ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Engineering Journals
Available on IEEE Xplore®

Research Article
https://doi.org/10.1631/ENG.ITEE.2025.0005

FTHOE: a Hamiltonian-driven fault-tolerant routing algorithm for wafer-scale interconnection networks

Shuaikang HOU[1], Qinrang LIU[2]✉, Wenbo ZHANG[1], Ping LV[1], Peijie LI[1], Wei GUO[1]

[1]*Information Engineering University, Zhengzhou 450001, China*
[2]*Institute of Big Data, Fudan University, Shanghai 200433, China*

Abstract: As application scenarios continue to grow in complexity, wafer-scale systems impose increasingly stringent requirements on the reliability of interconnection networks. Under inevitable process-induced manufacturing defects and environmental disturbances, node and link faults occur frequently in wafer-scale interconnection networks, making fault tolerance a key factor in improving overall system reliability. To address chiplet node faults and link faults in wafer-scale interconnection networks, this paper proposes a load-balancing virtual-channel-less fault-tolerant routing algorithm, termed FTHOE. The proposed algorithm is based on a Hamiltonian routing strategy and the odd–even turn model. By exploiting local fault vector information at the current node, FTHOE dynamically adjusts the output port selection priority, thereby shortening detour paths around faulty regions while effectively reducing the probability of packets being trapped in fault neighborhoods. At the same time, FTHOE preserves a relatively high degree of minimal path diversity by retaining the adaptiveness of Hamiltonian-based routing under fault conditions, thereby enhancing network load-balancing and overall communication performance. Simulation results demonstrate that, compared with existing fault-tolerant routing algorithms, FTHOE significantly reduces average network latency and improves throughput, exhibiting robust fault tolerance and load-balancing performance under complex fault scenarios.

Key words: Wafer-scale system; Fault-tolerant; Hamiltonian path; Odd–even turn model; Load balancing

1 Introduction

With the gradual slowdown of Moore's law (Moore, 1998) and the breakdown of Dennard scaling (Bohr, 2007), the traditional approach of achieving performance improvements solely through transistor scaling has become increasingly unsustainable. Conventional integrated circuit design is therefore confronted with multiple constraints simultaneously, including physical, packaging, and yield limits. Against this background, wafer-scale systems, which integrate a large number of prefabricated chiplets for computation, memory, and specialized acceleration on a single wafer, have gradually emerged as an important development direction for high-performance computing in the post-Moore era. Compared with single-chip solutions that rely on advanced process nodes, wafer-scale systems can achieve higher bandwidth density, lower communication latency, and better energy efficiency using more mature process technologies (Hu et al., 2024).

In recent years, both industry and academia have made significant progress in wafer-scale integration and emerging system architectures. For example, the wafer-scale engine (WSE) series (Pal et al., 2021) developed by Cerebras Systems integrates hundreds of thousands of computing cores on a single wafer and enables ultra-large-scale parallel computation through a high-bandwidth interconnection network. Tesla's Dojo system (Pal et al., 2019) constructs wafer-scale interconnects using a regular two-dimensional (2D) topology, achieving efficient inter-chiplet communication. In addition, a software-defined system on wafer (SDSoW) (Wu et al., 2024) reconstructs the system design paradigm through hardware–software co-design, providing a new technological path for large-scale heterogeneous integration. These representative systems collectively indicate that the interconnection network has become a key infrastructure determining the performance, energy efficiency, and reliability of wafer-scale systems.

✉ Qinrang LIU, qinrangliu@sina.com
Shuaikang HOU, https://orcid.org/0009-0000-4973-0563
Qinrang LIU, https://orcid.org/0000-0002-9957-7365
Wenbo ZHANG, https://orcid.org/0009-0000-6542-9797
Ping LV, https://orcid.org/0009-0008-1608-6597
Peijie LI, https://orcid.org/0009-0002-6280-7857
Wei GUO, https://orcid.org/0000-0002-1023-7277
CLC number: TP393.03

Received: Aug. 29, 2025; Revision accepted: Feb. 2, 2026;
Crosschecked: Mar. 3, 2026

Compared with traditional networks-on-chip (NoCs), wafer-scale system interconnection networks exhibit significant differences in terms of scale, heterogeneity, and manufacturing yield (Jerger et al., 2014). On one hand, wafer-scale systems typically consist of hundreds or even thousands of prefabricated chiplets, resulting in an interconnection scale far beyond that of single-chip NoCs. On the other hand, due to unavoidable process variations and defect distributions during wafer-scale manufacturing, multiple permanent node or link faults may already be present at the deployment stage (Charif et al., 2016). Consequently, wafer-scale systems impose more stringent system-level requirements on the fault tolerance of interconnection networks, where even a single-point fault or localized fault may trigger cascading effects (Nehnouh and Senouci, 2019). Related fault-tolerant concepts have also been studied in wireless sensor networks, e.g., the partition-based energy-efficient low-energy adaptive clustering hierarchy (PE-LEACH) protocol; however, their system models are fundamentally different from those of the wafer-scale interconnection networks considered in this work (Mohapatra and Rath, 2019).

From a system abstraction perspective, wafer-scale interconnection networks differ significantly from traditional NoCs in terms of physical scale and application scenarios; however, they share a high degree of structural similarity at the communication and routing levels. Specifically, as illustrated in Fig. 1, chiplets or computing array nodes in a wafer-scale system can be mapped to routing nodes in an NoC, while wafer-scale interconnection links can be abstracted as regular physical channels. Moreover, the hop-by-hop forwarding mechanism and local routing decision process are functionally consistent with the NoC routing model. Based on this mapping, wafer-scale interconnection networks can be reasonably abstracted, at the routing and fault-tolerance levels, as large-scale, heterogeneous, and fault-intensive 2D mesh networks. Although their physical implementations differ from those of conventional NoCs in scale and application context, the two exhibit strong similarities in key issues such as routing modeling, turn constraints, and fault avoidance. Therefore, adopting a 2D mesh NoC as the routing abstraction model for wafer-scale interconnection networks is reasonable and of significant research value.

However, the high fault density and complex fault modes encountered in wafer-scale systems pose significant challenges to the direct application of conventional NoC routing algorithms. Chiplet node faults and interconnection link faults may not only degrade communication performance but also lead to path unreachability, traffic concentration, and even system-level communication faults. Therefore, how to enhance the reliability and performance of wafer-scale interconnection networks under mixed-fault scenarios through routing algorithms, without incurring significant hardware complexity, has become a critical issue that urgently needs to be addressed.

To this end, considering the large scale, high fault density, and stringent system-level reliability requirements of wafer-scale interconnection networks, we systematically model fault behaviors and communication demands at the routing level and propose a fault model tailored for wafer-scale networks to characterize the impact of node and link faults on communication reachability. On this basis, a novel virtual-channel (VC)-less fault-tolerant routing scheme, termed fault-tolerant routing algorithm based on Hamiltonian paths and the odd–even turn mode (FTHOE), is developed for wafer-scale networks. The design of FTHOE is motivated by the observation that the Hamiltonian odd–even (HOE) (Bahrebar and Stroobandt, 2015) routing algorithm achieves a near-maximal degree of adaptiveness among VC-less minimal routing schemes in fault-free networks. However, HOE was not originally designed to handle node or link faults, under which its routing flexibility and direction multiplicity can degrade rapidly. To address this limitation, FTHOE extends HOE by incorporating a lightweight fault-tolerant mechanism based on routing port-priority reordering under local fault vectors. This mechanism preserves the original deadlock-free turn constraints while enabling effective detouring around faulty regions with minimal overhead. By maintaining a larger set of feasible minimal and near-minimal paths in the presence of faults, FTHOE provides more alternative routing options than existing VC-less fault-tolerant schemes. In contrast, most existing schemes trade routing adaptiveness for fault tolerance by introducing additional turn restrictions or conservative routing policies. As a direct consequence of the increased path diversity, FTHOE further exhibits improved load-balancing capability under non-uniform traffic and fault scenarios. Comprehensive analysis and gem5-based simulations under various traffic patterns and fault distributions demonstrate that the proposed scheme significantly improves traffic distribution and communication performance without incurring substantial hardware complexity, offering a practical routing solution that jointly

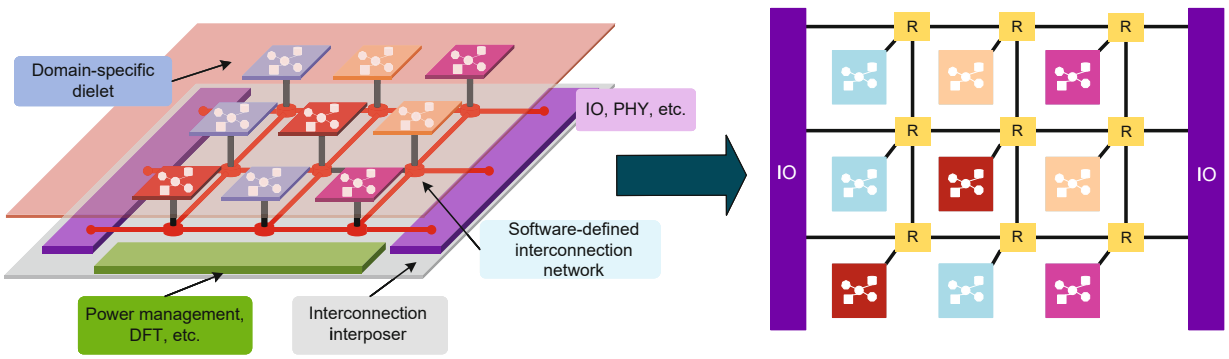

Fig. 1 Two-dimensional planar mapping of interconnection networks on SDSoW (IO: input/output; PHY: physical layer; DFT: design for testability)

considers reliability and performance for wafer-scale interconnection networks.

2 Related works

Although wafer-scale systems have attracted considerable attention in recent years, systematic research on fault-tolerant routing for on-wafer interconnection networks remains relatively limited. In particular, under conditions of large-scale heterogeneous integration and the coexistence of multiple faults, mature and unified theoretical frameworks for dynamic fault-tolerant mechanisms have yet to be established. Given the hop-by-hop communication characteristics exhibited by wafer-scale interconnects at the routing level, existing studies on fault-tolerant routing in NoCs provide important theoretical foundations and methodological references for the design of on-wafer interconnection networks.

In the field of wafer-scale system interconnection, existing studies have primarily focused on system architecture design, communication mechanism optimization, and manufacturing-yield awareness (Yang et al., 2025). Representative works such as the SDSoW architecture reconstruct the design paradigm of on-wafer systems through hardware–software co-design, emphasizing system-level reconfigurability and resource management capabilities, while systems such as Cerebras WSE and Tesla Dojo achieve ultra-large-scale parallel computing through regular 2D topologies and high-bandwidth interconnects. Although these studies have made significant progress in system architecture and communication bandwidth, interconnection reliability is typically addressed through static redundancy designs or higher-level task scheduling mechanisms, with relatively limited attention paid to dynamic fault-tolerant routing strategies under mixed-fault scenarios at the routing level. In particular, under the constraint of not introducing additional hardware redundancy, how to enhance the fault-tolerant and load-balancing capabilities of wafer-scale interconnection networks through routing algorithms remains insufficiently explored.

From an implementation perspective, fault-tolerant routing algorithms typically reflect a trade-off between hardware resource overhead and network performance. VC-based fault-tolerant routing approaches introduce multiple logical subchannels on top of physical channels, thereby effectively breaking channel dependency cycles and providing high path flexibility and load-balancing capability under complex fault conditions. The cost-aware fault-tolerant (CAFT) routing algorithm (Reza et al., 2019) introduces VCs along the Y direction and restricts specific turns to reduce routing decision complexity while ensuring deadlock-free operation. Other studies have proposed fully adaptive routing algorithms based on virtual network (VNet) partitioning (Zhang et al., 2021), deriving optimal VC allocation schemes and theoretically prove deadlock freedom. Joshi and Thakur (2023) further employed artificial neural networks to optimize VC allocation strategies, effectively improving network throughput and reducing communication latency. The tolerating both faulty links and routers (TFLR) algorithm, proposed by Ebrahimi and Daneshtalab (2015), is also based on VC allocation in the Y direction and always selects the shortest transmission path when a physical path exists. Although these VC-based methods exhibit significant advantages in improving network throughput and reducing deadlock risk, the additional buffering logic and control logic significantly increase hardware overhead and implementation complexity. In wafer-scale systems, where resource scale is extremely large and energy-efficiency constraints are stringent, VC-based approaches face practical challenges in terms of implementation complexity.

In contrast, VC-less fault-tolerant routing algorithms rely mainly on turn models to eliminate dependency cycles at the algorithmic level, thereby ensuring deadlock-free operation. A variety of turn-restriction mechanisms have been proposed in prior studies (Renani et al., 2022) and further extended to support fault avoidance. The FTOE turn model and the load-balancing fault-tolerant (LBFT) algorithm proposed by Xie et al. (2018) proactively detour faulty regions to shorten fault-tolerant paths and partially balance traffic around fault areas; however, their routing path diversity is relatively limited. Based on the FTOE turn model, Guan et al. (2023) proposed a low-overhead fault-tolerant routing algorithm that reduces the probability of packets entering faulty regions while providing a limited number of shortest-path options. The fault-route-NoC-mesh (FRNM) algorithm proposed by Rahaman et al. (2019) models multiple faulty nodes as convex faulty regions and employs adaptive routing strategies for detouring, but the construction of virtual boundaries introduces additional computational overhead. Although these approaches inherently reduce hardware complexity, they often suffer from restricted path diversity, load imbalance, and even aggravated local congestion under complex fault distributions.

Overall, existing research on wafer-scale systems has primarily focused on system architecture (Yu et al., 2025) and communication bandwidth optimization (Xu et al., 2025), while fault-tolerant routing studies in the NoC domain have accumulated rich methodologies for path selection and deadlock avoidance. However, most wafer-scale system studies lack systematic analysis of dynamic mixed-fault tolerance mechanisms at the routing level, whereas existing NoC fault-tolerant routing methods often struggle to simultaneously address path reachability, load balancing, and implementation complexity when directly applied to high fault-density wafer-scale systems. Motivated by this gap, the proposed FTHOE routing algorithm targets wafer-scale interconnection networks with high fault density and regular topological characteristics and enhances routing decision capability through a locally fault-aware priority reordering mechanism, without introducing VCs or additional hardware redundancy. As a result, FTHOE effectively improves path reachability and load-balancing capability under fault conditions while preserving deadlock-free operation.

3 Models

3.1 Hamiltonian routing strategy

In a 2D $m \times n$ mesh network topology, network nodes are precisely located using Cartesian coordinates. It can be demonstrated that each router node i ($0 \leq i < m \times n$) can be uniquely represented by the coordinates (x_i, y_i), where $x_i \in [0, m-1]$ represents the node position in the X direction,

and $y_i \in [0, n-1]$ represents the node position in the Y direction. To implement the Hamiltonian routing strategy, a snake-shaped numbering scheme is adopted for global node numbering. The numbering sequence L commences from the initial node $(0, 0)$. In even rows (where y_i is even), the nodes are numbered in ascending order from left to right, with x_i increasing. In odd rows (where y_i is odd), the nodes are numbered in descending order from right to left (with x_i decreasing), ensuring that the entire network forms a continuous, non-repeating Hamiltonian path. The specific numbering rules for each node can be defined as follows:

$$L_{(x_i,y_i)} = \begin{cases} y_i n + x_i, & \text{if } y_i \text{ is even,} \\ y_i n + n - x_i - 1, & \text{if } y_i \text{ is odd.} \end{cases} \quad (1)$$

In an 8×8 mesh network topology, based on the node numbering system of the Hamiltonian routing strategy, the network can be divided into two logically independent subnetworks: high-channel (H_{H}) subnetwork and low-channel (H_{L}) subnetwork. Within the H_{H} subnetwork, data packets are transmitted strictly in ascending order of node numbers. Conversely, within the H_{L} subnetwork, data packets are transmitted following a strict ascending order of node numbers. These transmission patterns are illustrated by the yellow and blue arrows in Fig. 2, separately. This strict unidirectional transmission mechanism ensures that no circular paths exist in the network topology, thereby fundamentally eliminating the possibility of cyclic dependencies. Consequently, the routing strategy based on Hamiltonian paths is both deadlock-free and livelock-free.

3.2 Fault models and fault perception

Faults in on-chip networks typically occur at router nodes or their connecting links. According to the type of the faulty component, our study considers two basic fault models: node fault model and link fault model. Under a coarse-grained fault modeling framework, a node fault exhibits cascading fault characteristics, whereby the faulty router itself becomes unavailable, and all its associated links are also regarded as failed. In contrast, a link fault only blocks direct communication in the corresponding direction, while the remaining links continue to operate normally. As illustrated in Fig. 3, in an 8×8 mesh topology, network nodes can be classified into faulty nodes, unsafe nodes, and safe nodes based on their proximity to the fault location. Unsafe nodes are defined as nodes that are directly adjacent to faulty nodes.

Fault-tolerant routing algorithms rely on the ability to perceive fault information to make reliable forwarding decisions. According to the visibility scope of fault information, fault-awareness mechanisms can generally be classified into two categories: global fault awareness and local fault awareness. Global fault awareness provides a complete view of the network fault topology but typically incurs high storage, communication, and maintenance overheads and may introduce additional synchronization complexity; in contrast, local fault awareness maintains only the health status of the current router and its directly adjacent nodes or links. Although it lacks foresight regarding remote faults, it offers significant advantages in terms of implementation complexity and hardware overhead, making it more suitable for large-scale wafer-level interconnection systems.

This work adopts a local fault-awareness model and assumes that each router maintains only the health status of nodes or links in its four directly adjacent directions (north, east, south, and west). This information is represented in the form of a local fault vector and is used as an input to the routing decision at the current node to determine the availability

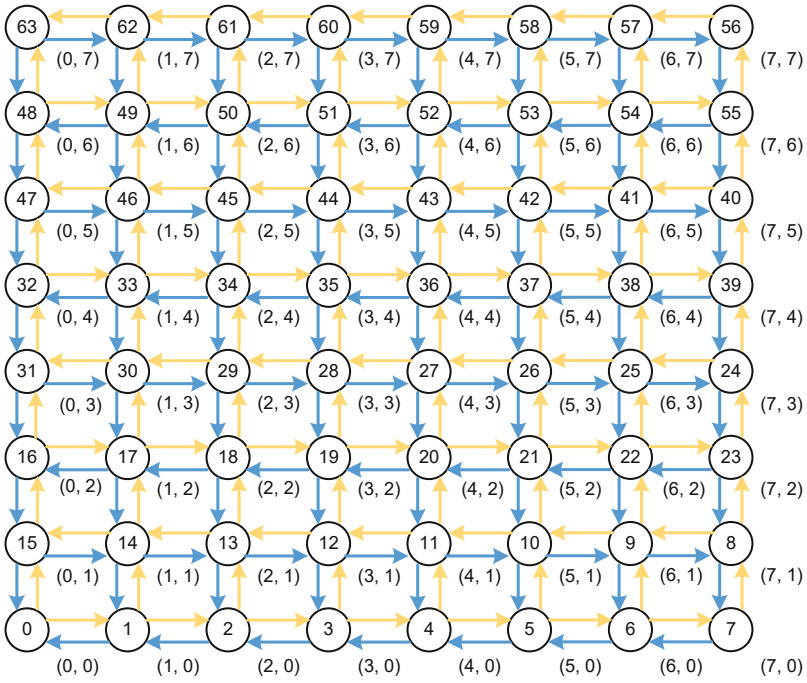

Fig. 2 Corresponding Hamiltonian node numbering and traversal paths in an 8×8 network

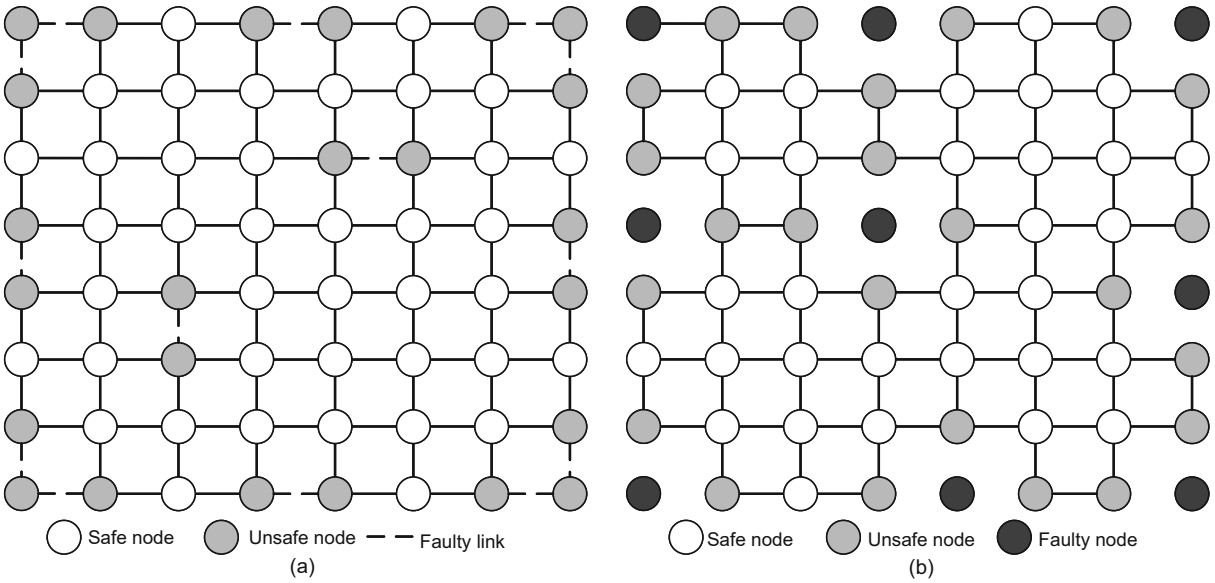

Fig. 3 Network fault models, where (a) is the link fault model and (b) is the node fault model, showing the two basic fault types considered in this work

of candidate output ports. The router does not rely on any global fault information and does not require awareness of fault states at nonadjacent nodes. Fig. 4 illustrates the adopted local fault-awareness model.

To support the construction of local fault vectors, each router in the network can be equipped with lightweight built-in self-test mechanisms, such as built-in self-test (BIST), test pattern generation (TPG), or test response analysis (TRA). It should be emphasized that the mechanisms discussed in this work are primarily intended for fault detection during offline testing or maintenance phases—such as system power-on initialization, periodic maintenance windows, or explicitly triggered diagnostic scenarios—rather than for cycle-by-cycle or high-frequency online fault monitoring. During normal operation, a router maintains only the fault flags of its directly neighboring nodes or links. This state can be refreshed through low-frequency updates, for example, upon detecting persistent communication anomalies, link unreachability, or after the completion of a maintenance phase. This work assumes that fault states are static or quasi-static in the temporal domain, so that local fault vectors do not change frequently during normal data transmission.

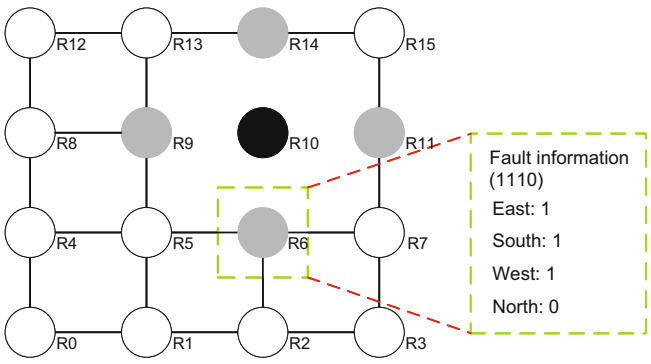

Fig. 4 Fault detection model illustrating a local 4-bit fault vector example

Under this model, the hardware overhead of the local fault vector is extremely low. Each router needs to maintain only a single 1-bit health flag for each of the four directions, resulting in constant storage overhead of $O(1)$. Updates to the fault vector do not participate in the normal data forwarding path, and their propagation or refresh frequency is far lower than the packet injection and routing computation rates. Therefore, they do not significantly impact the steady-state performance of the network. Since FTHOE uses the local fault vector only within local combinational logic for routing computation, any awareness latency is confined to the time window between fault occurrence and state update and does not affect the correctness of the algorithm.

3.3 Turn model

In VC-less NoC designs, turn models are a commonly adopted approach for deadlock elimination. The fundamental idea is to break channel dependency cycles by prohibiting a small subset of turns while preserving network connectivity. However, overly restrictive turn constraints can significantly reduce routing adaptiveness, which may in turn lead to load imbalance and performance degradation.

The Hamiltonian adaptive multicast unicast model (HAMUM) improves Hamiltonian-path-based routing by imposing different turn restrictions in the H_{H} and H_{L} subnetworks; however, it prohibits a relatively large number of turns, which may still limit path diversity under complex communication scenarios (Daneshtalab et al., 2011). In contrast, the HOE turn model (Bahrebar and Stroobandt, 2015) combines Hamiltonian numbering with the odd–even (OE) rule, prohibiting only the necessary turns within each potential channel dependency cycle while simultaneously disallowing 180° U-turns. As a result, HOE effectively eliminates deadlock under VC-less conditions while preserving a larger set of minimal-path candidates. In this work, the HOE turn model is adopted as the deadlock-free routing foundation, and its turn

constraints are illustrated in Fig. 5b.

The HOE rules are defined as follows:

Rule 1: East–south (ES) and north–west (NW) turns are prohibited in even rows.

Rule 2: North–east (NE) and west–south (WS) turns are prohibited in odd rows.

Under these rules, NE and ES turns cannot occur in the same row, and NW and WS turns cannot occur in the same row. As a result, no cyclic path can be formed along the top row, thereby avoiding deadlock. This design mitigates deadlock risks by eliminating the possibility of packets forming looping paths along the top row. Consistent with the traditional OE turn model, this scheme also prohibits 180° packet forwarding in the network. By prohibiting different turns in odd and even rows, it improves routing adaptiveness while ensuring deadlock elimination.

To quantitatively characterize the adaptiveness of different routing strategies under turn constraints, the degree of adaptiveness (DoA) is introduced as an evaluation metric. DoA describes the cardinality of the minimal-path candidate set permitted by a routing algorithm under deadlock-free constraints for a given communication state. Let the source node and the destination node be denoted as $S(X_\mathrm{s}, y_\mathrm{s})$ and $D(x_\mathrm{d}, y_\mathrm{d})$, respectively. In the ideal case of fully adaptive minimal routing without any turn restrictions, the theoretical upper bound of DoA between S and D is given by

$$\begin{aligned} &\mathrm{DoA}_{\mathrm{ideal}}(S,D) = \frac{(\Delta x + \Delta y)!}{\Delta x!\,\Delta y!}, \\ &\Delta x = \left|x_\mathrm{d} - x_\mathrm{s}\right|, \quad \Delta y = \left|y_\mathrm{d} - y_\mathrm{s}\right|, \end{aligned} \tag{2}$$

where Δx and Δy denote the horizontal and vertical hop distances between S and D, respectively. $\mathrm{DoA}_{\mathrm{ideal}}(S,D)$ therefore corresponds to the total number of shortest Manhattan paths between S and D, which equals the number of distinct permutations of Δx horizontal moves and Δy vertical moves.

The DoA of a specific routing algorithm is jointly determined by the turn constraints and the output port selection rules it adopts, and is generally lower than the combinatorial upper bound. Based on the relative position of the destination node with respect to the source node along the Hamiltonian path, unicast communication can be categorized into 16 distinct communication states, among which eight states in the H_H subnetwork are illustrated in Fig. 6. The DoA of the HOE routing algorithm under different communication states has been systematically derived in prior work; under fault-free, VC-less conditions, HOE can provide minimal-path diversity close to the theoretical upper bound (Bahrebar and Stroobandt, 2015).

Therefore, our study adopts the routing strategy under the HOE turn model as the baseline for the shortest-path DoA under fault-free conditions. It quantitatively compares, under unified geometric communication conditions (determined by the relative positions of the source and destination nodes), the shortest-path DoA at the packet injection stage between FTHOE, which is based on HOE, and oblivious and fault-tolerant routing (OFTR), which does not employ HOE turn constraints. The results are summarized in Table 1. Here, the terms "odd/even row" and "odd/even column" are evaluated based on the parity of the current node's coordinates at the packet injection stage, where the current node refers to the source node itself or a same-row node under HOE constraints. Fig. 6 illustrates the geometric relationships corresponding to different communication states and their associated turn-constraint structures, while Table 1 numerically instantiates the availability of shortest directions at the injection stage for representative distance configurations, thereby reflecting differences among routing strategies in terms of shortest-path reachability.

It should be emphasized that FTHOE is not intended to surpass HOE in terms of minimal-path diversity under fault-free conditions. Instead, it aims to preserve minimal-path reachability and an effective path-selection space as much as possible in the presence of faults. At the routing-decision

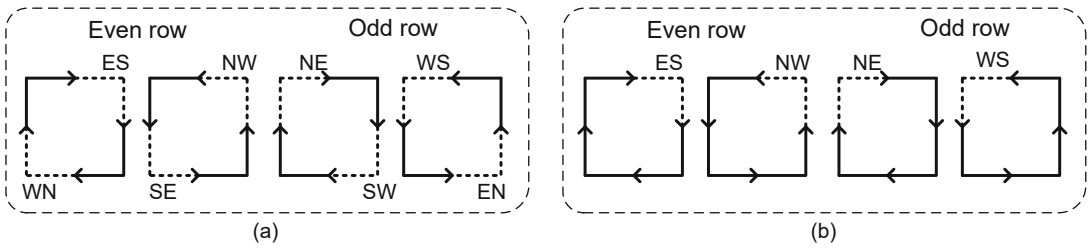

Fig. 5 Turn models illustrating the prohibited turns in odd and even rows: (a) HAMUM; (b) HOE

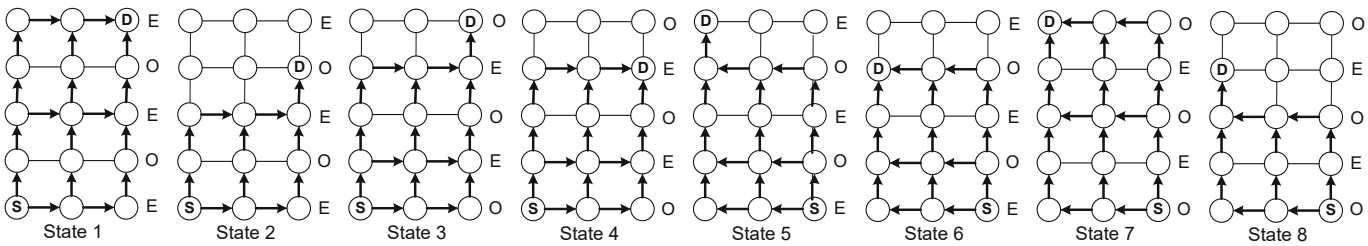

Fig. 6 Eight states of source nodes and destination nodes in H_H subnetworks (S: source node; D: destination node; E: even row; O: odd row)

Table 1 Minimal-direction availability under identical geometric conditions at packet injection (fault-free)

Direction class	Geometric condition	$M(\Delta x, \Delta y)$	FTHOE $\mathrm{DoA}_{\min}$	OFTR $\mathrm{DoA}_{\min}$
EN	Odd row/Odd column	{E, N}	2	1
	Odd row/Even column	{E, N}	2	2
	Even row/Odd column	{E, N}	2	1
	Even row/Even column	{E, N}	2	2
ES	Odd row/Odd column	{E, S}	2	2
	Odd row/Even column	{E, S}	2	1
	Even row/Odd column	{E, S}	2	2
	Even row/Even column	{E, S}	2	1
WS	Odd row/Odd column	{W, S}	2	1
	Odd row/Even column	{W, S}	2	2
	Even row/Odd column	{W, S}	2	1
	Even row/Even column	{W, S}	2	2
WN	Odd row/Odd column	{W, N}	2	2
	Odd row/Even column	{W, N}	2	1
	Even row/Odd column	{W, N}	2	2
	Even row/Even column	{W, N}	2	1
E/W/N/S	Same row ($\Delta y = 0$) or same column ($\Delta x = 0$)	{E}/{W}/{N}/{S}	1	1

$M(\Delta x, \Delta y)$: ideal minimal-direction set determined by Manhattan distance. $\mathrm{DoA}_{\min}$: number of minimal directions that can be selected at injection. The geometric conditions before and after "/" are for algorithms FTHOE and OFTR, respectively. Under fault-free, VC-less conditions, HOE provides $\mathrm{DoA}_{\min} = 2$ for all non-axis-aligned cases listed in this table, serving as the baseline

level, FTHOE strictly inherits the HOE turn constraints to maintain deadlock-free operation, while introducing a port-priority reordering mechanism driven by local fault vectors to dynamically adjust the selection order among the candidate output ports permitted by HOE. This mechanism provides a rule-level foundation for retaining more available minimal directions and avoiding unnecessary detours under fault conditions. Its concrete effects are further illustrated through a constructive routing example in Section 5.1.

4 FTHOE routing algorithm design

Building upon the HOE routing algorithm, we propose a fault-tolerant routing algorithm, termed FTHOE, which targets node and link fault scenarios. The proposed algorithm uniformly classifies packet transmission directions and constructs a formalized port-priority decision strategy by incorporating a small amount of locally available routing state information. In this way, FTHOE enables effective detouring around faulty regions without relying on global fault information.

FTHOE classifies packet transmission directions into eight direction classes based on the relative position of the destination node D with respect to the current node C: EN, WN, ES, WS, E, W, S, and N. For diagonal directions (EN/WN/ES/WS), there are two orthogonal candidate output ports under the minimal-path constraint, whereas for axial directions (E/W/S/N), only a single minimal-path output port is available. This direction classification provides a unified decision entry for subsequent port selection, avoiding the need to define routing rules separately for each geometric relation.

To formally describe the generation of port priorities, FTHOE unifies the implicit special conditions in the original algorithm into a set of local routing states, including: (1) parity of the row index of the current node, denoted as $\mathrm{odd}(y_c)$; (2) whether the source node S and the current node C are located in the same row (SameRow), i.e., $y_c = y_s$; (3) whether the current node is located in the row adjacent to the destination node, i.e., $y_c = y_d - 1$. All these states can be directly obtained from local coordinate information and do not rely on additional communication or global awareness. Table 2 summarizes the port-priority strategies of FTHOE under different direction classes and routing states.

For each direction class, the policy table first specifies a priority ordering of the minimal-path candidate output ports (P1 → P2). During routing decision-making, the router examines the candidate ports sequentially according to this priority order. When a higher-priority port is unavailable due to a fault or turn constraints, the routing decision automatically falls back to the next port in the priority sequence. If all ports in the minimal-path candidate set are unavailable at the current node, the router performs detouring forwarding using the predefined suboptimal ports specified in the policy table, thereby bypassing the faulty region and restoring progress toward the destination.

For each direction class, the algorithm first defines a geometric minimal-path candidate port set, which is determined solely by the relative geometric relationship between the source and destination nodes (i.e., the direction class) and is statically specified in the algorithm as an upper bound on minimal routing directions. For diagonal direction classes (EN, WN, ES, and WS), this geometric minimal-path set consists of a pair of orthogonal directions, whereas for axis-aligned direction classes (E, W, N, and S), the minimal direction is unique. Based on this geometric definition, state variables, such as row parity, SameRow, and the condition $y_c = y_d - 1$, are used only to constrain or reorder the admissible minimal-port subset

Table 2 Port-priority rules of FTHOE under different direction classes

Direction class	State condition	P1	P2	Suboptimal path
EN	Odd ∧ SameRow	E/N	N/E	W (S if necessary)
	Odd ∧ no SameRow	N	E	When E is unavailable, S
	Even ∧ $y_c \neq y_d - 1$	E/N	N/E	S
	Even ∧ $y_c = y_d - 1$	E	N/A	S
WN	Odd ∧ $y_c \neq y_d - 1$	W/N	N/W	S
	Odd ∧ $y_c = y_d - 1$	W	N/A	S
	Even ∧ SameRow	W/N	N/W	S
	Even ∧ no SameRow	N	W	S
ES	Odd	E/S	S/E	W
	Even ∧ SameRow	E/S	S/E	W
	Even ∧ no SameRow	S	E	W
WS	Odd ∧ SameRow	W/S	S/W	E
	Odd ∧ no SameRow	S	W	E
	Even	W/S	S/W	N
E	–	E	N/A	S (again N)
W	–	W	N/A	S (again N)
S	Odd	S	N/A	E (again W)
	Even	S	N/A	W (again E)
N	Odd	N	N/A	E (again W)
	Even	N	N/A	W (again E)

– indicates that no specific state condition is required; N/A: not available

at the current node within the geometric minimal-direction set, without introducing new minimal directions or expanding the minimal-path direction set. Consequently, state-dependent port-priority adjustments do not alter the set of direction turns that are globally permitted by the routing algorithm.

To further illustrate how the policy table operates under fault conditions, Algorithm 1 presents the routing decision process for packets in the EN direction class under FTHOE. When the destination node D is located in the northeast quadrant relative to the current node C, the packet is classified as belonging to the EN direction class, and its minimal-path candidate ports are east and north. Based on the current row parity, whether the packet originates from the same row as the source node, and the condition $y_c = y_d - 1$, the router determines the priority sequence of these candidate ports from the policy table and selects the first port that is both fault-free and compliant with the turn constraints. If all minimal-path candidate ports are unavailable, the packet is forwarded via a predefined suboptimal port specified in the policy table to enter an adjacent row or column, thereby bypassing the faulty region. In subsequent routing steps, once the packet leaves the fault neighborhood, the routing decision automatically reverts to the minimal-path candidate set corresponding to the EN direction class.

The above detouring behavior is realized through a purely combinational next-hop restriction mechanism. At each hop, the routing decision depends only on the current node's geometric relationship to the destination, the turn model, the local fault vector, and the input port direction, and accordingly generates a finite set of candidate output ports, without introducing any per-packet state or metadata that must be preserved across hops. The so-called "temporary allowance of otherwise prohibited turns" manifests only at the current hop and only when all minimal-path candidate ports are unavailable; in this case, a restricted set of suboptimal output ports is enabled to facilitate escape from the fault neighborhood. In the subsequent routing decision, the packet is again strictly constrained by the FTHOE turn model and the corresponding direction class, and prohibited turns are not persistently applied.

Algorithm 1 EN-direction routing decision in FTHOE

Input: current router C, destination node D, source node S, and input port inport

```
state ← {odd(y_c), SameRow(y_s = y_c), y_c = y_d − 1}
cand ← PolicyTable[EN][state]
for all dir ∈ cand do
    if Healthy(dir) and ¬UTurn(dir, inport) then
        return dir
    end if
end for
return AnyHealthyNonUTurnPort()
```

This mechanism remains consistent with the turn constraints of the original HOE routing and does not introduce any new illegal turns. The routing decision processes for the other direction classes are fully isomorphic to that of the EN direction class, differing only in the corresponding entries of the policy table. As long as at least one port in the minimal-path candidate set is available, the port-priority policy always selects an output from this set, thereby preserving minimal routing.

In contrast, the design of FTHOE does not primarily aim

to further reduce hardware overhead; rather, it seeks to preserve the minimal-path diversity inherent to the HOE turn model as much as possible without introducing VCs. By incorporating a port-priority reordering mechanism based on local fault vectors, FTHOE avoids overly restrictive path selection under complex fault scenarios, thereby exhibiting stronger robustness in terms of path diversity and load balancing. This improvement does not stem from increased hardware complexity, but rather from a different routing decision strategy.

5 Theoretical analysis

5.1 Algorithm examples

This subsection presents a constructive routing example to illustrate how the differences in minimal-path DoA summarized in Table 1 translate into concrete path-selection behaviors under fault conditions, and to further explain the latency and throughput trends observed in the subsequent performance evaluation. To compare different design philosophies, two VC-less fault-tolerant routing algorithms, LBFT (Xie et al., 2016) and OFTR (Guan et al., 2023), are selected as representatives of non-HOE-based schemes and are evaluated against the proposed FTHOE.

Under fault-free conditions, LBFT employs single-path forwarding and therefore has a minimal-path candidate set of size one. In contrast, both OFTR and FTHOE support multipath routing, and their state-level minimal-path DoA values are summarized in Table 2. For the communication pair shown in Fig. 7a, FTHOE preserves a more complete set of minimal directions under the corresponding communication state, thereby providing a larger number of equal-length minimal-path alternatives. This property lays the foundation for dispersing traffic and alleviating hotspots under high-load conditions. Unlike the DoA table in Section 3.3, which quantifies state-level minimal-path multiplicity using representative distance settings, Fig. 7 presents a concrete example that visually compares the routing behaviors of different algorithms under a specific fault scenario.

When faults occur in the network, some minimal directions may be directly blocked. In this case, the key distinction among routing algorithms lies not in reachability—which can usually be achieved through detouring—but in whether a usable set of minimal directions (i.e., the remaining minimal directions) can still be preserved. To avoid potential turn cycles and reduce implementation complexity, OFTR adopts relatively conservative turn rules and port-selection policies; as a result, it may forfeit still-available minimal branches under certain communication states and degenerate into near single-path routing. In contrast, while strictly preserving the HOE turn constraints, FTHOE leverages local fault vectors to reorder the priorities of candidate output ports, enabling the activation and selection of bypass branches that still satisfy the minimal-distance requirement under the same fault conditions. This capability effectively prevents premature transitions to non-minimal detours.

Taking the communication scenario illustrated in Fig. 7b as an example, under the given local fault distribution, OFTR is restricted to a limited set of minimal paths, and some feasible minimal bypass directions cannot be discovered. In contrast, by employing a port-priority reordering mechanism driven by local fault vectors, FTHOE activates additional minimal-path arrival directions while strictly adhering to the turn constraints, thereby successfully selecting minimal bypass paths that OFTR fails to identify. Consequently, this behavior alleviates traffic concentration within the fault neighborhood and is reflected in the subsequent performance evaluation as lower average network latency and higher saturation throughput.

Overall, existing VC-less fault-tolerant routing schemes often trade routing flexibility for lower implementation

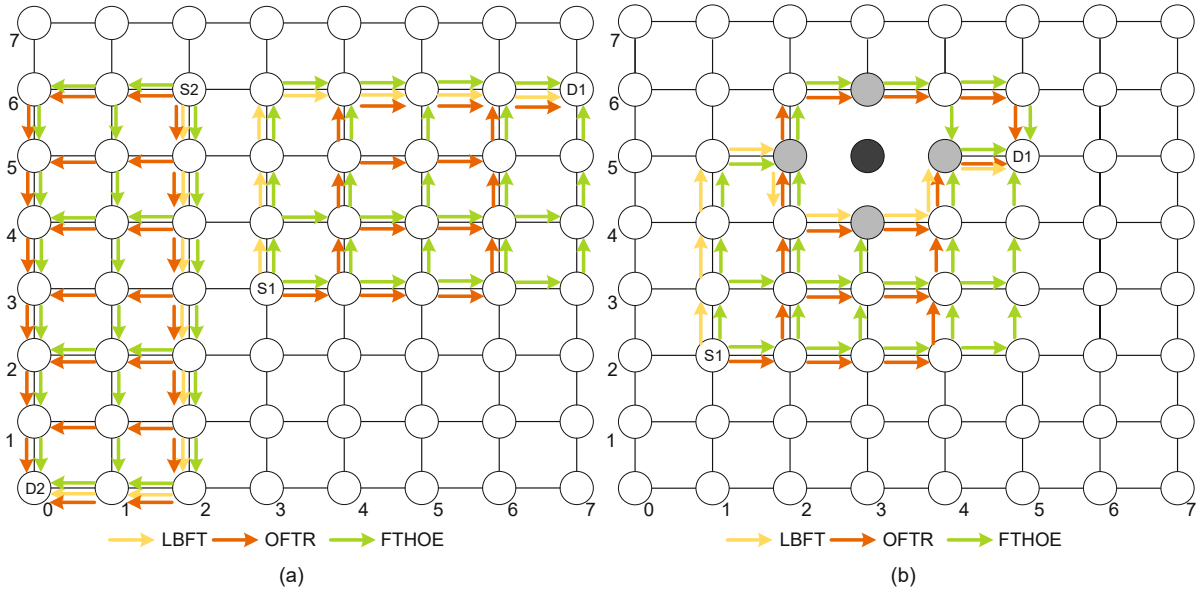

Fig. 7 Comparison of LBFT, OFTR, and FTHOE routing paths in a network, where (a) represents the distribution of fault-free routing transmission paths and (b) represents routing bypass paths in a faulty network (S1/S2: source nodes; D1/D2: destination nodes)

complexity by tightening turn restrictions or adopting conservative port-selection strategies; however, this typically reduces the set of minimal-path candidates and makes traffic more prone to concentration. In contrast, FTHOE aims to preserve, as much as possible, the minimal-path diversity inherent to the HOE turn model without introducing VCs. By incorporating a port-priority reordering mechanism driven by local fault vectors, FTHOE retains more available minimal directions within fault neighborhoods, thereby exhibiting stronger robustness in terms of path diversity and load balancing.

5.2 Deadlock avoidance

A routing algorithm is deadlock-free as long as no cyclic closed loop exists in the channel dependency graph (CDG) (Dally and Seitz, 1987). Therefore, proving that packet routing under the FTHOE algorithm cannot form any closed dependency cycles is sufficient to establish its deadlock freedom. As described in Section 4, the FTHOE routing algorithm is uniformly formulated as a direction-class- and state-driven port-priority strategy. The deadlock-freedom proof in this study relies solely on the set of allowed turns and is independent of the specific port-priority implementation; state variables such as SameRow and $y_c = y_d - 1$ are used only for port scheduling and do not introduce new turn types, and thus do not alter the channel dependency relationships.

Lemma 1 When packet routing follows the FTHOE turn model, no dependency cycles can be formed.

Proof When no faults occur in the network, packet routing follows the FTHOE turn rules. In even-row nodes, packets are not allowed to take ES and NW turns, and in odd-row nodes, packets are not allowed to take NE and WS turns. This ensures that NE and ES turns in a clockwise loop and NW and WS turns in a counterclockwise loop cannot occur in the same row of nodes, thereby preventing the formation of cyclic channel dependencies (i.e., no dependency cycles can be established in the CDG).

Lemma 2 When a turn prohibited by the FTHOE turn model occurs, it cannot form a dependency cycle.

Proof If each channel dependency loop formed during packet routing lacks at least one boundary edge, the network is deadlock-free. In the FTHOE turn model, the prohibited transitions are ES, NW, NE, and WS turns. The following describes each direction separately. After an even-row packet performs an ES transition, its south-adjacent node can perform only an SE transition, so the rightmost boundary of the channel loop cannot be formed (Fig. 8a). When an even-row packet performs an NW turn, its west-adjacent node can perform only a WN turn, so the upper boundary of the channel loop cannot be formed (Fig. 8b). When an odd-row packet performs an NE turn, its east-adjacent node can perform only an EN turn, so the upper boundary of the channel loop cannot be formed (Fig. 8c). When an odd-row packet performs a WS turn, its south-adjacent node can perform only an SW turn, so the leftmost boundary of the channel loop cannot be formed (Fig. 8d). Therefore, when a prohibited turn occurs in the FTHOE model, as long as the next node imposes the corresponding turn constraint, a channel dependency loop will not be formed.

Fault detouring in FTHOE is constrained by a next-hop restriction mechanism. Specifically, when a packet temporarily triggers a normally prohibited turn at the current node due to a fault to escape the fault neighborhood, its subsequent routing decisions are strictly recomputed according to the direction classes and the turn model, and repeated use of the prohibited turn at consecutive nodes is not permitted. This mechanism ensures that the temporary turn serves only as a local, one-time detour and is not persistently propagated along the subsequent path.

From the perspective of CDG, the overall structure of channel dependencies in FTHOE is statically constrained by its turn model. Although under fault conditions, a packet may temporarily employ a turn that is prohibited in the fault-free case at certain hops, this turn is effective only at the current node and is strictly limited by the next-hop restriction mechanism; therefore, it does not create propagatable channel-dependency edges in the CDG, nor does it extend the original structure of the channel dependency graph. Since the topology of the CDG is statically determined by the set of allowed turn constraints and does not vary with packet concurrency or local detouring behavior, even when multiple packets simultaneously trigger detour forwarding, no new cyclic dependencies are introduced into the CDG.

Theorem 1 The FTHOE routing algorithm is deadlock-free.

Proof In fault-free scenarios, as established by Lemma 1, packets cannot form dependency cycles in the CDG when following the FTHOE turn model. In the presence of faults, although some packets may trigger prohibited turns due to detouring, Lemma 2 demonstrates that such turns likewise do not lead to the closure of channel dependency cycles. Therefore, under all conditions, no closed directed cycles exist in the channel dependency graph of the FTHOE routing algorithm, and the algorithm is deadlock-free.

Since the aforementioned temporary turns are independently and consistently constrained at each node, packets cannot form inter-node cyclic channel dependencies through such

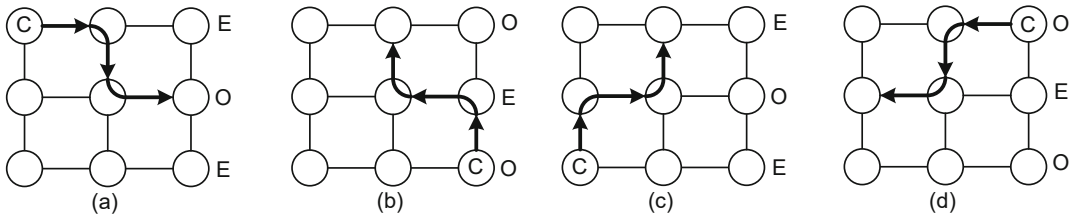

Fig. 8 Prohibited turns in the FTHOE transition model, where (a), (b), (c), and (d) represent ES, NW, NE, and WS turns, respectively (C: current node; E: even row; O: odd row)

turns. Therefore, FTHOE remains deadlock-free even under concurrent packet processing scenarios.

5.3 Livelock avoidance analysis

The previous subsection has established that the proposed FTHOE routing algorithm is deadlock-free under the imposed turn constraints. However, deadlock freedom alone does not necessarily preclude the possibility of livelock, especially in partially adaptive fault-tolerant routing schemes that rely solely on local information. Therefore, in this subsection, we analyze the livelock behavior of FTHOE under clearly stated assumptions.

In FTHOE, for any given direction class, a packet's routing choice at each hop is always restricted to a finite, ordered set of candidate output ports. This set is jointly determined by the direction class, the turn model, and the local routing state, and selection is performed through a port-priority policy. When shortest-path candidate ports are available, the routing decision guarantees that the packet makes monotonic progress toward the destination along at least one dimension. When shortest-path ports are temporarily unavailable due to faults or turn constraints, the algorithm triggers a constrained suboptimal detouring strategy, allowing the packet to temporarily deviate from the shortest path at the current hop to bypass the fault region.

Note that the above detouring behavior is strictly constrained in terms of spatial extent, persistence, and propagability. First, the fault model considered in this work assumes that the sets of faulty nodes and links are static or quasi-static in the temporal domain and have a limited spatial distribution. Under this premise, any fault-induced detour path is confined to the geometric vicinity of the fault region, with the detour depth and path stretch bounded by the scale of the fault area. Once a packet exits the fault neighborhood, the routing decision automatically reverts to the shortest-path candidate set corresponding to the given direction class. Second, FTHOE explicitly restricts the scope of prohibited-turn usage through the next-hop restriction mechanism: a temporary detour is effective only at the current hop, and repeated use of prohibited turns at subsequent nodes is not permitted, thereby preventing packets from forming persistent deflection loops within local regions.

Furthermore, even under concurrent scenarios in which multiple packets simultaneously trigger detouring, the routing decisions of FTHOE do not induce livelock behavior. This is because packet detouring decisions do not alter the structure of the channel dependency graph, nor do they introduce propagatable cyclic dependencies; concurrent detouring merely manifests as multiple packets temporarily selecting constrained suboptimal output ports at local nodes, without causing cumulative violations of the path-selection rules in either the temporal or spatial domain. Consequently, packets cannot experience infinite deflections, and both path stretch and the number of detours remain bounded.

In summary, under conditions of finite network scale, finite and static fault distributions, and finite injection rates, the FTHOE routing algorithm can effectively prevent packets from undergoing unbounded detouring behavior and thus exhibits livelock avoidance in an engineering sense. The simulation study likewise does not observe persistent detouring or pronounced tail-latency divergence, which is consistent with the above analysis.

6 Evaluations

To systematically evaluate the performance of the proposed FTHOE fault-tolerant routing algorithm under different network scales, traffic patterns, and fault conditions, cycle-accurate simulations are conducted using the Garnet interconnection network model (Agarwal et al., 2009) within the gem5 simulator (Lowe-Power et al., 2020). Unless otherwise specified, FTHOE and the comparison algorithms use identical network and simulation parameter settings to ensure fairness and reproducibility of the simulation results. The comparison algorithms include LBFT and OFTR, both of which are VC-less fault-tolerant routing schemes.

Specifically, packet and flit sizes, buffer depths, router and link latencies, VNet configuration, and VC allocation in Garnet are kept identical across all algorithms, as summarized in Table 3. Although multiple VNets and VCs are instantiated internally by Garnet for protocol separation and deadlock avoidance, routing decisions in all evaluated algorithms are VC-agnostic and do not rely on VC differentiation.

Table 3 Simulation configuration

Parameter	Setting
Simulator	gem5 with Ruby Garnet
Network topology	2D mesh, 8×8, 16×16, 32×32
Routing algorithms	FTHOE, OFTR, LBFT
VCs (routing usage)	None; routing decisions don't rely on VCs
VNets	3 (protocol-defined VNets)
No. of VCs per VNet	4 (implementation-level VCs in Garnet)
Data VC buffer depth	4 flits per data VC
Control VC buffer depth	1 flit per control VC
Flit size	16 bytes
Router latency	1 cycle
Link latency	1 cycle
Injection rate	0–0.20 flits/(node·cycle), step 0.01
Warm-up period	10 000 cycles
Measurement period	50 000 cycles
Traffic patterns	Uniform, transpose, bit-reverse, hotspot

VNets: virtual networks

All simulations use Garnet's default credit-based flow control with identical credit timing across all algorithms. No adaptive throttling, injection-rate control, or algorithm-specific flow-regulation mechanisms are enabled in any simulation. When delayed or stale fault awareness is evaluated, it is modeled independently of traffic injection and flow control.

6.1 Performance analysis

We systematically evaluate the performance of the proposed FTHOE routing algorithm under different traffic patterns and fault scenarios, and compare it with representative routing schemes, including LBFT and OFTR. The evaluation primarily focuses on average network latency, complemented by a quantitative analysis of network carrying

capability in terms of saturation injection rate and steady-state delivered throughput.

The average network latency is defined as the average number of clock cycles experienced by a packet from the time it is injected at the source node until its last flit is successfully received at the destination node. Network throughput is quantified using end-to-end delivered throughput, defined as the normalized ratio of the total number of flits successfully received by all nodes within a steady-state observation window to the product of the number of nodes and the number of cycles, expressed in unit of flit/(node·cycle). As the injection rate increases, when the network behavior transitions from approximately linear latency growth to a pronounced nonlinear latency increase, the corresponding injection rate is defined as the saturation injection rate. This saturation point reflects the maximum load level that the network can sustain while maintaining stable communication conditions. Unless otherwise stated, all throughput-related metrics reported in this section are obtained from 20 independent simulation runs, and mean values with 95% confidence intervals are computed following the same statistical methodology used for load-related metrics.

6.1.1 Fault-free and single-fault scenarios

First, the performance of each routing algorithm is evaluated under fault-free conditions. Fig. 9 illustrates the variation of the average network latency with the injection rate in an 8×8 mesh topology. The results show that, under the uniform, transpose, bit-reverse, and hotspot traffic patterns, FTHOE achieves latency levels comparable to those of the baseline schemes in the low- to moderate-injection-rate region, indicating that it does not introduce additional routing overhead in fault-free scenarios. As the injection rate further increases, the network gradually enters a congested state, and the average latency of all algorithms rises sharply. Compared with LBFT and OFTR, FTHOE exhibits a higher injection rate at the latency inflection point, indicating its ability to maintain stable communication under heavier traffic loads. This behavior is attributed mainly to the fact that, while preserving minimal-path constraints, FTHOE improves the utilization efficiency of available paths through a port-priority adjustment mechanism driven by local fault awareness, thereby delaying the onset of network saturation.

Subsequently, the fault-tolerant performance of each routing algorithm is evaluated under single-node and single-link fault conditions. Fig. 10 shows the variation of average network latency with the injection rate under the uniform and hotspot traffic patterns. Compared with the fault-free scenario, the network enters saturation at lower injection rates, and the average latency increases more rapidly as the injection rate grows. This behavior arises mainly because faulty nodes or links disrupt the original minimal-path structure, forcing a portion of the traffic to detour around fault neighborhoods and increasing contention on remaining resources. Compared with LBFT and OFTR, FTHOE exhibits a more gradual latency growth trend in these scenarios and consistently delays the onset of saturation, indicating superior path-selection flexibility and fault-tolerance capability under isolated single-fault conditions.

Table 4 summarizes the saturated delivered throughput under fault-free and single-fault scenarios. FTHOE consistently achieves the highest throughput across all evaluated traffic patterns, while exhibiting performance comparable to the comparison routing schemes in the fault-free case. Under

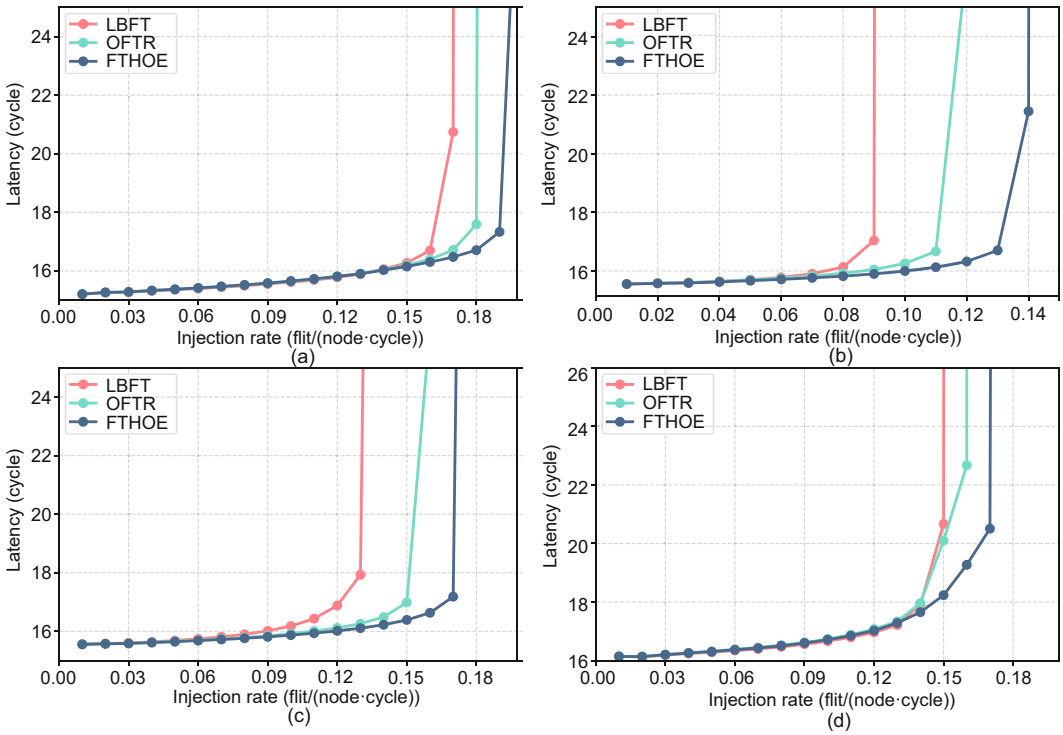

Fig. 9 Average network latency under fault-free conditions for four traffic patterns: (a) uniform; (b) transpose; (c) bit-reverse; (d) hotspot

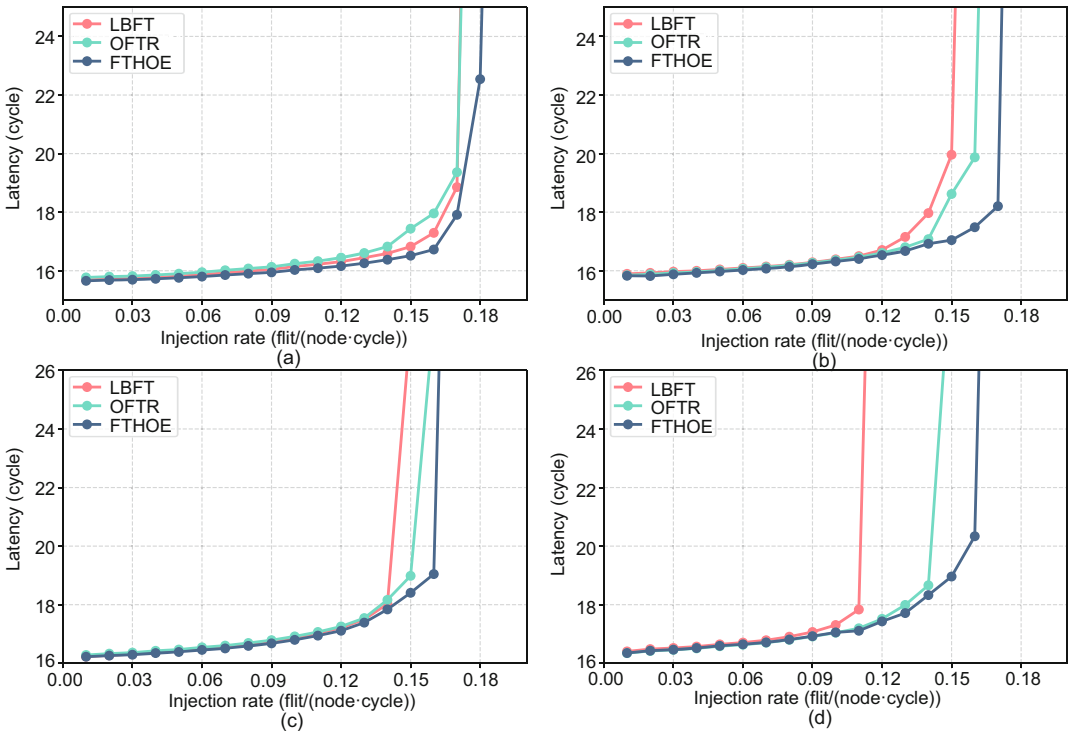

Fig. 10 Average network latency under single-node and single-link fault conditions for two traffic patterns: (a) uniform, 1 link fault; (b) uniform, 1 node fault; (c) hotspot, 1 link fault; (d) hotspot, 1 node fault

single-node and single-link faults, FTHOE maintains a clear throughput advantage, which becomes more pronounced under hotspot traffic, indicating improved congestion mitigation capability in the presence of isolated faults. All throughput and saturation metrics are obtained from 20 independent runs and evaluated with 95% confidence intervals. Near saturation, network throughput is dominated by structural bottlenecks rather than short-term traffic randomness; consequently, the associated confidence intervals are negligible and are omitted from the table for clarity.

Table 4 Saturated delivered throughput under various scenarios

Algorithm	Throughput (flit/(node·cycle))					
	Uniform			Hotspot		
	Fault-free	1 link	1 node	Fault-free	1 link	1 node
LBFT	0.1703	0.1677	0.1462	0.1507	0.1377	0.1099
OFTR	0.1851	0.1698	0.1547	0.1587	0.1475	0.1399
FTHOE	0.1970	0.1797	0.1644	0.1821	0.1562	0.1507

All values are averaged over 20 independent runs. 1 link: single-link fault; 1 node: single-node fault

6.1.2 Mixed-fault scenarios

To more closely reflect the practical defect characteristics of wafer-scale systems, in which compute node faults and interconnect degradation coexist, this work extends the fault model from single-fault to mixed-fault scenarios involving both node and link faults. In this setting, K nodes and L links are randomly selected to fail, with the constraint that the fault sets do not overlap, thereby avoiding duplicate counting.

Figs. 11–13 present the network performance comparison results under mixed-fault conditions with gradually increasing fault density ($K = L = 1, 2, 4$) for the uniform and hotspot traffic patterns. It can be observed that the throughput of all algorithms decreases compared with the single-fault scenario, and the network tends to enter saturation more readily. However, FTHOE is still able to maintain a relatively high throughput in the medium- to high-injection-rate region and exhibits a more gradual latency growth trend. These results indicate that the port-priority dynamic adjustment mechanism based on local fault vectors effectively reduces the packet stalling probability in the vicinity of faulty regions, thereby mitigating the performance degradation caused by mixed faults. Consequently, FTHOE demonstrates strong robustness under combined node and link fault conditions, making it well-suited for reliable operation in large-scale systems where multiple faults may occur simultaneously.

Table 5 reports the saturated delivered throughput under mixed-fault (link+node) scenarios with increasing fault density. As the fault density increases from 1+1 to 4+4, the throughput of all routing schemes degrades due to reduced path diversity and aggravated congestion. Nevertheless, FTHOE consistently delivers the highest throughput across all traffic patterns and fault densities, demonstrating superior robustness under compound fault conditions. Under high-density mixed-fault conditions (e.g., $K = L = 4$), the peak delivered throughput does not necessarily coincide with the saturation injection rate identified from latency curves. This behavior is expected, as severe compound faults induce persistent congestion around faulty regions, causing throughput to saturate earlier than latency-based indicators. In such scenarios, adaptive routing schemes such as OFTR may also exhibit slightly

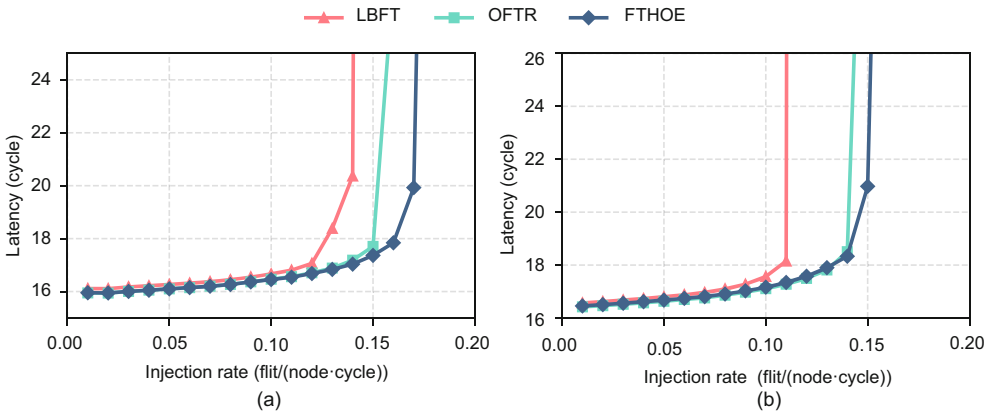

Fig. 11 Average network latency under mixed-fault conditions with $K = L = 1$: (a) uniform; (b) hotspot

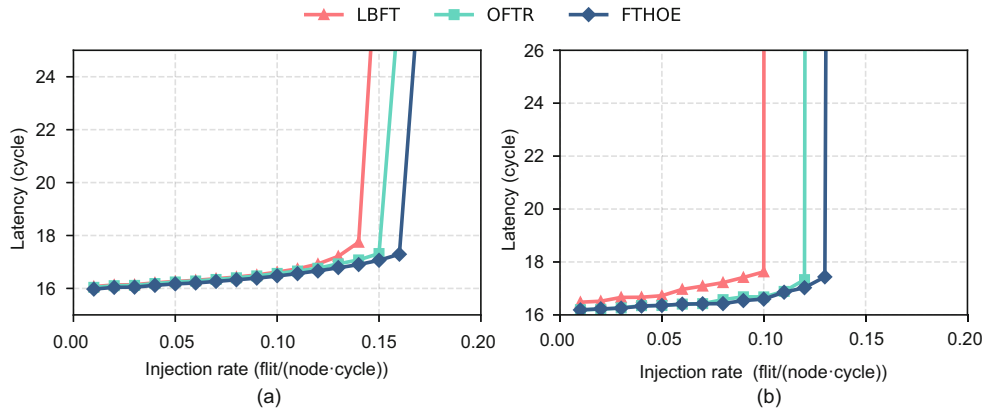

Fig. 12 Average network latency under mixed-fault conditions with $K = L = 2$: (a) uniform; (b) hotspot

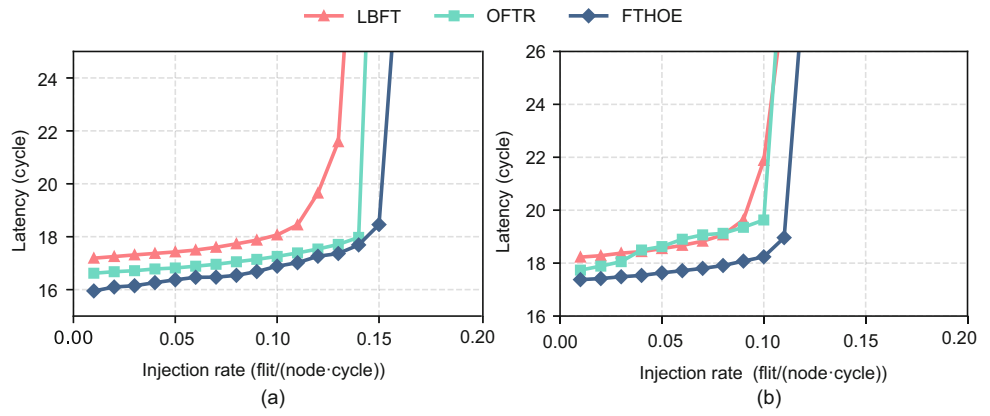

Fig. 13 Average network latency under mixed-fault conditions with $K = L = 4$: (a) uniform; (b) hotspot

Table 5 Saturated delivered throughput under mixed faults (K links+L nodes)

Algorithm	Throughput (flit/(node·cycle))					
	Uniform			Hotspot		
	$K = L = 1$	$K = L = 2$	$K = L = 4$	$K = L = 1$	$K = L = 2$	$K = L = 4$
LBFT	0.1455	0.1354	0.1029	0.1081	0.1030	0.0843
OFTR	0.1541	0.1450	0.1122	0.1366	0.1124	0.0844
FTHOE	0.1627	0.1538	0.1216	0.1453	0.1220	0.0931

All values are averaged over 20 independent runs

increased run-to-run variance, since the availability of detour paths is significantly constrained and routing decisions become more sensitive to local congestion fluctuations. Importantly, neither effect alters the relative performance trends among routing schemes, which remain consistent across all evaluated scenarios, and the overall throughput advantage of FTHOE is preserved.

6.2 Load balancing analysis

This subsection evaluates the load-balancing capability of different routing algorithms under fault conditions from the perspective of node-level traffic distribution. The node load L_i is defined as the total number of packets processed by router i within a given observation window, including packets forwarded, injected, and received at that node. Across all routing algorithms, the injection and reception operations are kept identical; therefore, they do not affect the relative differences in node-load distribution among the different routing strategies. Simulations are conducted on an 8×8 mesh network topology using the gem5 simulator under the hotspot traffic pattern with an injection rate of 0.02 flits/(node·cycle), comparing LBFT, OFTR, and FTHOE. All simulations are repeated 20 times, and the mean values along with 95% confidence intervals are reported.

Fig. 14 presents the node load heatmaps corresponding to different routing algorithms under the aforementioned simulation configuration. It can be observed that, in the presence of faults, all algorithms exhibit a certain degree of traffic aggregation, particularly in the vicinity of faulty regions and along the associated detour paths. Among them, LBFT shows pronounced traffic concentration under combined fault and hotspot conditions, whereas multipath routing algorithms alleviate this issue to some extent. However, relying solely on visual inspection makes it difficult to draw objective conclusions regarding load-balancing performance.

To this end, Table 6 reports quantitative metrics, including load standard deviation, coefficient of variation (CV), Gini coefficient, and normalized entropy. The results show that LBFT exhibits the highest load standard deviation, CV, and Gini coefficient, as well as the lowest normalized entropy under the hotspot scenario, indicating that single-path routing is ineffective at dispersing hotspot traffic. In contrast, both FTHOE and OFTR significantly outperform LBFT across all metrics, demonstrating that increased path diversity can effectively alleviate node-level load concentration.

To further clarify the intrinsic relationship between routing mechanisms and load-balancing metrics, this study introduces a simplified analytical framework based on explicit assumptions, which is used to explain how port-priority reordering influences path-selection probabilities and its effect on the expected node-load distribution.

Under steady-state communication conditions, it is assumed that a set of source–destination communication pairs in the network generate packets at a fixed injection rate, and that each packet is forwarded along the set of feasible shortest paths. At any given router, for a given packet, the set of selectable output directions is denoted as $\mathcal{P}(S, D)$. Let $p_i(f)$ denote the probability that a communication flow f passes through router i; then, the expected load of router i can be expressed as

$$\mathbb{E}[L_i] \propto \sum_f \lambda_f p_i(f), \tag{3}$$

where λ_f represents the injection rate of flow f. Note that $p_i(f)$ is not generated by explicit random selection; instead, it is implicitly induced by the output-port priority rules of the routing algorithm over long-term operation.

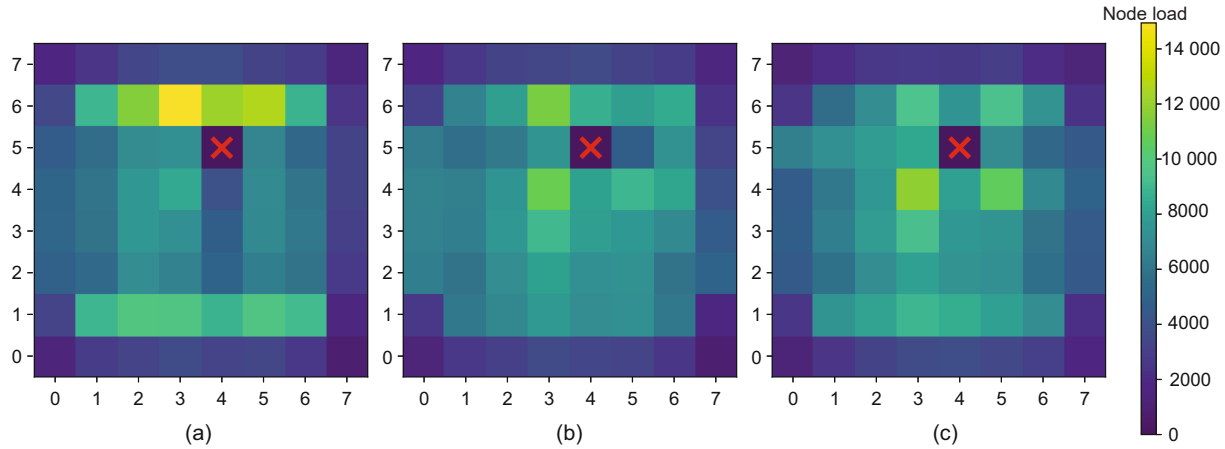

Fig. 14 Comparison of heatmaps for three algorithms under a hotspot traffic pattern: (a) LBFT; (b) OFTR; (c) FTHOE. The horizontal and vertical coordinates indicate the position of the node in the network

Table 6 Quantitative load-balancing metrics under hotspot traffic

Algorithm	Load standard deviation	CV	Gini coefficient	Normalized entropy
LBFT	2994.8 ± 0	0.5256 ± 0	0.2890 ± 0	0.9679 ± 0
OFTR	2452.1 ± 3.5	0.4330 ± 0.0006	0.2430 ± 0.0004	0.9756 ± 0.0001
FTHOE	2588.8 ± 3.2	0.4569 ± 0.0005	0.2583 ± 0.0003	0.9731 ± 0.0001

In OFTR, the output port-priority follows a fixed order, causing certain directions to be persistently preferred during long-term operation; as a result, routers along the corresponding paths exhibit higher traversal probabilities and are prone to load concentration. In contrast, FTHOE dynamically reorders output-port priorities based on the local fault vector and, while preserving shortest-path constraints and deadlock-free turn rules, effectively redistributes path-selection probabilities among multiple feasible directions. Consequently, in hotspot regions or in the vicinity of fault neighborhoods, the probability mass of traffic that would otherwise be concentrated on a small number of paths is dispersed across a larger set of routers, thereby reducing the spatial variance of $\{\mathbb{E}[L_i]\}$ in a statistical sense. This probability-level dispersion effect naturally explains the reductions in load standard deviation, CV, and Gini coefficient, as well as the increase in normalized entropy observed in Table 6.

Note that the design objective of FTHOE is not to simply minimize global load variance, but rather to achieve controlled traffic dispersion under combined hotspot and fault conditions while preserving routing constraints and behavioral stability. In some global statistical metrics, OFTR may exhibit slightly better load-balancing performance; however, across repeated simulations, FTHOE demonstrates narrower confidence intervals, indicating more stable and predictable load-distribution behavior. This trade-off between load-balancing capability and routing constraints makes FTHOE more suitable for network management and congestion mitigation in fault-prone scenarios.

6.3 Power and area analysis

This subsection compares the routing algorithms OFTR, LBFT, and the proposed FTHOE from the perspectives of hardware implementation overhead and runtime energy consumption. Note that a standard HOE router is adopted as the baseline implementation for evaluating the hardware cost of different fault-tolerant routing algorithms. HOE itself is not a fault-tolerant routing algorithm and does not incorporate fault-awareness logic; both OFTR and FTHOE introduce local fault-awareness mechanisms on top of the HOE router, whereas LBFT employs a more complex fault-information representation and control strategy. All simulations are conducted on an 8×8 mesh topology under the uniform traffic pattern and a dual-node fault scenario. Network behavior is analyzed through gem5 simulations, while router-level power and area are modeled and evaluated using the Design Space Exploration of Networks Tool (DSENT) at the 45-nm technology node, thereby ensuring comparability of the analysis results across different algorithms.

Fig. 15 presents the power breakdown of the three routing algorithms under different injection rates, where the total power consists of leakage power and dynamic power. It can be observed that, as the injection rate increases, the total power consumption of all three algorithms exhibits a gradual upward trend, which is driven mainly by the increase in dynamic power. LBFT consistently shows the highest power consumption across all injection rates, which is directly related to its introduction of larger fault information tables and additional control-state storage. In comparison, the total power consumption of FTHOE is slightly higher than that of OFTR, while the two remain within the same order of magnitude. This is mainly because FTHOE does not optimize energy efficiency by reducing the instantaneous hardware power consumption of individual routers, but rather by improving network-level operating behavior to sustain more effective communication activity at the same injection rate, which in turn leads to a slightly higher dynamic power consumption.

To eliminate the impact of injection-rate differences, our study further analyzes the relationship between energy per flit and average network latency. Fig. 16 illustrates the relationship between energy per flit and average flit latency at different injection rates. For each routing algorithm, the data points correspond to increasing injection rates from left to right. The results indicate that, as the injection rate increases, the energy per flit of all three algorithms generally decreases, reflecting improved utilization of network resources. Across the evaluated injection-rate range, the FTHOE curve consistently lies below or close to that of OFTR and noticeably below that of LBFT over comparable latency regions, indicating lower energy consumption at similar performance levels.

In terms of area analysis, it should be noted that all three routing algorithms adopt a VC-less implementation and do not modify the core data path structure of the router, including modules such as input buffers, switch allocators, and crossbars. Consequently, area differences among the algorithms mainly stem from the additional control state and associated storage structures introduced by each algorithm. Under unified router microarchitectural parameters (including the number of ports, buffer depth, and flit width), this work differentially models and evaluates the additional control state required by each routing algorithm on top of the HOE baseline router. To further illustrate the sources of area differences among the algorithms, Table 7 compares the additional control state and implementation complexity introduced by each routing algorithm relative to the HOE baseline. Table 8 presents the area estimation results for the three routing algorithms.

As can be observed from the results in these two tables, the three algorithms exhibit identical area footprints in components related to the core data path. The area overhead

Table 7 Comparison of router control state and implementation complexity

Algorithm	Local fault vector	Global/Multi-hop fault information	Additional control state	Data path modification
HOE	No	No	No	No
OFTR	4-bit	No	Small	No
FTHOE	4-bit	No	Small	No
LBFT	No	Yes	Large	No

Table 8 Area comparison of different routing algorithms (unit: mm^2)

Algorithm	Router area	Link area	Total area
LBFT	4.5782	0.0202	4.5984
OFTR	4.5687	0.0202	4.5889
FTHOE	4.5687	0.0202	4.5889

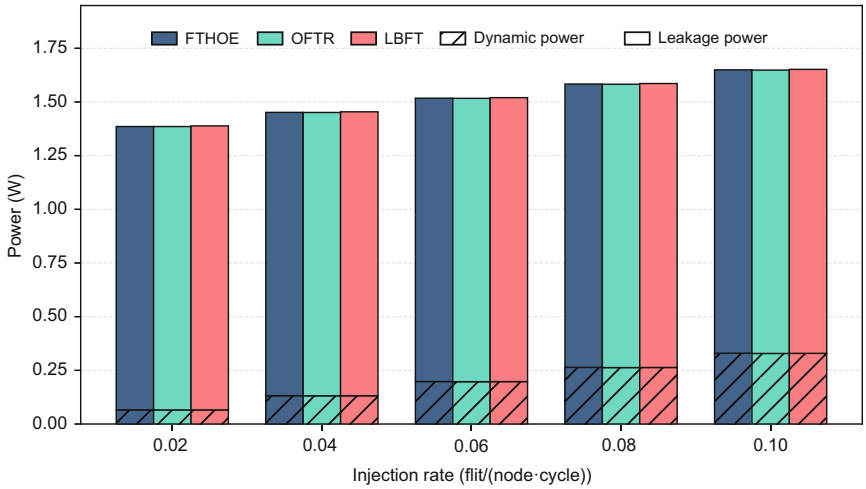

Fig. 15 Dynamic and leakage power breakdown of FTHOE, OFTR, and LBFT under different injection rates

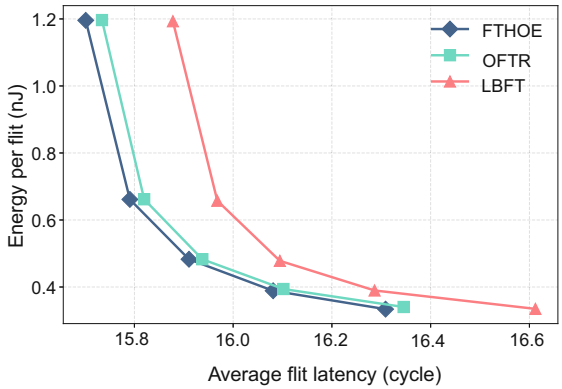

Fig. 16 Energy–latency trade-off of different routing algorithms at injection rates of 0.02–0.10 flits/(node·cycle)

of LBFT is higher than that of the other two algorithms because it introduces larger fault-information tables and additional control-state storage structures in each router, thereby substantially increasing the area of the control logic and associated storage elements. In contrast, FTHOE and OFTR have nearly identical area overheads and remain at the same order of magnitude. It should be noted that this increase mainly arises from the introduction of a 4-bit local fault-vector storage and a small amount of combinational control logic to support routing decisions on top of the HOE baseline router, rather than from changes to the data path structure. Since FTHOE reuses a local fault-awareness mechanism of the same granularity as OFTR and does not introduce global or multi-hop fault-information tables, its router area overhead remains at the same order of magnitude as that of OFTR and is significantly lower than that of LBFT. Overall, FTHOE achieves fault awareness and path-optimization capabilities without incurring significant hardware area overhead.

6.4 Sensitivity study

This subsection evaluates the robustness of the proposed FTHOE routing algorithm under representative variations of key architectural parameters as well as fault-related operational conditions. Instead of exhaustive parameter sweeps, we adopt a one-factor-at-a-time methodology and focus on trend-level analysis, which is sufficient for assessing scalability and robustness while keeping the simulation cost manageable.

We additionally examine the impact of buffer depth and link bandwidth variations within the evaluated parameter ranges; however, these changes do not alter the relative performance trends among routing algorithms. Therefore, to avoid redundancy, we focus the architectural-parameter sensitivity analysis on network size and router latency, which have more pronounced effects on scalability and throughput.

6.4.1 Network size scaling

The impact of network size on routing performance is examined by scaling the topology from an 8×8 mesh to 16×16 and 32×32 meshes under the uniform traffic pattern. To maintain comparable fault pressure across different scales, the number of injected faults is scaled proportionally with network size: one faulty node and one faulty link for the 8×8 mesh, two faulty nodes and two faulty links for the 16×16 mesh, and four faulty nodes and four faulty links for the 32×32 mesh.

For fair comparison across network sizes, the delivered throughput is normalized, for each routing algorithm, to its corresponding value in the 8×8 network. This normalization removes the effect of absolute node count and enables direct comparison of relative performance degradation as routing distances and contention increase.

As shown in Fig. 17, increasing the network size results in a substantial reduction in normalized throughput for all evaluated routing algorithms, which is expected due to larger average path lengths and intensified resource contention in larger meshes. Nevertheless, the relative performance ordering remains consistent across all evaluated scales. In particular, FTHOE consistently achieves the highest normalized throughput and exhibits a more gradual performance degradation as network size increases. These results suggest that the benefits of the proposed local fault-aware routing strategy are not confined to small-scale networks and remain observable as the network size and fault count increase proportionally.

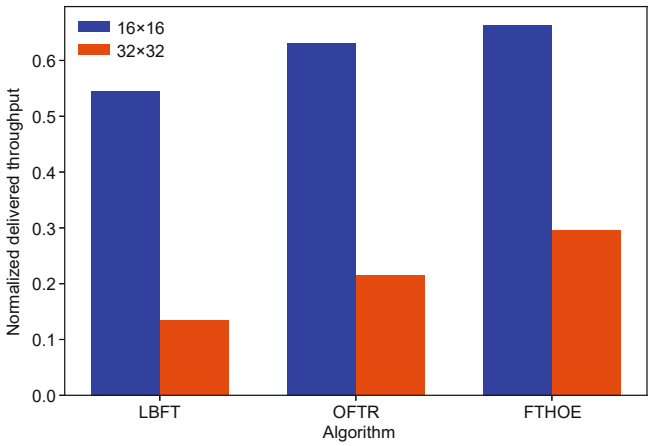

Fig. 17 Normalized delivered throughput under different network sizes (uniform traffic), normalized to the 8 × 8 baseline

6.4.2 Router latency

We evaluate the sensitivity of routing performance to router latency (i.e., pipeline depth in cycles) by varying the router latency from 1 to 2 cycles in an 8 × 8 mesh under uniform traffic. Increasing router latency introduces additional per-hop traversal delay, thereby increasing the in-network residence time of flits and reducing the achievable throughput.

The throughput results under varying router latency are summarized in Table 9. As expected, larger router latencies lead to a clear throughput degradation for all routing algorithms. Nevertheless, FTHOE consistently maintains the highest delivered throughput across different router latencies. This observation indicates that the effectiveness of FTHOE is robust to variations in router microarchitecture and does not rely on aggressively shallow pipeline designs.

Table 9 Impact of router latency on delivered throughput (uniform traffic, 8×8 mesh, single-node and single-link faults)

Algorithm	Throughput (flit/(node·cycle))	
	1 cycle	2 cycles
LBFT	0.1455	0.1080
OFTR	0.1541	0.1276
FTHOE	0.1627	0.1360

Overall, the sensitivity analysis demonstrates that FTHOE maintains stable performance advantages across variations in network scale and router latencies. While absolute throughput naturally decreases as network size increases or router latencies become larger, the relative performance trends among routing algorithms remain consistent. These results confirm that the proposed routing algorithm is robust to realistic architectural variations and is suitable for deployment in larger-scale and fault-prone on-chip networks.

6.4.3 Structured spatial fault sensitivity

In addition to parameter-level sensitivity, we examine the robustness of FTHOE under structured spatial fault patterns, which are commonly observed in wafer-scale manufacturing due to correlated defect mechanisms. Unlike isolated node or link faults, spatially contiguous faults impose stronger constraints on routing flexibility and are known to challenge routing strategies that rely primarily on local heuristics.

We evaluate three representative structured spatial fault patterns on an 8 × 8 mesh: a central 2 × 2 faulty block, a central 3 × 3 faulty block, and a contiguous 1 × 6 fault "crack" aligned along one dimension. All simulations are conducted using uniform random traffic at a fixed representative injection rate of 0.02 flits/(node·cycle). Each configuration is simulated for 20 independent runs, and the evaluation focuses on packet reachability and bounded latency degradation, rather than throughput saturation.

Table 10 summarizes the averaged injected and received flits, as well as the average packet latency, under the evaluated structured fault patterns. The results show that FTHOE successfully delivers all injected packets in all cases, with injected and received flit counts remaining closely matched, indicating the absence of deadlock or livelock. As fault severity increases, average packet latency rises due to enforced detours around faulty regions; however, the observed latency degradation remains bounded and scales with the size and structure of the faulty region.

Table 10 Robustness of FTHOE under structured spatial fault patterns

Fault pattern	No. of injected packets	No. of received packets	Average latency (cycle)
None	128 245	128 230.0	15.58
2 × 2 block	120 297	120 283.7	16.49
3 × 3 block	110 139	110 120.6	17.23
1 × 6 crack	116 233	116 210.8	19.48

These results indicate that the Hamiltonian-guided routing structure of FTHOE, combined with local fault awareness, is sufficient to preserve packet reachability and stable operation even in the presence of non-isolated, region-level defects that exhibit strong spatial correlation. This evaluation is intentionally framed as a robustness stress test for FTHOE. Consequently, routing schemes primarily designed for isolated fault scenarios are not considered under these structured patterns, as their behaviors under large correlated defect regions lie outside their original design assumptions.

6.4.4 Delayed and stale local fault awareness

In practical wafer-scale and large-scale NoC systems, fault detection and status propagation are inherently asynchronous. Routing decisions may therefore rely on delayed or locally stale fault information, rather than globally consistent and instantaneous fault updates. To evaluate the robustness of FTHOE under such conditions, we explicitly model delayed local fault awareness by allowing each router to maintain a cached local directional fault vector that is refreshed at a configurable interval.

In this simulation, a single-node fault (node 44) is activated at a known time T_{fault}. To isolate the impact of delayed fault awareness from pre-fault steady-state behavior, performance statistics are reset at T_{fault}, and only post-fault behavior

is reported. We consider two representative injection rates (0.08 and 0.12 flits/(node·cycle)) and compare instantaneous local updates (0 delay) against a highly delayed update interval of 20 000 cycles, which serves as a conservative stress-test configuration for stale fault information.

Table 11 reports the post-fault average flit latency under different refresh intervals. Across both load points, increasing the refresh interval from 0 to 20 000 cycles results in negligible latency variation (below 0.02%), with no consistent degradation trend observed. This indicates that delayed local fault awareness primarily introduces transient and localized routing detours, while the overall network performance degradation remains tightly bounded within the tested load range.

Table 11 Post-fault latency sensitivity to stale local fault-awareness refresh interval (single-node fault, hotspot traffic, 8 × 8 mesh)

Injection rate	Latency (cycle)		Sensitivity (%)
(flit/(node·cycle))	0	20 000	
0.08	16.8085	16.8108	+0.0137
0.12	17.4258	17.4243	−0.0085

Importantly, all routing decisions in FTHOE continue to strictly obey the HOE turn constraints, ensuring deadlock freedom and packet reachability regardless of fault-awareness delay. These results validate the practicality of the local fault vector premise adopted in FTHOE: even under highly delayed or stale fault updates, correctness is preserved, and performance impact remains minimal for the tested single-node fault scenario.

6.5 Applicability limits and fault-awareness assumptions

The proposed FTHOE routing algorithm is developed under a 2D mesh abstraction, which is commonly adopted as a logical routing model for analyzing wafer-scale interconnection networks. This abstraction enables the use of HOE turn constraints to guarantee deadlock-free operation while preserving minimal-path routing. Note that this abstraction is intentionally restrictive and serves as an analytical model for studying routing behavior, rather than implying that practical wafer-scale systems strictly adhere to a regular physical mesh topology.

1. Applicability to irregular and heterogeneous systems

FTHOE applies to grid-like topologies with bounded irregularity, such as meshes containing defective nodes or links, which frequently arise due to manufacturing imperfections in wafer-scale systems. In such scenarios, the Hamiltonian embedding and HOE turn constraints remain valid as long as the underlying connectivity preserves a mesh-like structure. More general irregular topologies that cannot be reasonably embedded into a Hamiltonian path without significant distortion are beyond the scope of this work.

With respect to multi-die or chiplet-based wafer-scale architectures, FTHOE operates purely at the logical routing level and relies only on relative node positions and locally available fault vectors. Consequently, it does not explicitly model computational heterogeneity or variations in link bandwidth across dies. While such heterogeneity may affect end-to-end performance through workload placement and traffic mapping, it does not compromise the correctness, reachability, or deadlock-free guarantee of the routing algorithm.

2. Assumptions on link delays and reconfigurable interconnect layers

The current design assumes uniform router and link delays, which is a standard modeling choice in turn-model-based NoC routing studies and facilitates tractable analysis of routing behavior. Extending FTHOE to explicitly account for non-uniform or weighted link delays would require incorporating latency-aware cost metrics into the routing decision process. Such extensions are orthogonal to the fault-tolerant routing mechanisms investigated in this study and are therefore not considered here.

Furthermore, FTHOE assumes a quasi-static logical topology during routing epochs. In systems equipped with reconfigurable interconnect layers or SDSoW-like fabrics, dynamic changes in logical connectivity may invalidate the original Hamiltonian numbering and the associated HOE turn constraints. In such cases, the Hamiltonian embedding and routing constraints would need to be recomputed when the logical topology undergoes significant reconfiguration. Efficient runtime re-embedding strategies and consistency management under dynamic reconfiguration remain important directions for future research.

3. Asynchronous and imperfect fault awareness

In practical wafer-scale and SDSoW systems, fault detection and status propagation are inherently asynchronous, and routing decisions may rely on stale or locally inconsistent fault information. FTHOE is deliberately designed to operate based solely on locally available fault vectors and does not assume globally synchronized fault updates.

When fault awareness is delayed or partially inconsistent, FTHOE may temporarily select suboptimal routing paths, resulting in increased path stretch or latency. However, because all routing decisions strictly adhere to deadlock-free HOE turn constraints, such inconsistencies primarily impact performance rather than functional correctness. Packet reachability and deadlock freedom are preserved even under imperfect fault information. Explicit modeling of fault-detection latency, propagation delay, and transient fault behaviors is left for future work.

7 Conclusions

This paper addresses wafer-scale interconnection networks where node and link faults coexist and introducing VCs or high hardware redundancy is undesirable. A VC-less fault-tolerant routing algorithm, termed FTHOE, is proposed based on the HOE turn model. By reordering port priorities using local fault vectors, FTHOE improves minimal-path reachability and path diversity under fault conditions while preserving deadlock-free operation. Theoretical analysis and empirical evaluation indicate that FTHOE avoids deadlock and livelock under finite faults and guarantees packet reachability. Gem5-based simulations demonstrate that FTHOE reduces network latency and maintains higher saturation throughput under the fault-free, single-fault, and mixed node–link fault scenarios, with limited

hardware overhead. Since FTHOE relies only on local fault information, delayed or partially inconsistent fault awareness primarily affects performance bounds rather than correctness, making it suitable for practical wafer-scale deployments. Future work will extend this approach to larger-scale networks and more dynamic fault environments.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2023YFB4404200).

Author contributions

Shuaikang HOU designed the research, conducted theoretical analysis and simulations, and drafted the paper. Qinrang LIU and Ping LV provided guidance and reviewed the paper. Wenbo ZHANG assisted with simulations and data organization. Peijie LI and Wei GUO revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declaration on the use of generative AI tools

During the preparation of this work, the authors used ChatGPT to improve the language and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Agarwal N, Krishna T, Peh LS, et al., 2009. GARNET: a detailed on-chip network model inside a full-system simulator. IEEE Int Symp on Performance Analysis of Systems and Software, p.33-42. https://doi.org/10.1109/ISPASS.2009.4919636

Bahrebar P, Stroobandt D, 2015. The Hamiltonian-based odd–even turn model for maximally adaptive routing in 2D mesh networks-on-chip. *Comput Electr Eng*, 45:386-401. https://doi.org/10.1016/j.compeleceng.2014.12.009

Bohr M, 2007. A 30 year retrospective on Dennard's MOSFET scaling paper. *IEEE Sol-State Circ Soc Newsl*, 12(1):11-13. https://doi.org/10.1109/N-SSC.2007.4785534

Charif A, Zergainoh NE, Nicolaidis M, 2016. Addressing transient routing errors in fault-tolerant networks-on-chips. 21st IEEE European Test Symp, p.1-6. https://doi.org/10.1109/ETS.2016.7519289

Dally WJ, Seitz CL, 1987. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans Comput*, C-36(5):547-553. https://doi.org/10.1109/TC.1987.1676939

Daneshtalab M, Ebrahimi M, Xu TC, et al., 2011. A generic adaptive path-based routing method for MPSoCs. *J Syst Architect*, 57(1):109-120. https://doi.org/10.1016/j.sysarc.2010.08.002

Ebrahimi M, Daneshtalab M, 2015. A light-weight fault-tolerant routing algorithm tolerating faulty links and routers. *Computing*, 97(6):631-648. https://doi.org/10.1007/s00607-013-0362-9

Guan J, Cai JP, Wang YQ, et al., 2023. A low-cost oblivious and fault-tolerant routing strategy for NoCs. *J Air Force Eng Univ*, 24(1):95-102 (in Chinese). https://doi.org/10.3969/j.issn.2097-1915.2023.01.014

Hu Y, Lin XH, Wang HZ, et al., 2024. Wafer-scale computing: advancements, challenges, and future perspectives. *IEEE Circ Syst Mag*, 24(1):52-81. https://doi.org/10.1109/MCAS.2024.3349669

Jerger NE, Kannan A, Li ZM, et al., 2014. NoC architectures for silicon interposer systems: why pay for more wires when you can get them (from your interposer) for free? 47th Annual IEEE/ACM Int Symp on Microarchitecture, p.458-470. https://doi.org/10.1109/MICRO.2014.61

Joshi B, Thakur MK, 2023. A traffic intensive virtual channels allocation scheme in network-on-chip. *Arab J Sci Eng*, 48(8):9619-9633. https://doi.org/10.1007/s13369-022-07191-9

Lowe-Power J, Ahmad AM, Akram A, et al., 2020. The gem5 simulator: version 20.0+. https://doi.org/10.48550/arXiv.2007.03152

Mohapatra H, Rath AK, 2019. Fault tolerance in WSN through PE-LEACH protocol. *IET Wirel Sens Syst*, 9(6):358-365. https://doi.org/10.1049/iet-wss.2018.5229

Moore GE, 1998. Cramming more components onto integrated circuits. *Proc IEEE*, 86(1):82-85. https://doi.org/10.1109/JPROC.1998.658762

Nehnouh C, Senouci M, 2019. A new fault tolerant routing algorithm for networks on chip. *Int J Embed Real-Time Commun Syst*, 10(3):68-85. https://doi.org/10.4018/IJERTCS.2019070105

Pal S, Petrisko D, Tomei M, et al., 2019. Architecting waferscale processors—a GPU case study. IEEE Int Symp on High Performance Computer Architecture, p.250-263. https://doi.org/10.1109/HPCA.2019.00042

Pal S, Liu JY, Alam I, et al., 2021. Designing a 2048-chiplet, 14336-core waferscale processor. 58th ACM/IEEE Design Automation Conf, p.1183-1188. https://doi.org/10.1109/DAC18074.2021.9586194

Rahaman MM, Ghosal P, Das TS, 2019. Latency, throughput and power aware adaptive NoC routing on orthogonal convex faulty region. *J Circ Syst Comput*, 28(4):1950055. https://doi.org/10.1142/S0218126619500555

Renani NB, Yaghoubi E, Sadehnezhad N, et al., 2022. NLR-OP: a high-performance optical router based on North-Last turning model for multicore processors. *J Supercomput*, 78(2):2442-2476. https://doi.org/10.1007/s11227-021-03920-3

Reza A, Jolani P, Reshadi M, 2019. CAFT: cost-aware and fault-tolerant routing algorithm in 2D mesh network-on-chip. *J Adv Comput Eng Technol*, 5(4):205-212.

Wu JX, Liu QR, Shen JL, et al., 2024. From SoC to SDSoW: a new paradigm for microelectronics development. *Sci Sin Inform*, 54(6):1350-1368 (in Chinese). https://doi.org/10.1360/SSI-2023-0219

Xie RL, Cai JP, Xin X, 2016. Simple fault-tolerant method to balance load in network-on-chip. *Electron Lett*, 52(10):814-816. https://doi.org/10.1049/el.2015.3150

Xie RL, Cai JP, Xin X, et al., 2018. LBFT: a fault-tolerant routing algorithm for load-balancing network-on-chip based on odd–even turn model. *J Supercomput*, 74(8):3726-3747. https://doi.org/10.1007/s11227-016-1935-0

Xu Z, Kong DH, Liu JX, et al., 2025. WSC-LLM: efficient LLM service and architecture co-exploration for wafer-scale chips. Proc 52nd Annual Int Symp on Computer Architecture, p.1-17. https://doi.org/10.1145/3695053.3731101

Yang QZ, Wei TQ, Guan SH, et al., 2025. PD constraint-aware physical/logical topology co-design for network on wafer. Proc 52nd Annual Int Symp on Computer Architecture, p.49-64. https://doi.org/10.1145/3695053.3731045

Yu XM, Jiang DC, Deng JY, et al., 2025. Cramming a data center into one cabinet, a co-exploration of computing and hardware architecture of waferscale chip. Proc 52nd Annual Int Symp on Computer Architecture, p.631-645. https://doi.org/10.1145/3695053.3731016

Zhang YJ, Fan WB, Han ZJ, et al., 2021. Fault-tolerant routing algorithm based on disjoint paths in 3-ary n-cube networks with structure faults. *J Supercomput*, 77(11):13090-13114. https://doi.org/10.1007/s11227-021-03799-0