than the direct Manhattan distance over the mesh. As a result, the overall average hop count of the ring–mesh lies between $E[H_{\text{local}}]$ and $E[H_{\text{hybrid}}]$, depending on the proportion of local versus nonlocal traffic.

*2) Worst Case Hop Count and Diameter:* The baseline $8 \times 8$ mesh has a diameter of 14 hops. In the ring–mesh, the worst case path requires at most four hops to reach the ring, one hop to traverse the express ring structure, and at most four hops to reach the destination

$$H_{\max} \leq 4 + 1 + 4 = 9. \tag{4}$$

Thus, the effective network diameter is significantly reduced relative to the mesh.

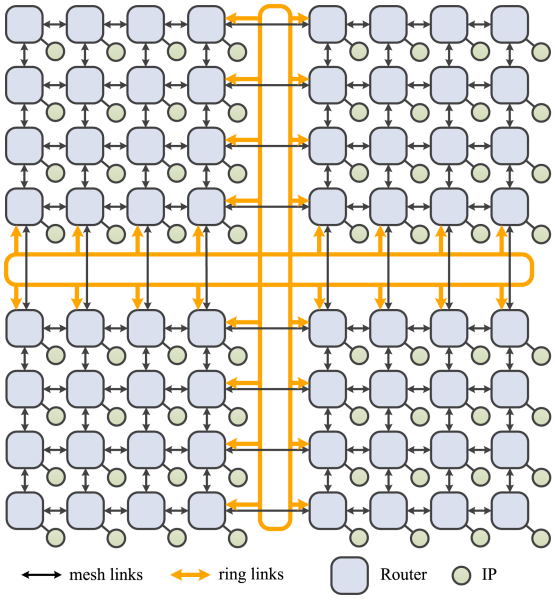


Fig. 2. Integration of mesh and ring networks.

*3) Bisection Bandwidth:* The central bisection of an $8 \times 8$ mesh contains eight bidirectional links. The added horizontal and vertical express rings contribute four additional links crossing the bisection. As a result, the bisection bandwidth can be expressed as

$$B_{\text{ringmesh}} = 12 \tag{5}$$

which represents a 50% increase over the mesh. This increase explains the more balanced load distribution observed in Fig. 1(b).

*4) Comparison With a Conventional Ring-Augmented Mesh:* A traditional ring-augmented mesh [15] connects only the nodes located on the ring; off-ring nodes must reach a ring interface via mesh routing. Our topology uses the same physical link set (mesh links + ring links) but allows unified mesh–ring hybrid routing. Consequently, all paths feasible in the conventional design remain available, while additional hybrid paths further reduce hop count and diameter. The bisection bandwidth remains identical because the number of ring links crossing the bisection is unchanged. The performance gains are achieved with modest additional control logic, as quantified in the hardware-overhead tables provided in Section VI.

## IV. Our Proposed RLRM System

### A. Overview of RLRM

In this work, we propose RLRM, a hybrid NoC topology, and its corresponding RL-based routing algorithm. As illustrated in Fig. 2, the core idea of RLRM is to introduce two high-speed ring topologies into the conventional 2-D mesh network. One ring, referred to as the horizontal ring, connects the central nodes along the 4th and 5th rows, while the other, referred to as the vertical ring, connects the central nodes along the 4th and 5th columns, thereby forming an enhanced ring-mesh hybrid topology.

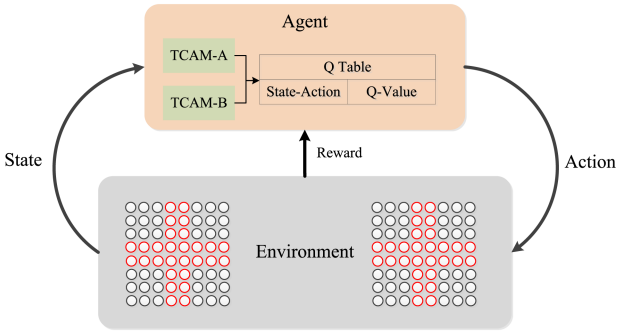


Fig. 3. RLRA framework.

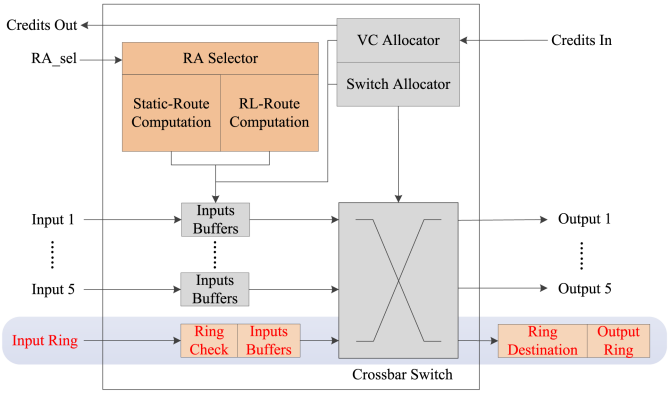


Fig. 4. Design of ring and mesh network node routers, using RA_sel signal to select routing algorithm.

This topology preserves the regularity and scalability of the mesh network while significantly enhancing communication capability in the central region through the introduction of high-speed ring interconnects. During path selection, packets can traverse conventional mesh routers. When conditions allow, they can instead use high-speed ring interconnects for rapid end-to-end delivery. This approach balances path flexibility with transmission efficiency. Under this structure, the load in the network center is effectively distributed, communication bottlenecks are alleviated, and overall system performance is substantially improved.

Fig. 3 presents an RL-based hybrid routing algorithm (RLRA). In the RLRA, each of the 64 nodes functions as an independent learning agent, maintaining its own Q-table. Each agent perceives the current network state through periodic local information collection and feedback from neighboring nodes. Upon receiving a packet, the agent selects the optimal action from its Q-table, deciding the next-hop routing port, which can be either a conventional mesh route or a high-speed ring path. After executing the action, the node computes an immediate reward based on network feedback and updates its Q-table, gradually optimizing its routing policy.

Through this architecture, RLRM achieves dual-level optimization: structural path improvement at the topology level and adaptive strategy learning at the algorithm level. This effectively mitigates congestion bottlenecks in the central region of traditional mesh networks and enhances communication performance and robustness in multicore NoCs under complex traffic patterns.

### *B. Ring-Augmented Mesh Topology*

To enhance communication efficiency in the central region of the NoC and reduce latency for medium and long-distance transmissions, this work proposes a hybrid structure that integrates local ring interconnects into a conventional $8 \times 8$ mesh. As shown in Fig. 2, the original mesh is preserved while a vertical ring is added between columns 4 and 5 and a horizontal ring between rows 4 and 5. These rings accelerate data movement in the center of the network, and unlike earlier ring enhanced meshes [15], the proposed ring–mesh allows routing paths to traverse both mesh and ring components, enabling more flexible and shorter communication paths.

The ring network adopts a bufferless point to point architecture with dedicated physical links, supporting single cycle forwarding and avoiding buffering, queuing, and arbitration overhead. When the source and destination lie on the same ring, packets can travel directly along the ring, reducing hop count and end to end latency.

The rings cover only the central $16 \times 2$ nodes, which limits the hardware cost while targeting the region where traffic is most concentrated. Experimental results indicate that this structure alleviates the central bottlenecks of a traditional mesh. All nodes retain their mesh connections, and routers are extended only with the required ring interfaces. Depending on destination and traffic conditions, the system may follow a mesh path, a ring path, or a hybrid path. The RL algorithm described in Section IV-C further refines this choice by adapting to congestion, improving routing flexibility and resource utilization.

When modifying the NoC topology, the corresponding updates to the router architecture are required. Traditional five port routers with four directional ports and one local port cannot support the newly added ring links. To enable communication over both mesh and ring structures, the router is extended with a ring in port and a ring out port, forming a seven port design. As illustrated in Fig. 4, nodes on the ring connect to dedicated physical links for packet reception and transmission, allowing the ring to operate alongside the mesh without disrupting conventional routing and preserving path flexibility in the hybrid topology.

The ring provides high-speed forwarding but has limited bandwidth due to its bufferless chained structure, and excessive traffic may create new bottlenecks. To balance performance and resource usage, a probabilistic routing mechanism is required to determine when ring traversal is beneficial. The ring–mesh shortest deterministic probabilistic routing algorithm, RSDR, implements this strategy and is given in Algorithm 1.

In RSDR, the usage of ring structures in path planning is controlled by a dynamic probability $P$. This mechanism is essential because the physical link resources of ring structures are limited. Without proper regulation, allowing all eligible traffic to pass through the ring may lead to congestion at ring

**Algorithm 1** Ring–Mesh Shortest Deterministic-Probabilistic Routing Algorithm

**Require:** Source node $S = (S_x, S_y)$, Destination node $D = (D_x, D_y)$
**Ensure:** Routing path from $S$ to $D$
**if** $S = D$ **then**
  **return** $[S]$
**end if**
Determine ring memberships:
$isHorizontalRing \leftarrow (S_y \in \{4,5\} \wedge S_x \in [1,8]) \vee (D_y \in \{4,5\} \wedge D_x \in [1,8])$
$isVerticalRing \leftarrow (S_x \in \{4,5\} \wedge S_y \in [1,8]) \vee (D_x \in \{4,5\} \wedge D_y \in [1,8])$
Generate random probability $p \sim U(0,1)$
$useRing \leftarrow (p \leq \frac{1}{1+e^{\alpha \cdot (L_{\text{ring}} - L_{\text{mesh}})}}) \wedge (isHorizontalRing \vee isVerticalRing)$
**if** $useRing$ **then**
  **if** $isHorizontalRing$ **and** $(S_y \in \{4,5\})$ **then**
    Route horizontally to $(D_x, S_y)$ via ring
    Apply XY routing from $(D_x, S_y)$ to $D$
  **else if** $isVerticalRing$ **and** $(S_x \in \{4,5\})$ **then**
    Route vertically to $(S_x, D_y)$ via ring
    Apply XY routing from $(S_x, D_y)$ to $D$
  **else if** $isHorizontalRing$ **and** $(D_y \in \{4,5\})$ **then**
    Apply XY routing from $S$ to $(S_x, D_y)$
    Route horizontally to $D$ via ring
  **else if** $isVerticalRing$ **and** $(D_x \in \{4,5\})$ **then**
    Apply XY routing from $S$ to $(D_x, S_y)$
    Route vertically to $D$ via ring
  **end if**
**else**
  Apply standard XY routing from $S$ to $D$
**end if**
**return** computed path

nodes, creating performance bottlenecks. To address this, we propose a dynamic calculation of $P$ based on real-time load using a logistic function

$$P = \frac{1}{1 + e^{\alpha \cdot (L_{\text{ring}} - L_{\text{mesh}})}} \tag{6}$$

where

$L_{\text{ring}} = R_{\text{req}}/R_{\text{cap}}$ represents the normalized load of the ring,
$L_{\text{mesh}} = M_{\text{req}}/M_{\text{cap}}$ represents the normalized load of the mesh, and
$\alpha$ is a tunable parameter (default value = 5).

This design enables the algorithm to automatically balance.

1) The low-latency advantage of the ring structure.
2) The distributed resources of the mesh network.

When the ring load is low ($L_{\text{ring}} \downarrow$), $P$ automatically increases to prioritize ring usage. Conversely, when the mesh load is relatively lower, traffic is guided through the mesh network. As demonstrated in Section V, this dynamic approach achieves 20.34% lower average latency compared to static routing strategies.

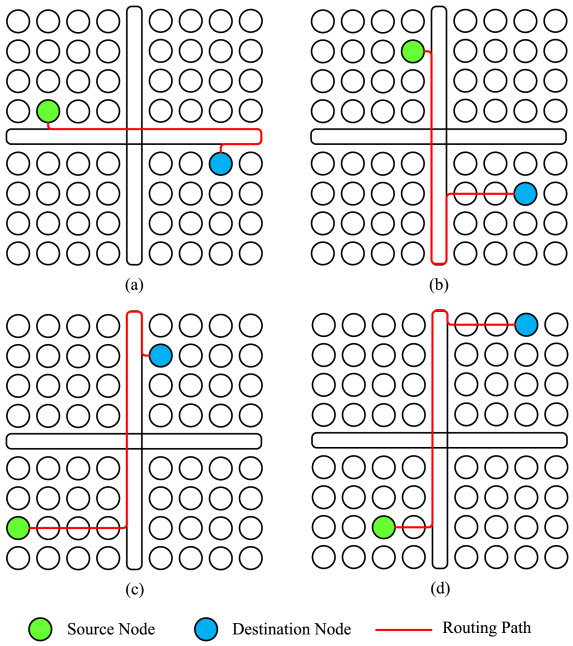


Fig. 5. Four typical scenarios of ring usage. (a) Both source ($S$) and destination ($D$) on the same ring. (b) $S$ on the ring and $D$ off the ring. (c) $D$ on the ring and $S$ off the ring. (d) Both $S$ and $D$ off the ring.

To demonstrate the algorithm, four representative scenarios of ring usage are considered, and their corresponding routing paths are shown in Fig. 5.

To support the proposed RSDR and the enhanced ring–mesh topology, the system introduces a certain amount of hardware resource overhead. As shown in Table I, the hardware overhead is mainly divided into topology enhancement and routing computation logic. In the topology part, the newly added horizontal and vertical rings introduce 28 bytes of bidirectional link overhead, along with 128 bytes of crossbar resources for ring interconnection. To enable the probabilistic decision-making mechanism, each node is equipped with a lightweight probability control unit, totaling 32 bytes.

In the routing computation part, the overhead includes a 2-bit flag for recording ring status (4 bytes), path selection logic (comparator + multiplexer, 32 bytes), and an $XY$-ring path selection state machine for hybrid decision-making (16 bytes). These components allow the system to dynamically switch between ring and mesh paths, while maintaining relatively low implementation complexity.

Compared with the baseline $8 \times 8$ mesh structure, Table II presents the overall overhead differences. The proposed design adds 128 bytes to the routing table, 112 bytes to ring buffers, and 36 bytes to control logic, resulting in a total increase of 276 bytes. This design achieves a substantial reduction in central communication latency and improved flexibility while maintaining relatively low overhead.

### C. RL-Based Hybrid Routing Algorithm

The escape VC always follows deterministic $XY$ routing and never uses ring links, thereby maintaining an acyclic channel dependency graph. Both the RSDR and the RLRA operate

TABLE I
HARDWARE OVERHEAD BREAKDOWN OF HYBRID RING–MESH DESIGN

| Category | Structure | Description | Size (Bytes) |
|---|---|---|---|
| Topology Enhancement | Horizontal Ring Interconnects | Bidirectional links (14 hops × 2 rings × 1B) | 28 |
| | Vertical Ring Interconnects | Bidirectional links (14 hops × 2 rings × 1B) | 28 |
| | Ring Crossbar | Additional 2-port routing (16 nodes × 2 ports × 4B) | 128 |
| | Probability Unit | 16 nodes × (8-bit LFSR + comparator) | 32 |
| Routing Computation | Ring Status Registers | 16 nodes × 2-bit flags (packed storage) | 4 |
| | Path Decision Logic | 16 nodes × (8-bit comparator + 8-bit MUX) | 32 |
| | XY-Ring Hybrid Logic | 16 nodes × 8-bit path selection FSM | 16 |

TABLE II
TOTAL OVERHEAD COMPARISON

| Metric | Baseline Mesh (B) | Proposed Design (B) | Overhead (B) |
|---|---|---|---|
| Routing Table Storage | 256 | 384 | +128 |
| Ring Buffers | 224 | 336 | +112 |
| Control Logic Storage | 48 | 84 | +36 |
| Total | 528 | 804 | +276 |

exclusively on nonescape VCs, meaning their adaptive or ring-based decisions cannot introduce cyclic dependencies. Since any packet can always fall back to the escape VC, global deadlock freedom is guaranteed even when adaptive routing selects ring paths.

To enhance the communication performance and adaptability of the hybrid ring–mesh topology under diverse traffic conditions, this work introduces an RLRA. The method deploys lightweight distributed agents at each node, enabling local decision-making and balancing the use of ring and mesh links. Compared with static or fixed-probability routing, RLRA aims to adjust ring–mesh usage according to traffic demand, alleviate queue buildup at hotspot links, and improve end-to-end latency and hop count through more efficient link utilization.

Each router maintains an independent Q-table and selects actions based on local observations and limited neighbor feedback, forming a distributed and scalable routing architecture. This design aligns with the local-control nature of NoCs, allowing rapid response to instantaneous congestion and avoiding the delays and bottlenecks associated with centralized decision-making. As each agent stores only state–action information relevant to its own ports, the state space remains compact, and the learning overhead is low. The distributed structure supports scalability, sustaining effective routing decisions as the system grows to large multicore configurations.

To capture the complete congestion status of each physical port, RLRM defines a 7-D state space for hybrid ring nodes

$$S = \left\{ \hat{Q}_L^{ring\_cw}, \hat{Q}_L^{ring\_ccw}, \hat{Q}_L^{E}, \hat{Q}_L^{S}, \hat{Q}_L^{W}, \hat{Q}_L^{N}, \hat{Q}_L^{local} \right\} \tag{7}$$

where

$$\hat{Q}_L^i = \frac{Q_L^i}{Q_{max}} \tag{8}$$

represents the normalized queue length of the corresponding port, ranging from 0 to 1. Here, $Q_L^{ring\_cw}$ and $Q_L^{ring\_ccw}$ denote the queue lengths along the clockwise and counterclockwise directions of the ring, $Q_L^E$, $Q_L^S$, $Q_L^W$, and $Q_L^N$ correspond to the east, south, west, and north directions of the mesh, and $Q_L^{local}$ indicates the local injection/receive pressure. $Q_{max}$ represents the maximum buffer size per port.

This design enables nodes to accurately perceive congestion at each interface, supporting topology-aware path selection.

The action space of a hybrid ring node includes all physical output ports, totaling seven possible actions

$$A = \{a_i \mid a_i \in \{to\_ring\_cw, to\_ring\_ccw, to\_E, to\_S, to\_W, to\_N, to\_local\}\} \tag{9}$$

where $to\_ring\_cw$ and $to\_ring\_ccw$ forward packets along the ring, $to\_E$, $to\_S$, $to\_W$, and $to\_N$ forward along mesh directions, and $to\_local$ handles local injection or termination. This mapping prevents invalid actions and enhances the efficiency of the learning process.

To balance transmission latency, ring load balance, and congestion control, RLRA defines the reward function

$$r_t = w_1 \cdot (-Latency_t) + w_2 \cdot U_{ring\_balance,t} - w_3 \cdot C_{congestion,t} \tag{10}$$

where

$$U_{ring\_balance} = 1 - \frac{\left| Q_L^{ring\_cw} - Q_L^{ring\_ccw} \right|}{Q_L^{ring\_cw} + Q_L^{ring\_ccw} + \epsilon} \tag{11}$$

and

$$C_{congestion} = \max_i \hat{Q}_L^i \quad i \in \{ring\_cw, ring\_ccw, E, S, W, N, local\}. \tag{12}$$

$U_{ring\_balance}$ is the ring utilization balance and $C_{congestion}$ represents the maximum congestion. The weight parameters $w_1$, $w_2$, and $w_3$ are listed in the parameter table.

The lookup algorithm is shown in Algorithm 2. RLRA adopts an asynchronous distributed Q-learning architecture, where each node maintains an independent Q table. The Q-values are updated as follows:

$$Q(s,a) \leftarrow Q(s,a) + \eta \left[ R + \gamma \max_{a'} Q\left(s', a'\right) - Q(s,a) \right]. \tag{13}$$

The initial learning rate is set to $\eta_0$, decaying exponentially to $\eta_{min}$. The discount factor is $\gamma = 0.8$, and the Q values are limited to the range $[-5, 15]$. Periodic neighbor averaging

**Algorithm 2** RL-Based Hybrid Routing at Node N

**Require:** Packet $p$ with destination $D$, local state $s$
**Ensure:** Output port selection
**if** $N = D$ **then**
  Deliver $p$ locally
**end if**
$s \leftarrow$ GetState($N$, $D$)
**if** rand() $< \epsilon$ **then**
  $a \leftarrow$ RandomEligibleAction($N$)
**else**
  $a \leftarrow$ TCAM_Lookup($Q\_table$, $s$)
**end if**
ExecuteRouting($a$)
Observe next state $s'$ and reward $R$
**if** $N$.idle_cycle **then**
  Update $Q(s, a)$ via Eq.(4)
**end if**

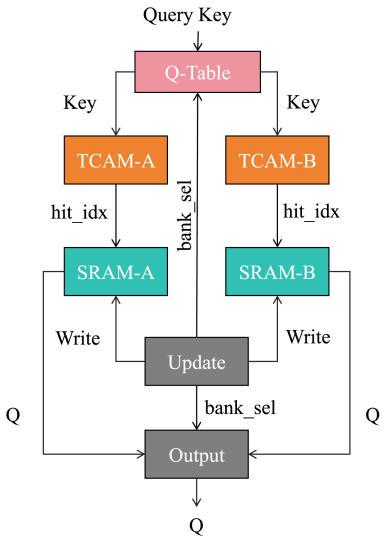


Fig. 6. Illustration of TCAM-SRAM dual-buffer system.

takes place every $T_{sync}$ cycles. The $\epsilon$-greedy policy starts with $\epsilon = 0.3$, with a ring action exploration probability of 0.4. The value of $\epsilon$ decays linearly to $\epsilon_{min} = 0.05$, and only topology-reachable valid actions are considered.

To meet NoC timing constraints, an 18-bit state and 3-bit action are encoded in a TCAM-SRAM dual-buffer system, as shown in Fig. 6. Two independent TCAM/SRAM banks (A/B) and a 1-bit pointer register are used. TCAM-A/SRAM-A serves queries, while TCAM-B/SRAM-B updates in the background. After writing, the pointer atomically flips, activating the new bank without interrupting queries. The query latency corresponds to the duration of writing the complete table and performing the flip, while the design requires only minimal area overhead (a 1-bit pointer and a small MUX), thus enabling zero-downtime Q-table upgrades.

The parameter settings for RLRA are shown in Table III, while the storage and latency overhead of RLRM are provided in Table IV.

TABLE III
RLRA PARAMETERS

| Symbol | Description | Value |
|---|---|---|
| $Q_{max}$ | Max queue length per port (flits) | 8 |
| $\eta_0$ | Initial learning rate | 0.1 |
| $\eta_{min}$ | Min learning rate | 0.01 |
| $\gamma$ | Discount factor | 0.8 |
| $Q_{min}$ | Min Q value | -5 |
| $Q_{max}$ | Max Q value | 15 |
| $T_{sync}$ | Neighbor sync period | 100 |
| $\epsilon_0$ | Initial exploration rate | 0.3 |
| $\epsilon_{min}$ | Min exploration rate | 0.05 |
| $\Delta t_{ring}$ | ring broadcast interval (cycles) | 10 |
| $\Delta t_{mesh}$ | mesh broadcast interval (cycles) | 20 |

RLRM employs a hybrid strategy switching mechanism. Under low traffic, nodes use probabilistic action selection to reduce power consumption, while under high traffic, RL selects optimal paths. The ring exploration probability is

$$\epsilon_{ring} = \epsilon_0 + \Delta\epsilon, \quad \Delta\epsilon = \frac{ring\_ports}{total\_ports} \cdot \epsilon_0. \tag{14}$$

This ensures sufficient ring exploration during early training and adaptively switches routing strategies based on traffic load, balancing energy efficiency and performance.

### *D. Implementation Strategies in High-Performance Application Scenarios*

Motivated by the demands of bandwidth and latency sensitive applications, such as high-performance computing and AI accelerators, this work proposes three scalable topology enhancement strategies built on the baseline $8 \times 8$ ring–mesh [Fig. 7(a)], which contains one horizontal and one vertical rings. These extensions aim to improve on-chip communication while maintaining controllable hardware overhead.

*1) Inwardringmesh:* As illustrated in Fig. 7(b), an additional set of rings is placed at the same locations as the baseline rings, forming a dual-ring structure. The two rings operate independently, and routers replicate ring-related buffers and arbitration logic. A 2-bit ring identifier and a 2:1 selector are added per port, doubling instantaneous bandwidth with minimal wiring. This topology benefits workloads with predictable communication and unidirectional traffic convergence.

*2) Horizontalringmesh:* As shown in Fig. 7(c), two extra horizontal rings are added above and below rows 4 and 5, and two extra vertical rings are added beside columns 4 and 5, forming a triple horizontal and vertical ring band. The additional high-speed paths reduce hop count for long-distance multicast communication and help disperse central hot spots. Its regular banded structure also simplifies layout and routing, making it well suited for row-dominated and column-dominated computation patterns.

*3) Annulusmesh:* As depicted in Fig. 7(d), four layers of concentric rectangular rings are deployed on the $8 \times 8$ mesh, covering the central four nodes and then 12, 20, and 28 nodes. This hierarchical design supports communication with diverse

TABLE IV
STORAGE AND LATENCY OVERHEAD OF RLRM

| Module | Description | Storage Overhead | Query/Update Latency |
|---|---|---|---|
| TCAM | 18-bit state + 3-bit action, parallel query | 14 kbit | 2 cycles |
| SRAM | Stores Q values corresponding to TCAM entries | 8 kbit | 2 cycles |
| Dual-Buffer Flip | Zero-downtime Q table upgrade | 1-bit + MUX | Atomic flip |
| Broadcast | ring/mesh ports broadcast | Queue info | 10-20 cycles |

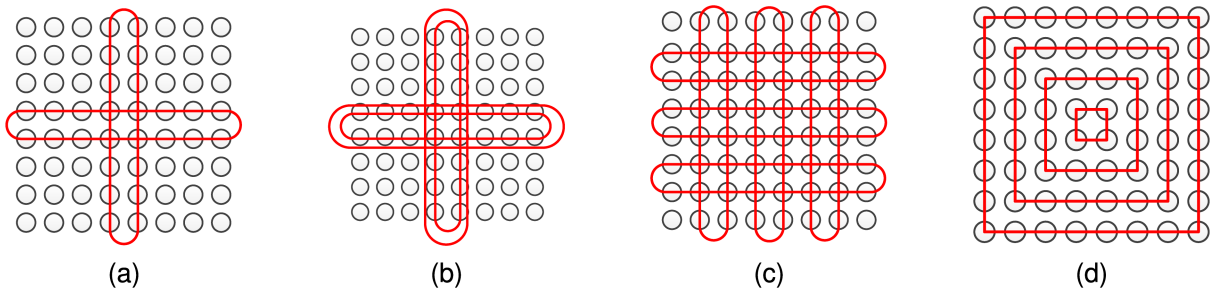


Fig. 7. Optimized ring layouts in ring–mesh for high-performance scenarios. (a) Ring–mesh, (b) inwardringmesh, (c) horizontalringmesh, and (d) annulusmesh.

TABLE V
HARDWARE OVERHEAD OF ENHANCED RING-BASED TOPOLOGIES

| Topology Configuration | Additional Link Buffer (B) | Additional Control Logic (B) |
|---|---|---|
| inwardringmesh | 56 | 32 |
| horizontalringmesh | 112 | 64 |
| annulusmesh | 196 | 128 |

TABLE VI
ASIC-LEVEL OVERHEAD COMPARISON OF BASELINE MESH AND RLRM

| Metric | Baseline | RLRM | Increase (%) |
|---|---|---|---|
| Area ($mm^2$) | 0.770 | 0.807 | +0.037 (4.8%) |
| Dyn. Power (mW) | 112.4 | 119.4 | +7.0 (6.2%) |
| Leakage (mW) | 9.6 | 9.9 | +0.3 (3.1%) |
| Critical Path (ns) | 1.23 | 1.29 | +0.06 (4.9%) |

distances and offers improved fault tolerance. Inner rings handle short range traffic, while outer rings serve long range flows, reducing average hop count. Redundant inner paths further enhance reliability under failures.

Table V summarizes the hardware overhead of the enhanced ring-based topologies relative to the baseline dual-ring design. The inwardringmesh topology adds only 56B of link buffer and 32B of control logic, offering a balanced tradeoff between performance gain and resource cost. The horizontalringmesh variant introduces 112B of additional link buffer and 64B of control logic, extending acceleration in both horizontal and vertical directions. The annulusmesh topology, which incorporates four concentric rings, results in the largest increase with 196B of link buffer and 128B of control logic and is well suited for hierarchical and collective communication. These results demonstrate a scalable design space that enables selecting topologies according to application demands and available hardware budgets.

### E. ASIC-Level Hardware Overhead Analysis

While Tables I and II summarize the byte-level resource increase introduced by the proposed ring–mesh topology, a more practical evaluation requires ASIC-level assessment. Using a 28-nm CMOS standard-cell library with typical SRAM and logic densities, the additional 276B of storage together with the extended control logic results in an estimated area increase of 0.037 $mm^2$. As reported in Table VI, this corresponds to approximately 4.8% of the baseline $8 \times 8$ mesh router area of 0.77 $mm^2$.

Power estimation based on switching activity extracted from Garnet simulations indicates that the RLRA decision logic and the TCAM–SRAM dual-buffer introduce an additional 6.2% dynamic power and 3.1% leakage power. Timing analysis further shows a small increase in the router critical path, rising from 1.23 to 1.29 ns, which remains well within the 5-ns clock period used in our NoC configuration.

As summarized in Table VI, the hybrid topology and RL-based routing incur a modest hardware overhead, significantly lower than the 10%–25% increase commonly reported for bypass-enhanced or speculative-router NoCs, while delivering substantial improvements in latency and load balancing.

## V. EXPERIMENTS AND ANALYSIS

### A. Experimental Setup

To evaluate the proposed hybrid ring–mesh topology and the RL-based routing algorithm, we conducted experiments using the `gem5` simulator [32] with the Garnet3.0 NoC model [33]. The baseline is an $8 \times 8$ mesh with 64 routers, each providing five ports. In the enhanced design, two high-speed rings are added: one along rows 3 and 4 and one along columns 3 and 4, and routers on these rings are equipped with one or two extra ports according to their locations.

All routers use wormhole switching with two virtual channels per port and a buffer depth of eight flits, and the flit size is 128 bits. Baseline and proposed systems share the same clock

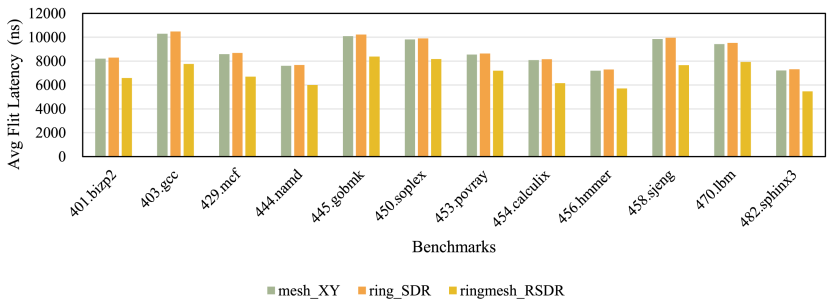


Fig. 8. Flit latency reduction of ring–mesh topology and its corresponding routing compared with mesh and ring.

TABLE VII
SIMULATION CONFIGURATION PARAMETERS

| Parameter | Value |
|---|---|
| Topology | $8 \times 8$ mesh, ring, ringmesh, inwardringmesh, horizontalringmesh, annulusmesh |
| Routing | XY, SDR, RSDR, RLRA, Switchover, LAMP, ReconfRing_NoC |
| VCs per Port | 4 |
| Buffer Depth | 8 flits/VC |
| Flit Size | 128 bits |
| Clock Period | 5 ns |
| Traffic Patterns | SPEC CPU2006, uniform, hotspot, transpose, bit complement |
| Simulation Cycles | $10^6$ warm-up + $10^7$ measurement |
| Metrics | Latency, Throughput, Energy |

frequency, injection rate, and buffer configuration to ensure fair comparison.

Workload behavior is evaluated using benchmark traces from SPEC CPU2006 [34]. Each simulation includes $10^6$ warm-up cycles and $10^7$ measurement cycles. The main metrics are packet latency, throughput, and energy consumption, and area cost is derived analytically from the added ports and the TCAM–SRAM routing structures. The key configuration parameters are listed in Table VII.

To complement real-application workloads and assess robustness under diverse communication pressures, we additionally include four representative synthetic traffic patterns—uniform random, hotspot, transpose, and bit complement. For these patterns, we report representative results at a moderate injection rate (0.10 flits/node/cycle), which is sufficient to reveal latency and congestion trends without requiring full injection-rate sweeps. Across these patterns, the proposed ring–mesh topology reduces average latency by 16%–27% relative to a conventional mesh, and RLRA further improves latency by 3%–6% over deterministic routing. These lightweight experiments confirm that the proposed topology and routing algorithm remain effective across both balanced and adversarial traffic conditions.

### B. Topology Structure Evaluation

Experiments were conducted on three topologies—the traditional bidirectional ring, the mesh, and the hybrid ring–mesh—to evaluate the performance advantage of ring–mesh under different routing strategies. Deterministic shortest-path routing and enhanced deterministic routing were applied across 12 SPEC CPU2006 programs, using average flit and packet latency as the primary metrics.

Under the mesh topology with *XY* routing, the 401.bzip2 benchmark has flit and packet latencies of 8204.40 and 8407.47 cycles, while the ring topology performs slightly worse at 8286.45 and 8519.62 cycles. The ring–mesh lowers latency to 6580.44 and 6751.23 cycles, achieving reductions of 19.79% relative to mesh and 20.61% relative to ring, demonstrating its effectiveness in mitigating transmission delay.

These improvements hold across applications. For the compute-intensive 403.gcc, flit latency decreases from 10294.58 in mesh and 10487.51 in ring to 7751.19 in ring–mesh, reductions of 24.71% and 26.07%. For the memory-intensive 429.mcf, flit latency decreases from 8589.20 in mesh and 8690.09 in ring to 6690.33 in ring–mesh, reductions of 22.09% and 23.00%. Although ring–mesh incurs a slight increase in power consumption, the latency gains provide clear overall performance benefits. Figs. 8 and 9 illustrate flit and packet latency comparisons.

We also observe a correlation between latency reduction and benchmark characteristics. Compute-intensive benchmarks and those with larger communication distances (e.g., 403.gcc) benefit more from the ring–mesh topology, as the ring paths reduce hop count and central congestion. Benchmarks with stronger locality (e.g., 456.hmmer) show smaller improvements because fewer packets require long-distance traversal.

### C. Routing Algorithm Performance

Based on the advantages of the ring–mesh topology, this section evaluates RSDR, RLRA, the dynamic switching strategy, and two state-of-the-art schemes, LAMP [10] and ReconfRing_NoC [15], using 12 SPEC CPU2006 benchmarks. The evaluation focuses on average flit latency, packet latency, and power consumption. RLRA consistently achieves the lowest latency among all routing strategies.

For 401.bzip2, RLRA reduces flit latency from 6580.44 to 6273.18 cycles, a 4.67% improvement, and packet latency from 6751.23 to 6430.04 cycles, a 4.76% improvement, with power rising from 0.728 to 0.794 W. LAMP records 7392.42 cycles, and ReconfRing_NoC 6495.42 cycles. For 403.gcc, RLRA reduces flit latency from 7751.19 to 7460.31 cycles, a 3.75% reduction, and packet latency from 7813.56 to 7524.90 cycles, a 3.69% reduction, while power increases from

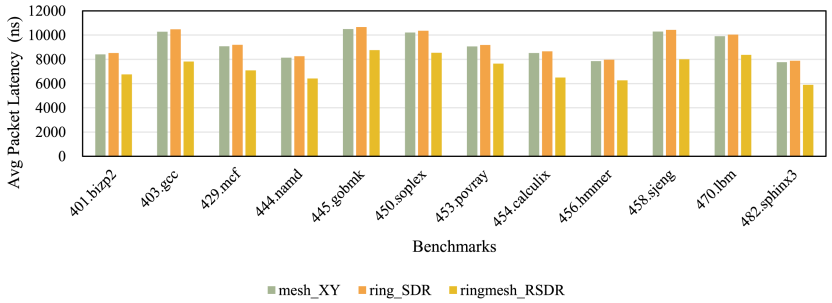


Fig. 9. Packet latency reduction of ring–mesh topology and its corresponding routing compared with mesh and ring.

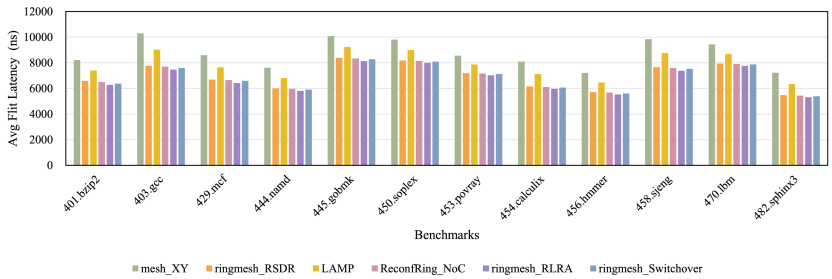


Fig. 10. Comparison of flit latency under different routing algorithms in ring–mesh topology.

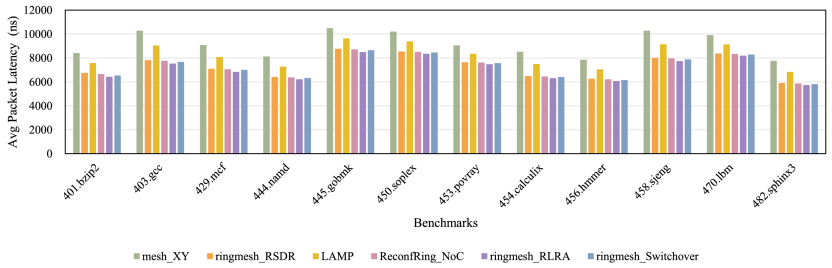


Fig. 11. Comparison of packet latency under different routing algorithms in ring–mesh topology.

0.915 to 0.974 W. LAMP is slower, and ReconfRing_NoC, although improved, remains inferior to RLRA. For 429.mcf, RLRA lowers flit latency from 6690.33 to 6422.07 cycles, a 4.01% reduction, with power rising from 0.746 to 0.803 W. LAMP yields the highest latency, while ReconfRing_NoC performs better than RSDR but below RLRA.

The dynamic switching strategy achieves latency close to RLRA with lower power. For 401.bzip2, switching provides 6367.89 flit latency, 6532.13 packet latency, and 0.764 W power, maintaining most RLRA benefits with reduced energy. For 470.lbm, switching reaches 7863.41 cycles and lowers power from 0.861 to 0.827 W.

Across all benchmarks, deterministic routing produces the lowest power but the highest latency. RLRA offers the best performance at increased power cost, while the switching strategy provides an effective latency–energy balance. Compared with LAMP and ReconfRing_NoC, both RLRA and the switching strategy consistently achieve lower flit and packet latency. Figs. 10 and 11 show the latency results, and Fig. 12 presents the power consumption.

A brief analysis of the power–latency tradeoff has been added. As shown in Fig. 12, RLRA lowers latency but raises power consumption because of increased switching on ring paths. The switchover strategy reduces this overhead by using deterministic routing under low traffic, achieving notable power savings while preserving most performance gains. These results provide clear tradeoff points for designers to balance latency and energy.

### D. Ring Network Layout in High-Performance Scenarios

To assess the impact of adding rings in high-performance scenarios, we extend ring–mesh to inwardringmesh, annulusmesh, and horizontalringmesh and evaluate them on 12 SPEC CPU2006 benchmarks under multiple routing strategies using flit latency, packet latency, and power.

Under RL-optimized routing, additional rings reduce latency. For 401.bzip2, ring–mesh achieves 6273.18 cycles, while inwardringmesh, annulusmesh, and horizontalringmesh reach 6032.41, 6101.38, and 5891.26 cycles, corresponding to reductions of 3.82%, 2.74%, and 6.10%, respectively. Power increases with added rings. Inwardringmesh, annulusmesh, and horizontalringmesh consume 0.853, 0.921, and 0.896 W, showing a clear latency–power tradeoff.

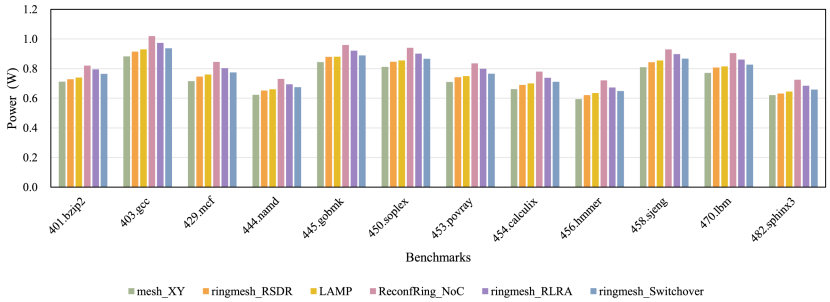


Fig. 12. Comparison of power under different routing algorithms in ring–mesh topology.

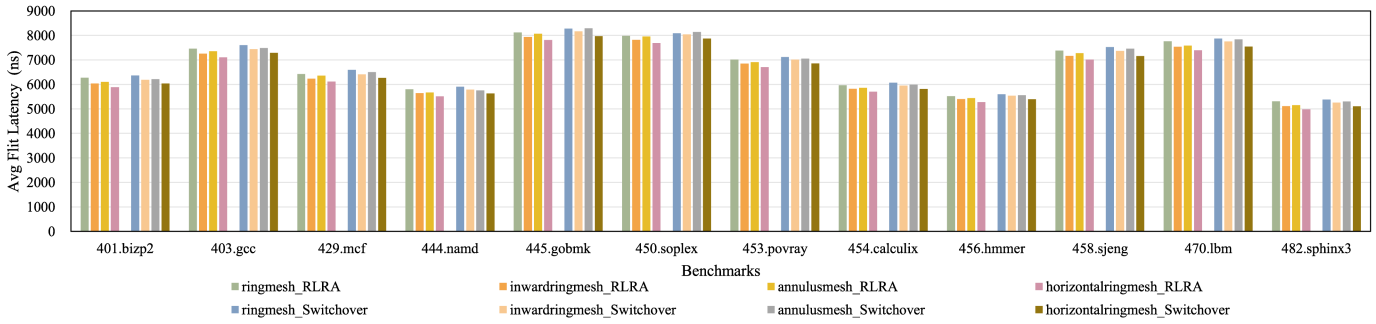


Fig. 13. Comparison of flit latency for topologies with different ring and mesh combinations.

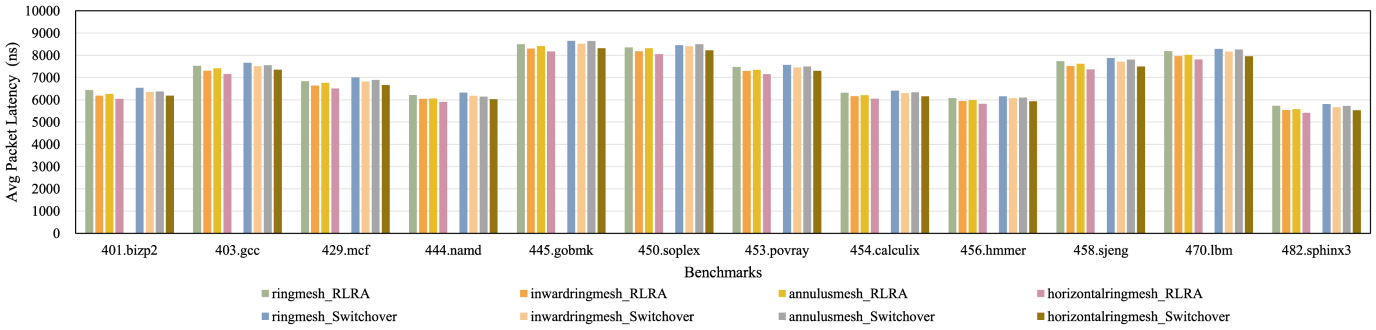


Fig. 14. Comparison of packet latency for topologies with different ring and mesh combinations.

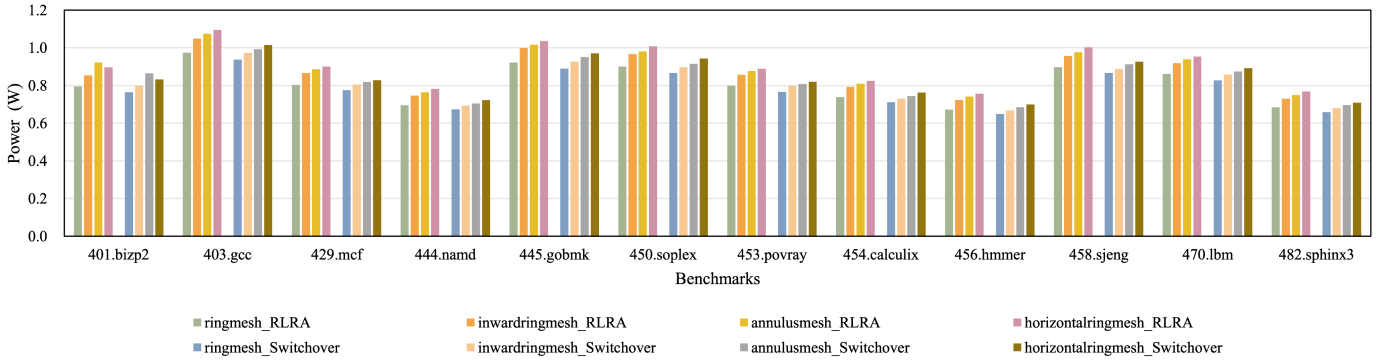


Fig. 15. Comparison of power for topologies with different ring and mesh combinations.

We also apply a traffic-adaptive switching strategy that uses deterministic routing at low load and RL routing under congestion. For 401.bzip2 on ring–mesh, switching yields 6367.89 cycles and 0.764 W. For inwardringmesh, annulusmesh, and horizontalringmesh, flit latency and power become 6187.61 cycles and 0.799 W, 6218.45 cycles and 0.864 W, and 6033.38 cycles and 0.832 W, respectively. Figs. 13 and 14 report latency, and Fig. 15 reports power.

Overall, more rings generally provide lower average latency but incur higher power. This trend is illustrated in Figs. 13–15.

We further observe a locality-dependent preference. Workloads with strong row and column affinity tend to favor horizontal-ringmesh, whereas workloads with highly distributed traffic favor annulusmesh. This suggests that topology selection can be guided by the communication profile when targeting specific latency and power budgets.

We note a benchmark-specific exception on 401.bzip2, where annulusmesh_RLRA exhibits slightly higher power than horizontalringmesh_RLRA, as shown in Fig. 15. This is because annulusmesh contains more multilevel ring links and RLRA may activate longer outer ring traversals for distributed traffic, which increases ring-link switching activity. To corroborate this explanation, we profiled the fraction of flits traversing ring links on 401.bzip2 and observed that annulusmesh_RLRA forwards about 15% more ring-traversing flits than horizontalringmesh_RLRA. This is consistent with the reported power trend.

### E. Scalability Discussion

Although the experimental evaluation in this work focuses on the widely adopted $8 \times 8$ (64-node) mesh, which is the standard configuration used by prior hybrid topology and RL-based NoC studies, such as LAMP [10], reconfigurable-ring NoC [15], and Adapt-NoC [21], the proposed RLRM architecture is inherently scalable.

The scalability of the design stems from both the structural characteristics of the ring-augmented topology and the locality of the RL-based routing mechanism. In an $N \times N$ mesh, the additional orthogonal rings continue to cover $\mathcal{O}(N)$ nodes, and the reduction in hop count for medium- and long-distance communication grows proportionally with network dimension. This means that the performance advantages introduced by the ring paths are preserved and can even become more pronounced, as the mesh size increases.

Furthermore, the hardware cost associated with the RL agent does not depend on the network size. As shown in Tables III and VII, each router maintains a small and fixed-size Q-table, resulting in $\mathcal{O}(1)$ storage per node and $\mathcal{O}(N^2)$ storage across the entire network, which is consistent with the complexity of conventional distributed routing architectures. Since the routing policy relies solely on local queue status and neighbor feedback, both the learning convergence behavior and the per-hop decision latency remain stable as the system scales.

Overall, these properties indicate that the proposed RLRM system can scale to larger NoC configurations without architectural changes. While experiments will be explored in future work, the theoretical and structural analysis supports the scalability of the design.

## VI. Conclusion and Future Work

This article presents the RLRM system, which is a hybrid NoC topology that combines high-speed ring interconnects with an *RL*-based routing algorithm. This design effectively addresses the load imbalance and hotspot issues found in traditional mesh networks. Through theoretical analysis and experimental evaluation, the proposed approach demonstrates significant improvements in both performance and energy efficiency. Regarding topological innovation, the proposed ring–mesh hybrid topology reduces average latency by 20.34% and achieves a maximum reduction of 24.71% compared to traditional mesh structures under SPEC CPU2006 benchmarks. In terms of intelligent routing, the RL-based mechanism employs a 7-D state space and a dynamic reward function to achieve adaptive load balancing across ring and mesh interconnects. Under the traffic-adaptive switching strategy, the system maintains latency improvements of 3.8%–6.1% while reducing power consumption by 3.9%–12.9% compared to pure RL routing. Compared with the state-of-the-art schemes, such as LAMP and ReconfRing_NoC, the proposed RLRM system consistently achieves lower latency across all workloads. These results highlight the advantages of jointly optimizing both topology and routing intelligence over methods that improve only routing or only topology. For high-performance scaling scenarios, the proposed homogeneous dual-ring, horizontal–vertical triple-ring, and concentric annulus structures further reduce latency by 6.1% through multiring cooperation, providing scalable optimization solutions for diverse application scenarios.

Future work will improve the RL model's learning speed and decision efficiency using attention-based networks and supervised–RL methods. Hardware work will optimize the ring–mesh topology, including ring arbitration, interface controllers, and layout design.

## References

[1] H. Yang, J. Fang, Y. Hou, X. Su, and N. N. Xiong, "Reinforcement learning-driven adaptive prefetch aggressiveness control for enhanced performance in parallel system architectures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 36, no. 5, pp. 977–993, May 2025.

[2] S. Das, C. Karfa, and S. Biswas, "Formal modeling of network-on-chip using CFSM and its application in detecting deadlock," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 4, pp. 1016–1029, Apr. 2020.

[3] W. Wang, X. Wang, J. Wang, N. N. Xiong, S. Cai, and P. Liu, "Ensuring cryptography chips security by preventing scan-based side-channel attacks with improved DFT architecture," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 3, pp. 2009–2023, Mar. 2022.

[4] J. Spieck, S. Wildermann, and J. Teich, "On transferring application mapping knowledge between differing MPSoC architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4289–4300, Nov. 2022.

[5] Z. Jiang et al., "NPRC-I/O: An NoC-based real-time I/O system with reduced contention and enhanced predictability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 12, pp. 4629–4642, Dec. 2023.

[6] A. Cakin, S. Dilek, and S. Tosun, "Energy-aware application mapping methods for mesh-based hybrid wireless network-on-chips," *J. Supercomput.*, vol. 80, no. 11, pp. 15582–15612, Jul. 2024.

[7] P. Papaphilippou and T. Van Chu, "Efficient deadlock avoidance for 2-D mesh NoCs that use OQ or VOQ routers," *IEEE Trans. Comput.*, vol. 73, no. 5, pp. 1414–1426, May 2024.

[8] R. Ma, J. Huang, S. Zhang, Y. Xie, and G. Luo, "NoCFuzzer: Automating NoC verification in UVM," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 44, no. 1, pp. 371–384, Jan. 2025.

[9] B. N. K. Reddy, M. Zia Ur Rahman, and A. Lay-Ekuakille, "Enhancing reliability and energy efficiency in many-core processors through fault-tolerant network-on-chip," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 5, pp. 5049–5062, Oct. 2024.

[10] H. Chen, P. Chen, X. Luo, S. Huai, and W. Liu, "LAMP: Load-balanced multipath parallel transmission in Point-to-Point NoCs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5232–5245, Dec. 2022.

[11] Z. Kang et al., "LSM-based hotspot prediction and hotspot-aware routing in NoC-based neuromorphic processor," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 32, no. 7, pp. 1239–1252, Jul. 2024.

[12] J. Fang, Z. Wei, Y. Liu, and Y. Hou, "TB-TBP: A task-based adaptive routing algorithm for network-on-chip in heterogenous CPU-GPU architectures," *J. Supercomput.*, vol. 80, no. 5, pp. 6311–6335, Mar. 2024.

[13] I. Pérez, E. Vallejo, and R. Beivide, "S-SMART++: A low-latency NoC leveraging speculative bypass requests," *IEEE Trans. Comput.*, vol. 70, no. 6, pp. 819–832, Jun. 2021.

[14] H. Farrokhbakht, P. V. Gratz, T. Krishna, J. S. Miguel, and N. Enright Jerger, "Stay in your lane: A NoC with low-overhead multi-packet bypassing," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Apr. 2022, pp. 957–970.

[15] L. Wang, L. Liu, J. Han, X. Wang, S. Yin, and S. Wei, "Achieving flexible global reconfiguration in NoCs using reconfigurable rings," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 611–622, Mar. 2020.

[16] T.-R. Lin, D. Penney, M. Pedram, and L. Chen, "A deep reinforcement learning framework for architectural exploration: A routerless NoC case study," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2020, pp. 99–110.

[17] A. Ejaz and I. Sourdis, "FastTrackNoC: A NoC with FastTrack router datapaths," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Apr. 2022, pp. 971–985.

[18] H. Chen, P. Chen, J. Zhou, L. H. K. Duong, and W. Liu, "ArSMART: An improved SMART NoC design supporting arbitrary-turn transmission," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 5, pp. 1316–1329, May 2022.

[19] P. Chen et al., "Contention minimization in emerging SMART NoC via direct and indirect routes," *IEEE Trans. Comput.*, vol. 71, no. 8, pp. 1874–1888, Aug. 2022.

[20] D. Xu, Y. Ouyang, W. Zhou, Z. Huang, H. Liang, and X. Wen, "RMC_NoC: A reliable on-chip network architecture with reconfigurable multifunctional channel," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 12, pp. 2061–2074, Dec. 2023.

[21] H. Zheng, K. Wang, and A. Louri, "Adapt-NoC: A flexible network-on-chip design for heterogeneous manycore architectures," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Feb. 2021, pp. 723–735.

[22] P. Iff, M. Besta, M. Cavalcante, T. Fischer, L. Benini, and T. Hoefler, "Sparse Hamming graph: A customizable network-on-chip topology," in *Proc. 60th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2023, pp. 1–6.

[23] Y. Xiao, S. Nazarian, and P. Bogdan, "Plasticity-on-chip design: Exploiting self-similarity for data communications," *IEEE Trans. Comput.*, vol. 70, no. 6, pp. 950–962, Jun. 2021.

[24] H. Chen, Z. Zhang, P. Chen, X. Luo, S. Li, and W. Liu, "MARCO: A high-performance task m apping a nd r outing Co-optimization framework for point-to-point NoC-based heterogeneous computing systems," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 5s, pp. 1–21, Oct. 2021.

[25] M. Safari, Z. Shirmohammadi, N. Rohbani, and H. Farbeh, "LETHOR: A thermal-aware proactive routing algorithm for 3D NoCs with less entrance to hot regions," *J. Supercomput.*, vol. 78, no. 6, pp. 1–25, Apr. 2022.

[26] A. Almeida da Silva, L. Nogueira, A. Coelho, J. A. N. Silveira, and C. Marcon, "Securet3d: An adaptive, secure, and fault-tolerant aware routing algorithm for vertically–partially connected 3D-NoC," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 33, no. 1, pp. 275–287, Jan. 2025.

[27] C. Sun, Y. Ouyang, and H. Liang, "Architecting a congestion pre-avoidance and load-balanced wireless network-on-chip," *J. Parallel Distrib. Comput.*, vol. 161, pp. 143–154, Mar. 2022.

[28] H. Sun, B. Yuan, N. N. Xiong, J. Song, W. Ding, and Q. Liu, "MDRL-ETT: A multiagent deep reinforcement learning-enhanced transmission tomography system to detect anomalous geological structures," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 10, pp. 6205–6217, Oct. 2024.

[29] Y. Xue et al., "AOME: Autonomous optimal mapping exploration using reinforcement learning for NoC-based accelerators running neural networks," in *Proc. IEEE 40th Int. Conf. Comput. Design (ICCD)*, Oct. 2022, pp. 364–367.

[30] J. Fang, H. Cheng, Y. Wang, and R. Zhai, "DRCD: A regional-contention-driven arbitration policy for CPU–GPU heterogeneous systems," *J. Supercomput.*, vol. 81, no. 3, p. 473, 2025.

[31] D. Merchán, M. Winkenbach, and A. Snoeck, "Quantifying the impact of urban road networks on the efficiency of local trips," *Transp. Res. A, Policy Pract.*, vol. 135, pp. 38–62, May 2020.

[32] Y. M. Qureshi, W. A. Simon, M. Zapater, K. Olcoz, and D. Atienza, "Gem5-X: A many-core heterogeneous simulation platform for architectural exploration and optimization," *ACM Trans. Archit. Code Optim.*, vol. 18, no. 4, pp. 1–27, Dec. 2021.

[33] S. Chen et al., "Floorplet: Performance-aware floorplan framework for chiplet integration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 43, no. 6, pp. 1638–1649, Jun. 2024.

[34] K. Rani and H. K. Kapoor, "Investigating frequency scaling, nonvolatile, and hybrid memory technologies for on-chip routers to support the era of dark silicon," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 4, pp. 633–645, Apr. 2021.

**Juan Fang** (Member, IEEE) received the M.S. degree from Jilin University of Technology, Changchun, China, in 1997, and the Ph.D. degree from the College of Computer Science, Beijing University of Technology, Beijing, China, in 2005.

In 1997, she joined the College of Computer Science, Beijing University of Technology. Since 2015, she has been a Professor at Beijing University of Technology, where she is currently with the College of Computer Science.

**Jingming Guo** (Student Member, IEEE) received the B.S. degree in computer science and technology from Beijing University of Technology, Beijing, China, in 2023, where he is currently working toward the M.S. degree in computer technology.

His main research interests include intelligent computing and high-performance computing.

Mr. Guo is a Student Member of CCF.

**Yubin He** (Student Member, IEEE) received the B.S. degree from Beijing University of Technology, Beijing, China, in 2023, where he is currently working toward the M.S. degree.

His main research interests include computer architecture.

Mr. He is a Student Member of CCF.

**Qi Ming** received the B.S. degree from the School of Automation, Beijing Institute of Technology (BIT), Beijing, China, in 2018, and the Ph.D. degree from BIT, in 2024.

He is an Associate Professor at the College of Computer Science, Beijing University of Technology (BJUT), Beijing. His research interests include edge intelligence, computer vision, and remote sensing.