Path-Diversity-Aware Fault-Tolerant Routing Algorithm for Network-on-Chip Systems

Yu-Yin Chen, En-Jui Chang, Hsien-Kai Hsin,
Kun-Chih (Jimmy) Chen, *Member, IEEE*, and An-Yeu (Andy) Wu, *Fellow, IEEE*

Abstract—Network-on-Chip (NoC) is the regular and scalable design architecture for chip multiprocessor (CMP) systems. With the increasing number of cores and the scaling of network in deep submicron (DSM) technology, the NoC systems become subject to manufacturing defects and have low production yield. Due to the fault issues, the reduction in the number of available routing paths for packet delivery may cause severe traffic congestion and even to a system crash. Therefore, the fault-tolerant routing algorithm is desired to maintain the correctness of system functionality. To overcome fault problems, conventional fault-tolerant routing algorithms employ fault information and buffer occupancy information of the local regions. However, the information only provides a limited view of traffic in the network, which still results in heavy traffic congestion. To achieve fault-resilient packet delivery and traffic balancing, this work proposes a Path-Diversity-Aware Fault-Tolerant Routing (*PDA-FTR*) algorithm, which simultaneously considers path diversity information and buffer information. Compared with other fault-tolerant routing algorithms, the proposed work can improve average saturation throughput by 175 percent with only 8.9 percent average area overhead and 7.1 percent average power overhead.

Index Terms—Network-on-Chip (NoC), fault-tolerant adaptive routing, selection strategy, path diversity

1 Introduction

With the advance of semiconductor technology, the increased hardware complexity and interconnection delay become the limiting factors of System-on-Chip (SoC) performance. To increase the efficiency of interconnections and meet data transfer requirements, the Network-on-Chip (NoC) architecture has been proven as a flexible, scalable, and reusable solution for chip multiprocessor (CMP) systems [1], [2]. However, with the increasing number of cores and the scaling of network in deep submicron (DSM) technology, the NoC system is subject to manufacturing defects [3], and hence becomes a faulty NoC. Intel commentators forecast that one-fifth of 100 billion transistors will fail in manufacturing process [4]. These faulty NoC systems may have defects in processing elements (PE), routers and/or interconnections. The problem with defective PEs can be solved through disabling the faulty PEs and task re-mapping [5]. However, the problem of defective routers and interconnections is critical. Defects in these components may reduce the number of available routing paths, which impairs packet delivery and even causes system failure.

Due to the faulty routers and faulty interconnections, the number of packet routing paths are decreased, which results in highly unbalanced traffic distribution and heavy traffic congestion. The path congestion spreads from a faulty region to source nodes, which grows into a congestion tree and worsen the traffic congestion in the faulty network. To overcome aforementioned problem, a fault-tolerant routing algorithm is required. Fig. 1 illustrates an example to deliver a packet from its source (0, 4) to the destination (4, 0) by applying odd-even routing turn model [6] in a 5-by-5 faulty mesh. Due to the involved odd-even turn model, some routing paths are forbidden. Figs. 1a, 1b, and 1c illustrate the different decisions of routing direction between our design and conventional fault-tolerant routing algorithms. To precisely evaluate traffic condition, many researchers usually use local buffer occupancy (*BO*) information [7], [8]. *BO* is effective in reflecting the dynamically changing traffic. However, it is a local information that only provides a limited view of network traffic condition, which results in non-effective routing path decision [9], [10], [11]. Fig. 1a demonstrates the routing decision made by conventional *BO*-based fault-tolerant routing algorithms employing local *BO* information and local fault information. As shown in Fig. 1a, the packet is sent toward the East because 1) the faulty router is three-hop away from the source router, thus, its fault notification is out-of-reach, and 2) there are more free buffer slots in the East than that in the South. Because of the small routing flexibility, this kind of routing decision results in heavy traffic congestion in the East.

On the other hand, the authors in [12] purposed path diversity (*PD*)-based design concept. *PD*, defined as all the possible paths that a packet can be routed given a source-destination pair, is a static information directly related to the adaptiveness of routing paths. Fig. 1b demonstrates the routing decision of *PD*-based fault-tolerant routing algorithms. Although the packet is informed about the distant

- *Y.-Y. Chen, E.-J. Chang, H.-K. Hsin, and A.-Y. (Andy) Wu are with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan. E-mail: {yyjasmine, enjui}@access.ee.ntu.edu.tw, ckcraig@access.ee.ntu.edu, andywu@ntu.edu.tw.*
- *K.-C. (Jimmy) Chen is with the Computer Science and Engineering Department, National Sun Yat-Sen University, Kaohsiung City 804, Taiwan. E-mail: kcchen@mail.cse.nsysu.edu.tw.*

Manuscript received 12 Sept. 2015; revised 25 June 2016; accepted 1 July 2016. Date of publication 7 July 2016; date of current version 15 Feb. 2017. Recommended for acceptance by M. Huebner.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2016.2588482

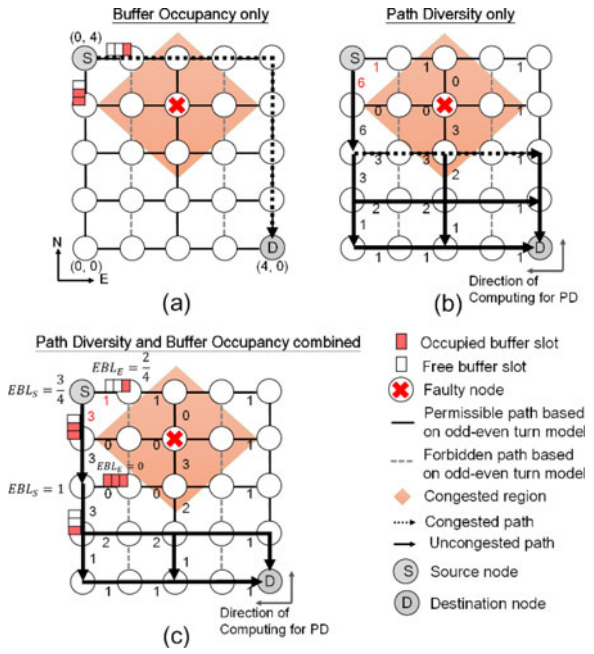

Fig. 1. Routing decisions made by (a) BO-based fault-tolerant routing results in low routing flexibility, (b) PD-based fault-tolerant routing results in poor routing path quality in the congested faulty region, and (c) the proposed *PDA-FTR* simultaneously considers both BO and PD.

fault, it may still encounter congested region in the faulty region in which the congestion tree is easily grown. In summary, these conventional fault-tolerant routing algorithms only take neighboring regions into consideration but ignore the condition when packets go beyond these regions.

To maintain the correctness of system functionality in the faulty NoC (i.e., the NoC system contains several permanent faults) and balance the traffic load, in this work, we propose a Path-Diversity-Aware Fault-Tolerant Routing (*PDA-FTR*) algorithm which simultaneously considers path adaptiveness and routing path quality. In *PDA-FTR*, we integrate *PD* information and *BO* information to acquire the Effective Buffer Length (*EBL*) of routing direction. *EBL* is a measurement of the routing delay, where higher *EBL* implies shorter routing delay. The routing decision made by our proposed routing algorithm is shown in Fig. 1c where the packet is transmitted to a less congested region and away from the faulty region. Our goal is to design a fault-tolerant routing algorithm to efficiently quantify the flexibility of routing function and improve selection quality in order to balance traffic load and achieve fault-resilient data transfer. The main contributions of this paper are summarized as follows

1) **Analysis of Path Diversity:** We introduce a novel index for selecting routing direction called Path Diversity. Besides, we comprehensively analyze two different kinds of fault location and proposed Fault-Location-Based *PD* (*FPD*).
2) **Path-Diversity-Aware Fault-Tolerant Routing Algorithm:** Based on the *FPD*, we propose a fault-tolerant routing algorithm to overcome fault problem and congestion problem in a faulty NoC. In *PDA-FTR*, we propose *EBL* as a metric for path selection. The proposed *PDA-FTR* improves average saturation throughput by 175 percent compared with the existing fault-tolerant routing schemes in different traffic patterns.
3) **VLSI Architecture Design of *PDA-FTR*:** An area-efficient router architecture of *PDA-FTR* algorithm with the proposed regional *PD* table is designed and evaluated using TSMC 90 nm CMOS process. With the proposed regional *PD* table, the *PDA-FTR* router becomes scalable with respect to topology size.

The rest of this paper is organized as follows. We review fault-tolerant routing algorithms in Section 2. Section 3 introduces the concept of *PD* and Fault-Location-Based *PD*. The *PDA-FTR* algorithm is presented in Section 4. The results of various experiments are shown in Section 5 and are followed by *PDA-FTR* router architecture in Section 6. Finally, we conclude this work in Section 7.

2 Review of Related Works

In recent years, many fault-tolerant routing algorithms have been proposed to deal with the problem of unreachable packets due to the occurrence of faulty routers and faulty links in NoCs. Fault-tolerant routing algorithms can be classified into two categories: 1) flooding-based fault-tolerant routing algorithms, and 2) turn model-based fault-tolerant routing algorithms, which are introduced in this section.

2.1 Flooding-based Fault-Tolerant Routings

Flooding-based fault-tolerant routing algorithms [13], [14] achieve fault-resilient packet delivery through sending redundant replicate packets over different routes to increase the probability of successful transmission. However, these approaches have high chance to encounter deadlock and livelock. In addition, these algorithms lead to significant area and storage overhead as well as power consumption. Therefore, we do not consider such high-cost solution in the resource-limited NoC system.

2.2 Turn Model-Based Fault-Tolerant Routings

The turn model-based fault-tolerant routing algorithms prohibit packets making certain turns to avoid deadlock, which leave some paths untaken and hence lead to unbalanced traffic load.

Modified X-First considers the fault location when deciding the output direction [15]. However, it is deterministic and can only handle one-fault system (i.e., system has only one faulty router). *Gradient* is an adaptive routing which considers various alternative paths for the packet delivery when the main routing path exists faulty routers [16]. It divides the system by Gradient Line into eight zones and each zone is provided with one main path and two alternative paths with different routing priorities. Minimal and defect-resilient *(MD)* routing algorithm routes packet adaptively through the shortest path and each router possesses fault condition within two-hop links [17]. Furthermore, *MD* considers buffer information and chooses the path with the greatest distance dimension between source and destination. In [18], the Hamiltonian-based Odd-Even (*HOE*) routing algorithm was proposed. Based on the Hamiltonian path graph, the network can be identified into two subnetworks. By applying odd-even turn to the two subnetworks, *HOE* routing algorithm can increase the degree of routing adaptiveness. Because turn model-based fault-tolerant routing algorithms are deadlock-free, no additional deadlock-recovery mechanism is required.

TABLE 1
Turn Model-Based Fault-Tolerant Routing Algorithms

Fault-Tolerant Routing Algorithms	Routing Function	Selection Function	
		Fault Info.	Traffic Info.
Modified X-First [15]	Deterministic	+	-
Gradient [16]	Adaptive	+	+
MD [17]	Adaptive	+	+
HOE [18]	Adaptive	-	+
Proposed *PDA-FTR*	Adaptive	++	+

-: Information is not available.
+: Local information (one or two-hop information).
++: Global information.

2.3 Analysis of Conventional Fault-Tolerant Routing Algorithms

Table 1 summarizes the attributes of the above-mentioned related works. As shown in Table 1, the conventional fault-tolerant routing algorithms have limited fault information and restricted routing paths, which results in severe performance degradation in faulty NoC. The detailed reasons are summarized below:

- **Lack of regional/global fault information:** The previously mentioned works only possess fault information within one to two hops distance. However, these kinds of local fault information make the routing packet cannot evade from faults in the early stage. The packet latency increases as a large number of routing packets are blocked around the faulty nodes.
- **Restricted path diversity and emergence of traffic hotspot around fault:** In conventional fault-tolerant routing algorithms, especially deterministic fault-tolerant routing, each packet only has a single path for detouring when it encounters faulty nodes. This reduction in path diversity causes traffic congestion around the faulty nodes. Moreover, congestion may be propagated and worsen the system performance.

To deal with such problems, we adopt *PD* information which measures the freedom degree of the routing paths. Our goal is to obtain large-scale fault information through integrating fault information into *PD*. In addition, we provide more detour paths to reduce the probability of forwarding packet toward the regions around faults.

3 Analysis of Path Diversity for Path Selection

In this section, we first define *PD* and analyze its property. Then, we introduce Fault-Location-Based *PD* and will propose Path-Diversity-Aware Fault-Tolerant Routing algorithm in Section 4 based on this analysis.

3.1 Definition of Path Diversity

The *PD* indicates the degree of freedom of a route, which is influenced by three factors: 1) the routing algorithms, 2) the network topologies, and 3) the distances between local routers and destinations. Because there is a non-linear relationship between these three factors and *PD*, we simply define *PD* as a function below

$$PD = f(R, T, D(local\ node,\ destination\ node)), \quad (1)$$

where *R* is the usable path given by turn model; *T* is a factor related to the topology and its size; *D* is the routing distance in *x* and *y* directions for each pair of communicating nodes (i.e., *local node* and *destination node*). From the view of routing algorithms, lesser turn model restrictions leads to higher *PD* because of bigger *R*. From the view of topology, if a network has more connections between adjacent nodes, the *PD* is higher because of bigger *T*. Besides, a long-distance source-destination pair leads to bigger *D* than a short-distance one, which increases *PD* value.

Obviously, based on the concept (1), we can only make *PD* as a function of the combination of routing algorithms *R* and routing distance *D* as computing the exact number of possible path for a given topology *T*. For example, if the factor *R* is the partially adaptive routing function, which adopts odd-even turn model, and factor *T* is a mesh-based topology, the exact path diversity for packets going in the East direction can be formulated by [6]

$$PD = \frac{[h' + (hc_y)]!}{h'!(hc_y)!} \quad (2)$$

and

$$h' = \begin{cases} \dfrac{hc_x - 1}{2}, \text{if } hc_x \text{ is odd} \\ \dfrac{hc_x}{2}, \text{if } hc_x \text{ is even} \end{cases}, \quad (3)$$

where hc_x and hc_y are hop counts between source and destination in the *x* and *y* direction, respectively. Both hc_x and hc_y imply the topology and distance. The two possible choices of h' are a result of the restrictions of odd-even turn model [6], which is the factor *R*. The *PD* represents the number of routing paths which we can use for delivering the packets until reaching the destination.

Faults reduce the available routing paths and hence reduce *PD*. This *PD* information is computed after chip testing and fault diagnosis. Chip testing and fault diagnosis can be done with Built-in-self-diagnosis (BISD) based mechanisms or Design-for-test (DFT) based mechanisms to locate the faulty routers [21]. With this fault information, *PD* is computed during system warmup time. Obviously, due to the warmup-time computation, it does not give the system extra burdens during system operation while offering extra information.

By using (2), Fig. 2 illustrates an example to compute *PD*. Fig. 2 shows the *PD* of fully adaptive routing function for a packet going from local node *Lcl* at (1,4) to its destination *Dst* at (4,0). For fully adaptive routing function, it has no restriction in both *x* and *y* direction in every node; thus, we expect to have equal *PD* in output directions for any given node. However, Fig. 2 does not show the result as what we have expected. The reason is that the routing distance *D* is involved in our calculation of *PD* by using (2). As shown in Fig. 2, the distance between the local router and the destination in *x* and *y* direction are 3 and 4, respectively. Therefore without normalization, the *y* direction has higher *PD* than the *x* direction. To make *PD* only be affected by the factor *R*, which is the restriction in turn model, we need to normalize *PD* in order to eliminate the effect of routing distance *D* in the computation of *PD*.

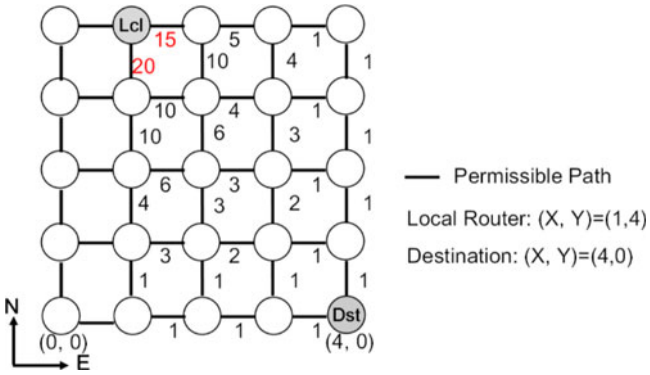

Fig. 2. The schematic sketch of path diversity for a fully adaptive routing function with the destination (4, 0).

3.2 Normalization of Path Diversity

Assume that there is a packet going from the local router to destination. The hop counts from the local router to its destination in the x and y direction are hc_x and hc_y, respectively. In the case of fully adaptive routing, the path diversity in x and y direction are

$$PD_x = \frac{[(hc_x - 1) + (hc_y)]!}{(hc_x - 1)!(hc_y)!} \tag{4}$$

and

$$PD_y = \frac{[(hc_x) + (hc_y - 1)]!}{(hc_x)!(hc_y - 1)!}, \tag{5}$$

which are not the same in the denominator. To make them identical to each other, we can add the terms hc_x and hc_y in (4) and (5) to obtain normalized path diversity (*NPD*),

$$NPD_x = \frac{[(hc_x - 1) + (hc_y)]!}{(hc_x - 1)!(hc_y)!(hc_x)} = \frac{[(hc_x - 1) + (hc_y)]!}{(hc_y)!(hc_x)!} \tag{6}$$

and

$$NPD_y = \frac{[(hc_x) + (hc_y - 1)]!}{(hc_x)!(hc_y - 1)!(hc_y)} = \frac{[(hc_x) + (hc_y - 1)]!}{(hc_x)!(hc_y)!}. \tag{7}$$

The added terms mean the hop counts in the x and y direction. Therefore, we refine the calculation of *PD* by dividing a distance factor (*DF*). The *NPD* can be defined as

$$NPD = \frac{PD}{DF}, \tag{8}$$

where

$$DF = \begin{cases} hc_x, & \text{for East or West paths} \\ hc_y, & \text{for North or South paths.} \end{cases} \tag{9}$$

This normalization makes the derived path diversity be only affected by the imbalanced property of the routing function. Note that the *DF* provided above is for regular mesh and may vary in other topologies.

By utilizing *NPD*, we can extract the real *PD* distribution in different directions. For instance, a path with higher *NPD* means that routing function permits more possibilities in this direction, which is inherently robust for heavy load.

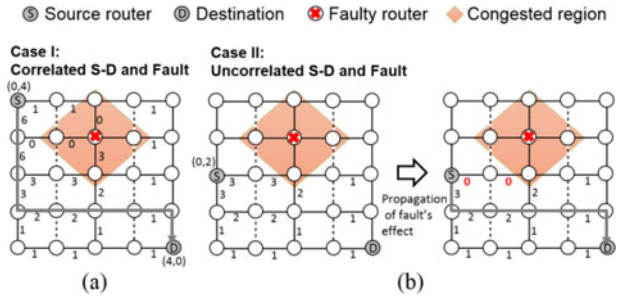

Fig. 3. Fault and source-destination pair are correlated in (a) and are uncorrelated in (b).

Hence, we can select a path with higher *NPD* to deliver packets when there are two candidates.

3.3 Fault-Location-Based Path Diversity

As mentioned before, *NPD* represents the number of available paths from current node to destination node. Because the traffic hotspot is easily occurred around the faulty nodes, the faulty paths including faulty routers should be isolated to ensure the connectivity of these available paths. Hence, it is necessary to remove the influence of the faulty paths in *NPD* computation and prevent the packets from routing toward the congestion region around the faults. In this work, we consider the fault locations in *NPD* computation and define the new metric as Fault-Location-Based *PD*. Based on the *FPD* information, we can not only deliver the packets to the highest path diversity but also have better capability to detour from the faulty nodes.

We explain the *FPD*-based in the following two cases:

- **Case I: Correlated Source-Destination Pair and Fault:** Fault locates in between source-destination pair as shown in Fig. 3a, thus they are correlated. Because we re-compute *NPD* through eliminating the faulty paths, *NPD* is reset to zero when encountering faults. Hence, as using odd-even turn model, the *FPD* of (0, 4) in the South direction is expressed as

$$FPD_S^{(x,y)} = \begin{cases} FPD_E^{(x,y-1)} + FPD_S^{(x,y-1)}, & \text{if router } (\mathrm{x}+1, \mathrm{y}-1) \text{ and} \\ & (\mathrm{x}, \mathrm{y}-2) \text{ are not faulty} \\ 0, & \text{Otherwise,} \end{cases} \tag{10}$$

  where x is 0 and y is 4. The superscript of *FPD* is the location of the router, and the subscript of *FPD* is the direction. Since router (2, 3) is a faulty router, the $FPD_E^{(x,y-1)}$ is reset to 0 and

$$FPD_S^{(x,y)} = FPD_S^{(x,y-1)}. \tag{11}$$

  By using the proposed *PD* re-computation (i.e., *FPD*), the *FPD* information can help the routing packets.
- **Case II: Uncorrelated Source-Destination Pair and Fault:** When fault locates outside of the minimal routing region between source and destination, we should thoroughly exploit the fault information. Fig. 3b illustrates an example. Based on (6) and (7), the *NPD* of both candidate channels (i.e., east and

south) at (0, 2) are identical. However, the gradual build-up congestion around the fault does have an impact on the quality of the path. Hence, to reduce the congestion booming rate around the faults, we reduce the probability of forwarding packet toward the congested region. To achieve this goal, we calculate *FPD* of the remaining usable paths after excluding the congested routers around faults. The *FPD* of (0, 2) in the east can be expressed as

$$FPD_E^{(x,y)} = \begin{cases} FPD_E^{(x+1,y)}, & \text{if router } (\mathrm{x}+1,\mathrm{y}) \text{ is not within the congestion region} \\ 0, & \text{Otherwise,} \end{cases} \tag{12}$$

where

$$FPD_E^{(x+1,y)} = \begin{cases} FPD_E^{(x+2,y)} + FPD_S^{(x+2,y)}, & \text{if router } (\mathrm{x}+2,\mathrm{y}) \text{ is not within the congestion region} \\ 0, & \text{Otherwise.} \end{cases} \tag{13}$$

Since router $(x+2, y)$, is within the congested region when x and y are equal to 0 and 2, respectively. Hence, the accumulated *FPD* is equal to zero, which is shown as

$$FPD_E^{(x+1,y)} = 0 \rightarrow FPD_E^{(x,y)} = 0. \tag{14}$$

Because the fault information is included in *FPD* information, each router can identify the better candidate channel for packet delivery. The *FPD* information is stored in the routing table of each router, which will be used for the appliance in routing function.

4 Path-Diversity-Aware Fault-Tolerant Routing Algorithm

Based on the *FPD* information, we propose a Path-Diversity-Aware Fault-Tolerant Routing algorithm. *PDA-FTR* initially adopts minimal routing paths for packet delivery. However, as minimal routing paths are blocked by faults, *PDA-FTR* will employ non-minimal ones to prevent packet congestion in the faulty region. To ensure the deadlock free, in this paper, we involve the non-minimal Odd-Even turn model, which was proved to be both deadlock-free and live-lock-free in [22]. If there are two more routing path candidates, the *PDA-FTR* refers to the information of Effective Buffer Length (*EBL*) in each candidate channel to select the better output channel, which will be introduced in this Section.

4.1 PDA-FTR Using Effective Buffer Length

To route packets toward candidate channel with the highest *FPD*, we use minimal adaptive routing when a minimal path exists as shown in Fig. 4a and 4b. On the other hand, when minimal path is no longer available (i.e., when *FPD* of the minimal direction is zero), we adopt non-minimal adaptive routing or so-called detour to prevent packet stalling as shown in Fig. 4c. In addition, we integrate *FPD* with *BO* information, which is defined as Effective Buffer Length in this paper.

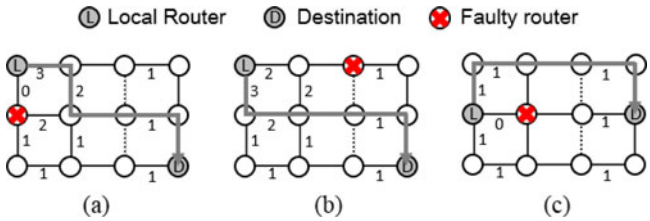

Fig. 4. *PDA-FTR* adopts (a) (b) minimal odd-even routing and adopts non-minimal odd-even routing in (c).

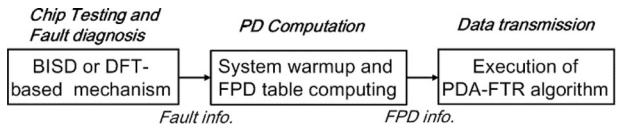

Fig. 5. The overview of system operation.

The high *FPD* of a channel implies taking this direction is likely to avoid downstream congestion and reduce the blocking delay caused by the downstream faulty routers. Moreover, to prevent the traffic congestion in the neighboring routing region around the local router, we need to consider the aforementioned *BO* information, which can reflect the shorter historical information of output port around the local router.

To jointly evaluate the effect of faults and traffic status, we combine the information of *FPD* and *BO* to propose Effective Buffer Length as a simple metric to quantify the congestion level of an output channel. This metric is the weighted free slot determined by *FPD*, which can be expressed as a product term of normalized *FPD* and free slot (i.e., the non-occupied buffer spaces). For example, the selection criteria between eastward and southward output channel are given by

$$EBL(o) = \frac{FPD(o)}{FPD(E) + FPD(S)} \times free\_slot(o) \tag{15}$$

and

$$SOC = \arg\ \max[EBL(o)],\ o \in COC, \tag{16}$$

where the variable o represents the output direction in the local router (i.e., east or south). *SOC* is the selected output channel, and *COC* is the set of candidate output channels provided by the routing function. Therefore, when the routing function returns two or more candidate output directions for a packet, we would select the direction with higher *EBL*.

4.2 Flowchart of PDA-FTR Algorithm

The flowchart of the proposed *PDA-FTR* composes of three processes and the system overview is shown in Fig. 5. After the chip manufacturing, BISD or DFT-based mechanism is used to test chip functionality and diagnose permanent faults. Besides, the BISR mechanism was proposed to further identify the permanent faults after system operation in warmup time [26]. With this fault information, we compute an *FPD* table during system warmup. As shown in Fig. 6a, we first determine whether fault locates in between source and destination or not. If the source-destination pair and fault belong to the correlated case, we reset *FPD* to zero when encounters fault. Otherwise, we eliminate the routing path within the congested region. Then, the *FPD*

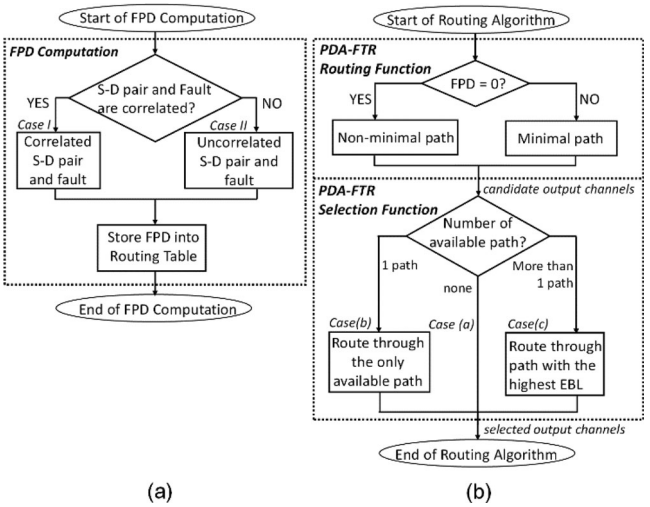

Fig. 6. The flow chart of (a) *FPD* table construction and (b) proposed *PDA-FTR* routing algorithm.

information is used to assist with the routing algorithm. Data transmission of *PDA-FTR* algorithm composes of both routing function and selection function:

1) *PDA-FTR Routing Function:* Based on the information in the built *FPD* table, we can obtain the corresponding *FPD* of each case. Then, routing function searches for all usable paths and excludes the paths with zero *FPD* or faulty paths. In the end, routing function provides all the candidate output channels (i.e., *COC*) to selection function, as shown in Fig. 6b. Because of the table-based *COC* determination, the computational complexity in routing function phase is constant (i.e., *O(1)*).
2) *PDA-FTR Selection Function: PDA-FTR* selection function will evaluate the path availability as shown in Fig. 6b. There are three possible combinations of availability information. Case(a): none of the *COC* is available, Case(b): only one of the *COC* is available, and Case(c): more than one *COC* are available. By considering these different cases, *PDA-FTR* takes corresponding actions:
   - **In Case (a):** *PDA-FTR* does nothing because the packet will be stalled. In the next cycle, *PDA-FTR* will perform selection function again.
   - **In Case (b):** *PDA-FTR* selects the available one right away even if its *EBL* is lower because it is difficult to predict the waiting time for the channel with higher *EBL* to become available. Thus, choosing the available one can add some variability into *PDA-FTR*. Besides, this avoids sending the packets to the same direction every time without increasing the total latency.
   - **In Case (c):** According to the available output channels and destination identification, *PDA-FTR* transmits packets toward *COC* with the highest *EBL*.

5 Performance Evaluation

In this section, we compare *PDA-FTR* with other fault-tolerant routing schemes under different traffic patterns for the single-fault system and for the multiple-fault system. Then, we adopt unreachable packet ratio to evaluate the fault-tolerance capability and employ statistical traffic load distribution to evaluate the traffic balancing ability of *PDA-FTR*. Lastly, we evaluate the performance scalability of *PDA-FTR* using network throughput as a metric.

5.1 Simulation Environment and Setup

The simulation results are evaluated by the SystemC NoC simulator, Noxim [23]. The system runs with the wormhole switching mechanism and matrix arbitration. Each channel has an input buffer with the depth of 4 flits and each packet is 8-flit-long. The experiments use both synthetic and real traffic scenarios to evaluate performance. For the synthetic traffic, we analyze an 8-by-8 mesh network under random, shuffle and bit-reversal traffic. For the real traffic, this study uses a realistic traffic of multimedia systems (MMS) [24] under a 5-by-5 mesh and (1944,972) Low-Density Parity-Check (LDPC) in 802.11n standard [28] under an 8-by-8 mesh to evaluate the performance of the proposed *PDA-FTR* scheme. The simulation time is 12,000 cycles and the first 2,000 cycles are for warming up in order to measure the steady-state performance of the NoC system.

The average packet latency and saturation throughput serve as performance metrics. Saturation throughput is defined as the packet injection rate (*PIR*) where average packet latency worsens to more than twice of zero-load latency (i.e., the average packet latency when there is no serious network congestion) of a baseline fault-tolerant routing scheme [10]. In addition, we evaluate fault-tolerance ability and traffic balancing ability which will be defined later.

5.2 Experimental Results with Synthetic and Real Traffic Data

5.2.1 Performance of PDA-FTR in Single-Fault NoC

The first simulation is the performance comparison employing *Modified X-First* [15], *Gradient* [16] and Minimal and defect-resilient [17] as the competitor algorithms under random, shuffle, bit-reversal, MMS traffic, and LDPC traffic. Fig. 7 demonstrate the average latency under various packet injection rates with one faulty router in the network. Fig. 7a illustrates the average latency of *PDA-FTR* and the other competitor algorithms under random traffic pattern. *Modified X-First* has the worst performance because it is deterministic and has only one candidate output channel. Therefore, congestion builds up easily on certain paths, which easily degrades the performance. On the other hand, *MD* considers both buffer information and location of source-destination pair to determine the best path, which makes better performance. However, its fault information only extends to two-hop link away which limits its performance. *PDA-FTR* outperforms the other competitor algorithms due to having sufficient path diversity and considering both buffer and *PD* information. The improvement of *PDA-FTR* is 13.3-184 percent in terms of saturation throughput comparing to the other algorithms. Fig. 7b illustrates the average latency of *PDA-FTR* under shuffle traffic. When compared to the competitor algorithms, *PDA-FTR* improves 3.9-83.2 percent in terms of saturation throughput. As for bit-reversal traffic shown in Fig. 7c, *PDA-FTR* has an improvement of 15.9-31.7 percent.

As shown in Fig. 7d and 7e, we demonstrate the performance under a realistic MMS traffic and LDPC traffic.

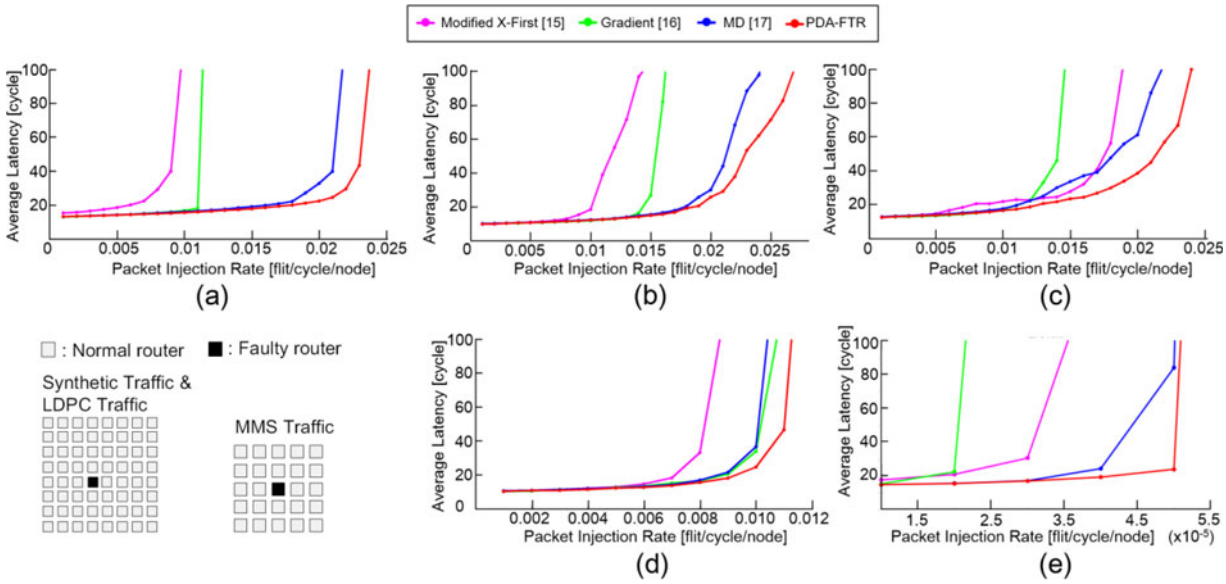

Fig. 7. The average latency of *Modified X-First*, *Gradient*, *MD*, and *PDA-FTR* approaches in single fault system under (a) random traffic, (b) shuffle traffic, (c) bit-reversal traffic, (d) MMS, and (e) (1944, 972) LDPC traffic.

The MMS is an integrated video/audio system that includes an H.263 video encoder, an H.263 video decoder, an MP3 audio encoder, and an MP3 audio decoder [24]. This traffic of MMS is partitioned into 40 distinct tasks. These communication tasks are mapped and scheduled on 25 IPs in a 5-by-5 mesh-based NoC system. Under this traffic pattern, the saturation throughput of *PDA-FTR* improves by 4.5-31 percent. On the other hand, the (1944, 972) LDPC is a widely used in most of the advanced wireless technology under 802.11n standard. Compared with other algorithms, the proposed *PDA-FTR* can improve 21.9-66.7 percent saturation throughput. These results show that *PDA-FTR* still performs better than the other routing schemes in the MMS and LDPC codes. Therefore, *PDA-FTR* also has a high potential to support this kind of multimedia systems with low average latency.

To quantify the network performance, the saturation throughputs of different routing algorithms under various traffic scenarios are summarized in Fig. 8. Since *Modified X-First* (*M. X-First*) has the worst performance among the related works, we use it as the baseline algorithm for comparison of saturation throughput normalization in Fig. 8. The results show that *PDA-FTR* acquires the highest overall saturation throughput with an average normalized saturation throughput of 1.75.

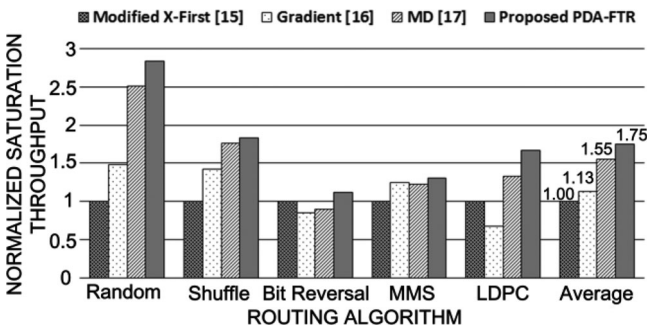

Fig. 8. Normalized performance improvement of different routing algorithms under various traffic scenarios.

5.2.2 Performance of PDA-FTR in Multiple-fault NoC

In addition to the single-fault network, we also evaluate the performance of *PDA-FTR* on a multiple-fault network. Because *Modified X-First* can only tolerate single fault [15], it is not evaluated in this section. As shown in Fig. 9, *PDA-FTR* outperforms the other two competitors by 23.3-26.9 percent for a system with two faulty routers under random traffic, 1.3-33.3 percent under shuffle traffic, 24.6-149.2 percent under bit-reversal traffic, 3.3-4.4 percent under MMS traffic, and 25-150 percent under LDPC traffic, respectively. In the cases of the system with four faulty routers as shown in Fig. 10, *PDA-FTR* improves by 9.1-14.3 percent under random traffic, 4.4-39.4 percent under shuffle traffic, 22.0-159.7 percent under bit-reversal traffic, 10.2-455.6 percent under MMS traffic, and 20-200 percent under LDPC traffic, respectively. The experimental results are in accordance with the claim that *PDA-FTR* has high fault-tolerance capability as it can maintain a slow degradation in performance as the number of fault increases.

5.3 Evaluation of Fault-Tolerance Ability

We adopt unreachable packet ratio to evaluate the fault-tolerance ability. This metric indicates the ratio of packets which is blocked by faults and cannot be successfully transmitted to its destination. It is expressed as follows:

$$Unreachable\ Packet\ Ratio = \frac{No.\ of\ unreachable\ packets}{No.\ of\ total\ packets}. \quad (17)$$

We simulate the system having one, two, and four faulty routers (as shown in Fig. 11), and then compare their unreachable packet ratio. From Table 2, we conclude that *PDA-FTR* has better fault-tolerance capability with successfully transmitted packet ratio above 98 percent than other related works even when there are four faulty routers in the network.

5.4 Statistical Traffic Load Distribution (STLD)

To evaluate traffic-balancing ability of the network, we employ statistical traffic load distribution. *STLD* [25] shows the result of sending the same amount of packets into the

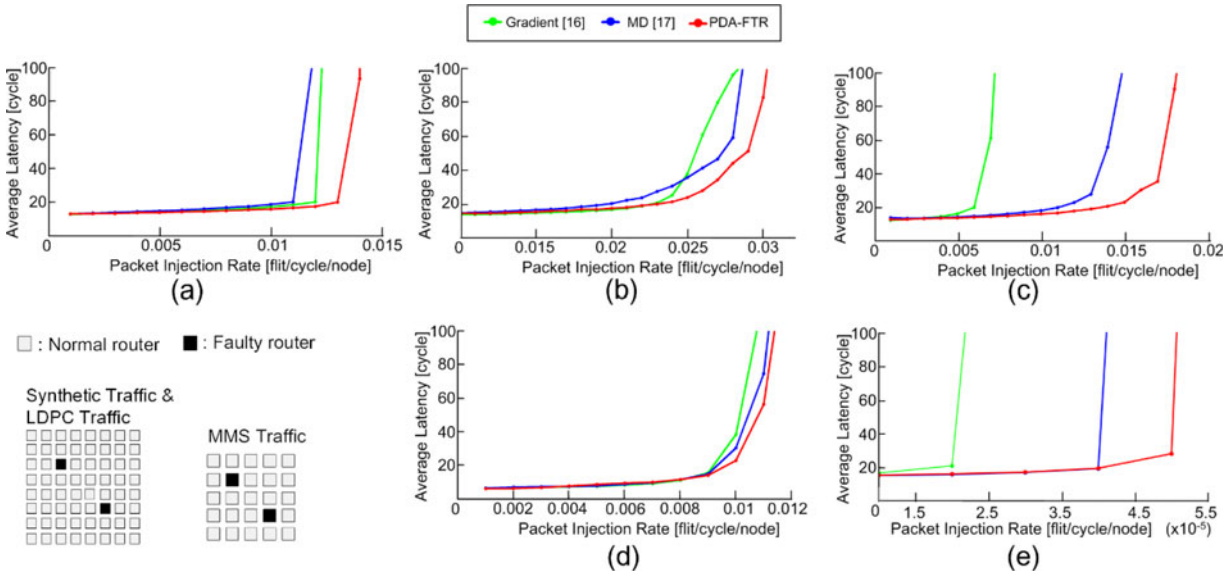

Fig. 9. The average latency of *Gradient*, *MD* and *PDA-FTR* under (a) random traffic, (b) shuffle traffic, (c) bit-reversal traffic, (d) MMS traffic, and (e) LDPC traffic where the network has two faulty routers.

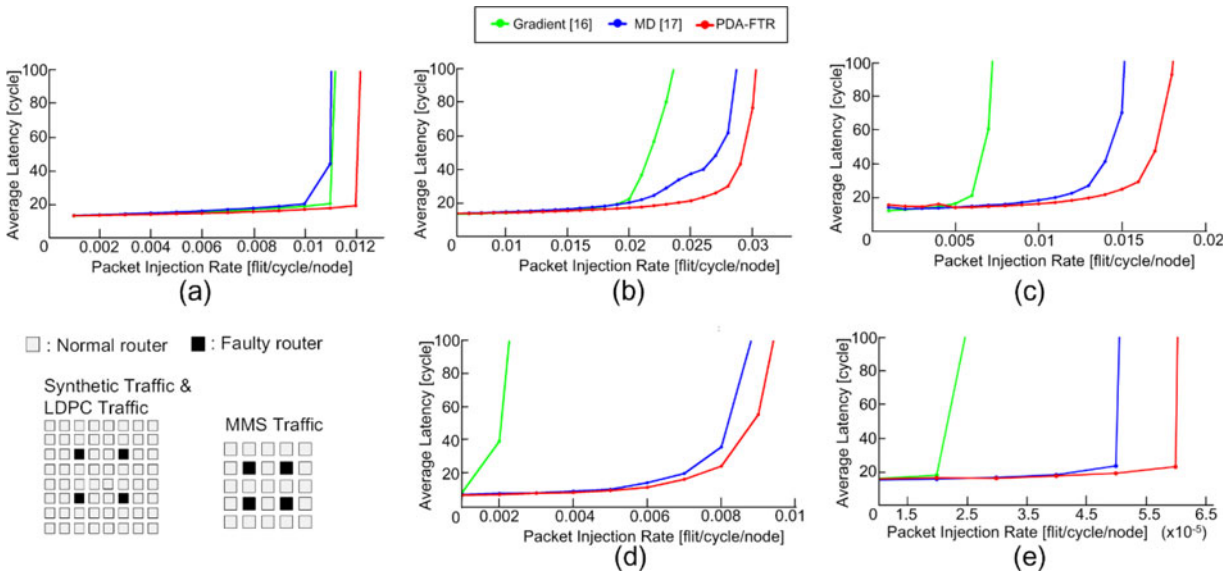

Fig. 10. The average latency of Gradient, MD and PDA-FTR under (a) random traffic, (b) shuffle traffic, (c) bit-reversal traffic, (d) MMS traffic, and (e) LDPC traffic where the network has four faulty routers.

network at the packet injection rate of saturation throughput. In this work, we inject a total of 1,000,000 flits under random traffic into the networks with one, two and four faulty routers as the design examples. *MD* is adopted as the competitive algorithm because it has better performance among the related works. The resulting *STLD* is shown in Fig. 11 and the color bar quantifies the range of the statistical traffic loads. The higher number of routed flits implies the heavier the traffic load that a router is bearing. Systems possessing better traffic-balancing ability can distribute packets more evenly to all routers in order to avoid regional congestion and have lower standard deviation. Fig. 11 illustrates that *PDA-FTR* has fewer congested routers than *MD*. Furthermore, Table 3 shows the quantitative results where *PDA-FTR* has lower standard deviation among all environment with an average of 38.1 percent improvement compared with *MD*.

5.5 Analysis on Performance Scalability

We evaluate performance scalability using network throughput, which is defined as the accepted traffic of the network at a given latency. The precise definition of network throughput is presented below:

$$Network\ Throughput = Saturation\ Throughput \times No.\ of\ nodes. \tag{18}$$

In this experiment, we measure the performance at the saturation latency in single-fault systems under various topology sizes. *MD*, the strongest candidate among the competitive algorithms in [15], [16], [17], is adopted for comparison. Fig. 12 demonstrates that network throughput of *PDA-FTR* scales with topology size. Under 18-by-18 mesh NoC, the network throughput of *PDA-FTR* is better than *MD* by 1.22 times. This experiment illustrates that

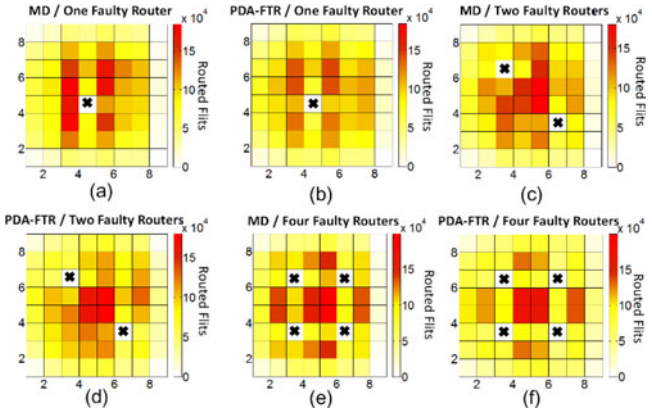

Fig. 11. The statistical traffic load distribution of MD in (a), (c), (e) and of *PDA-FTR* in (b), (d), (f) under network with one, two and four faulty routers, respectively.

TABLE 2
Unreachable Packet Ratio

Routing Algorithms	Numbers of Faulty Routers		
	1	2	4
Modified X-First [15]	2.36%	-	-
Gradient [16]	1.91%	2.40%	4.30%
MD [17]	0.34%	2.10%	2.40%
Proposed *PDA-FTR*	0.04%	0.20%	1.20%

TABLE 3
Standard Deviation of Traffic Load

Number of Faults	Routing Algorithm	
	MD [17]	**Proposed *PDA-FTR***
1 Faulty router	15,124	8,232 **(−45.6%)**
2 Faulty routers	14,345	9,724 **(−32.2%)**
3 Faulty routers	13,285	8,423 **(−37.0%)**
Average	14,251	8,793 **(−38.1%)**

- Unit: flits

PDA-FTR is a scalable routing algorithm in which it is suitable for both small and large mesh-based NoC systems.

In summary, our proposed *PDA-FTR* integrates fault information into *PD* (i.e., *FPD*) and concurrently employs local buffer information (i.e., *BO*) when selecting the routing path. Therefore, *PDA-FTR* has higher saturation throughput, greater fault-tolerance ability, better traffic-balancing ability and higher performance scalability than the previous works in the faulty NoC.

6 Architecture Design of Proposed *PDA-FTR* Algorithm

6.1 *PDA-FTR* Router Architecture

The router architecture for a mesh-based network topology implementing *PDA-FTR* is shown in Fig. 13a. Each router contains an *FPD* table, a set of first-in-first-out (FIFO) input buffer, an adaptive routing algorithm, a matrix allocator and a crossbar switch circuit. The adaptive routing algorithm contains routing function and selection function, whose architecture are shown in Fig. 13b and 13c, respectively.

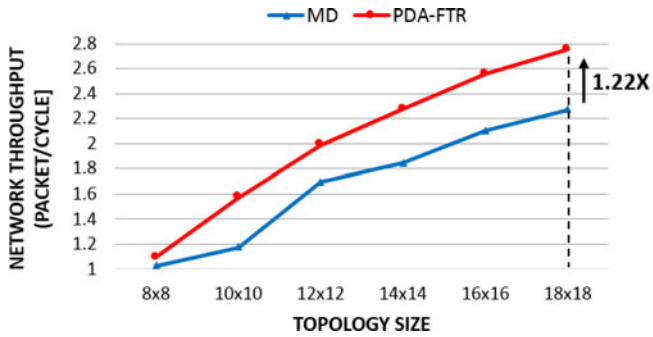

Fig. 12. Performance scalability of PDA-FTR under various topology size.

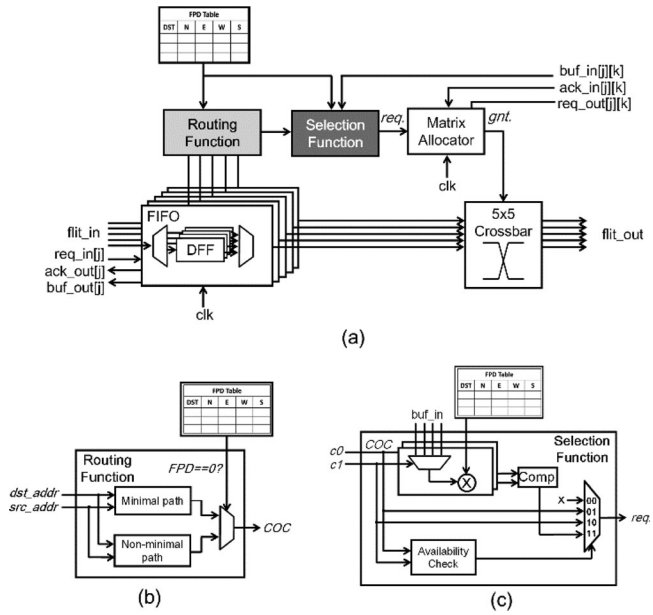

Fig. 13. The architecture of (a) the proposed *PDA-FTR* router, (b) the routing function of *PDA-FTR* algorithm, and (c) selection function of *PDA-FTR* algorithm.

In the routing function of *PDA-FTR*, there are two types of routing modes: 1) minimal routing path and 2) non-minimal routing path. The inputs of this module are source address, destination address, and the *FPD* value. Compared with the baseline router architecture, an additional routing unit and a multiplexer are added for determining the minimal or non-minimal routing paths based on the input addresses. After executing the routing function, the candidate output channels will be provided to the selection function.

The selection function will choose the best output channel among the given *COCs*. For each *COC*, the corresponding buffer information (*buf_in*) and *FPD* value are multiplied to obtain *EBL* information. Then, the comparator selects the *COC* with higher *EBL*. Meanwhile, the availability check block checks the reservation table whether the *COC* is being reserved by other packets. If the *COC* is being reserved, it will not be chosen as the output channel even if it has higher *EBL* value.

6.2 Cost Reduction on FPD Table

To implement *PDA-FTR*, we need to store *FPD* table in every router. However, storing the entire *FPD* table in a router seems to be an unpractical way. The number of

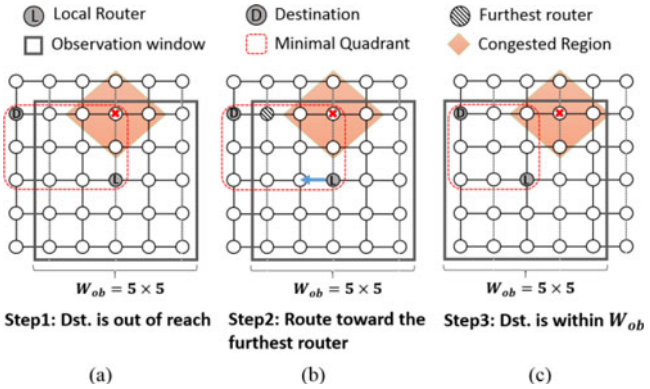

Fig. 14. Illustration of *PDA-FTR* using regional *FPD* table: (a) Step 1: check whether the destination is reachable with the regional table; (b) Step 2: if no, forward to the furthest router of the same direction; (c) Step 3: if yes, route directly toward the destination.

entries in *FPD* table is $(k^2 - 1) \times 4$, where k represents topology size and four is the number of directions (i.e., North, South, East, and West). Because NoCs have limited resources, the central problem of *PD*-based routing is that the table entries increase significantly as the scale of a $k \times k$ mesh NoC increases. Thus, two problems arise.

- **Scalability problem:** The number of table entries in a router is increased by $O(k^2)$. Therefore, the memory cost increases markedly faster than k.
- **Increase in time and power for table access:** Along with the scalability problem, computing timing cost and power consumption grow significantly when searching for a certain entry in a large table.

These problems result in high design cost and adversely affect network performance. Therefore, *FPD* table cost reduction is necessary for achieving a cost-efficient fault-tolerant adaptive routing scheme.

To reduce the cost of memory, we propose regional *FPD* table to cover most routing paths for packet transmission while minimizing performance degradation. Due to the data locality, processing elements that often communicate with each other are usually placed in proximity to each other for minimizing the delay in data delivery. Based on this property of NoC, a suitable size of the table can help reserve the *PD* information of popular paths for making a routing decision with very little performance degradation when compared to full *FPD* table. Therefore, the original *FPD* table can be scaled down by retaining only the popular source-destination pairs, as shown in Fig. 14. The size of the observation window (W_{ob}) is defined as hop counts on the side of the square.

Because of the involved non-minimal odd-even turn model, the routing turn is restricted on the odd column and even column. Therefore, we can only record the regional information within a W_{ob}. On the other hand, to make each local router has balanced *FPD* information toward the four directions (i.e., North, South, East, and West), as shown in Fig. 14. Consequently, with 5-by-5 regional *FPD* table, *PDA-FTR* obtains sufficient information to route packet toward both odd column and even column from the local router. For a k-by-k mesh-based NoC system, the number of table entries is expressed as

TABLE 4
Router Area and Total Power Consumption with Different Routing Algorithms @ 400 MHz

	Routing Algorithm			
	Modified X-First [15]	*Gradient [16]*	*MD [17]*	Proposed *PDA-FTR*
FIFO	43,406	45,922	45,836	45,961
Routing Unit	1,654	2,760	3,334	3,174
Crossbar	3,313	3,313	3,313	3,313
Arbiter	1,078	1,188	1,188	1,188
Routing Table	0	0	0	1,143
Total Area (um^2)	49,451	53,183	53,671	54,779
Total Power (mW)	13.87	13.98	14.04	14.85

$$Table\ entries = \begin{cases} (k^2 - 1) \times 4, \text{if } k < 5 \\ (5^2 - 1) \times 4, \text{if } k \geq 5 \end{cases}, \tag{19}$$

where k is the topology size.

Because some information for distant destinations are not kept in the regional table, one cannot directly obtain *FPD* information from W_{ob} of the local router. Fig. 14a illustrate an example. Because the location of the destination is out of the observing window, the routing algorithm may not discover a routing path in one step for the destinations in local router. In such case, *PDA-FTR* checks its regional *FPD* table in every routing step. If the destination is out-of-reach, *PDA-FTR* employs *FPD* value of the furthest router in the same direction and route the packet toward this furthest router, as illustrated in Fig. 14b. Ultimately, as shown in Fig. 14c, the destination will be reachable using the *FPD* table of the local router. With the above consideration, we only need to store the partial *FPD* information in the router. The complexity of *FPD* table is reduced from $O(k^2)$ to a constant cost of $O(1)$ for each router, while the k is greater than five. Obviously, this constant cost does not relate to the network size. Therefore, the proposed regional *FPD* table still has better scalability in large-scale NoC size.

6.3 Analysis of Hardware Overhead

Table 4 and Fig. 15 report the summary on breakdown of router area. The router is extended from the generic router architecture [27] and synthesized with TSMC 90nm CMOS process at 400 MHz without pipeline. The result shows that the router with *PDA-FTR* has an area overhead of 10.77 percent compared to the baseline design, *Modified X-First* [15]. It is worth noting that regional *FPD* table only accounts for

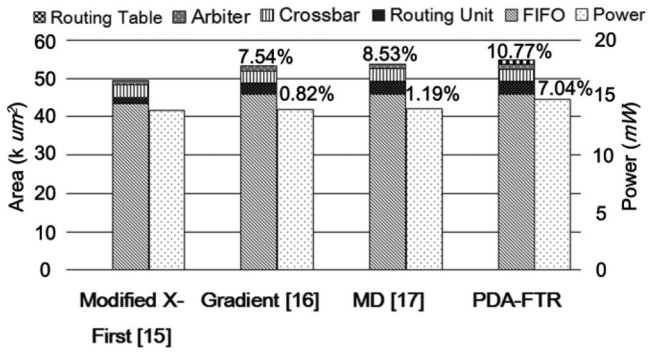

Fig. 15. The breakdown of router area and total power consumption of a router with respect to different routing algorithms.

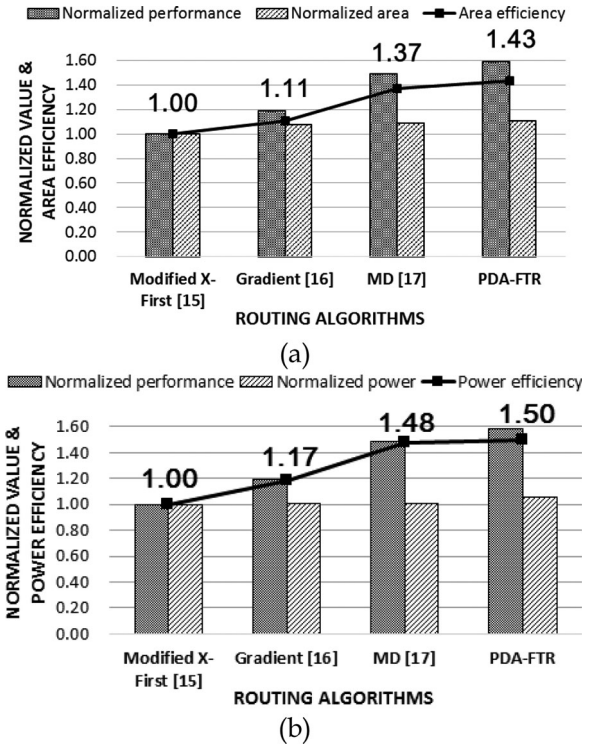

Fig. 16. The comparison of (a) area efficiency and (b) power efficiency of different routing algorithms.

2 percent of *PDA-FTR* router area. On the other hand, compared with *Modified X-First* [15], *Gradient* [16] and *MD* [17] have an area overhead of 7.54 and 8.53 percent respectively, which are relatively lower than *PDA-FTR*.

In addition to the analysis of area overhead, the total power consumption of a router with respect to different routing algorithm are also shown in Table 4 and Fig. 15. Compared with the baseline *Modified X-First* [15] router, the proposed *PDA-FTR* router has power overhead of 7.04 percent because of the extra routing table. In the next section, we will show that although the implementation of *PDA-FTR* requires a larger area and power overhead, its area and power efficiency are better than other related works.

6.4 Evaluation of Hardware Efficiency

Adaptive routing algorithm improves network performance at the cost of hardware overhead. Thus, it is necessary to evaluate the effectiveness of each routing algorithm through area and power efficiency. Area and power efficiency quantifies the performance improvement that each percentage increased in implementation cost can gain, and they can be defined as

$$Area\ Efficiency = \frac{Staturation\ Throughput\ (flit/cycle)}{Area\ Overhead\ (\mu m^2)} \tag{20}$$

and

$$\text{Power}Efficiency = \frac{Staturation\ Throughput\ (flit/cycle)}{\text{Power}Overhead\ (mW)} \tag{21}$$

Fig. 16 demonstrates the comparisons of area and power efficiency of *PDA-FTR*, which are derived from normalized performance and the normalized hardware overhead. In this case, we adopt *Modified X-First* [15] as the baseline algorithm for comparison. *Gradient* [16], *MD* [17] and *PDA-FTR* all acquire area efficiency that is greater than 1, which implies that the increment in hardware cost brings along a greater scale in performance improvement. In particular, *PDA-FTR* has an area efficiency of 1.43 and power efficiency of 1.50 which are the highest among all and thus proven its effectiveness and cost-efficiency as a fault-tolerant routing algorithm.

7 Conclusion

This paper introduces the concept of *FPD* and proposes the *PDA-FTR* algorithm to achieve fault-resilient packet delivery and to balance traffic load. Based on the *FPD* information, we integrate it with local buffer information for path selection. Moreover, we compare our work with other related works in term of saturation throughput, fault-tolerance ability, traffic-balancing ability and performance scalability. The experiments show that overall *PDA-FTR* achieves the better performance in a faulty NoC. The designed router architecture justifies that our proposed work is cost-efficient.

Acknowledgments

This work was supported by the National Science Council of Taiwan under Grant MOST 102-2220-E-002-001, MOST 104-2220-E-002-003, and MOST 104-2218-E-110-011-MY2.

References

[1] L. Benini and G. DeMicheli, "Networks on chip: A new SoC paradigm," *IEEE Comput.*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
[2] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2001, pp. 684–689.
[3] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L.-S. Peh, "Research Challenges for On-Chip Interconnection Networks," *IEEE Micro*, vol. 27, no. 5, pp. 96–108, Sep./Oct. 2007.
[4] S. Furber, "Living with failure: Lessons from nature?" in *Proc. Eur. Test Symp.*, May 2006, pp. 4–8.
[5] A. Singh, M. Shafique, A. Kumar, and J. Henkel, "Mapping on multi/many-core systems: Survey of current and emerging trends," in *Proc. ACM/IEEE Des. Autom. Conf.*, May 2013, pp. 1–10.
[6] G.-M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 7, pp. 729–738, Jul. 2000.
[7] J. Hu and R. Marculescu, "DyAD—Smart routing for networks-on-chip," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2004, pp. 260–263.
[8] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," in *Proc. Des., Autom. Test Eur.*, Mar. 2003, pp. 1126–1127.
[9] G. Ascia, V. Catania, and M. Palesi, "Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip," *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 809–820, Jun. 2008.
[10] P. Gratz, B. Grot, and S. W. Keckler, "Regional congestion awareness for load balance in network-on-chip," in *Proc. IEEE Int. Symp. High-Performance Comput. Archit.*, Feb. 2008, pp. 203–214.
[11] E.-J. Chang, H.-K. Hsin, S.-Y. Lin, and A.-Y. Wu, "Path-congestion-aware adaptive routing with a contention prediction scheme for network-on-chip systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 33, no. 1, pp. 113–126, Jan. 2014.
[12] Y.-H. Kuo, P.-A. Tsai, H.-P. Ho, E.-J. Chang, H.-K. Hsin, and A.-Y. Wu, "Path-diversity-aware adaptive routing in network-on-chip systems," in *Proc. IEEE Int. Symp. Embedded Multicore SoCs*, Sep. 2012, pp. 175–182.
[13] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based Ad Hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, Jun. 2006.

[14] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Fault tolerant algorithms for network-on-chip interconnect," in *Proc. IEEE Symp. VLSI*, Feb. 2004, pp. 46–51.
[15] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2D-mesh network-on-chip," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2008, pp. 441–446.
[16] I. Pratomo and S. Pillement, "Gradient – an adaptive fault-tolerant routing algorithm for 2D mesh network-on-chips," in *Proc. Des. Archit. Signal Image Process.*, Oct. 2012, pp. 1–8.
[17] M. Ebrahimi, M. Daneshtalab, J. Plosila, and F. Mehdipour, "MD: Minimal path-based fault-tolerant routing in on-chip networks," *Proc. 18th Asia South Pacific Des. Autom. Conf.*, Jan. 2013, pp. 35–40.
[18] P. Bahrebar and D. Stroobandt, "The Hamiltonian-based odd-even turn model for maximally adaptive routing in 2D mesh networks-on-chip," *Comput. Electr. Eng.*, vol. 45, pp. 386–401, Jul. 2015.
[19] S. Pasricha and Y. Zou, "NS-FTR: A fault tolerant routing scheme for networks on chip with permanent and runtime intermittent faults," in *Proc. 16th Asia South Pacific Des. Autom. Conf.*, Jan. 2011, pp. 443–448.
[20] A. Zinzuwadia, P. Parandkar, R. Verma, and S. Katiyal, "An efficient deadlock-free NARCO based fault tolerant routing algorithm in NoC architecture," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 2, pp. 227–234, Feb. 2012.
[21] K.-C. Chen, S.-Y. Lin, W.-C. Shen, and A.-Y. Wu, "A scalable built-in-self-recovery (BISR) VLSI architecture and design methodology for 2D-mesh based On-chip networks," *Des. Autom. Embedded Syst.*, vol. 15, no. 2, pp. 111–132, Jan. 2011.
[22] W. C. Tasi, K. C. Chu, Y. H. Hu, and S. J. Chen, "Non-minimal, turn-model based NoC routing," *Microprocessors Microsyst.*, vol. 37, pp. 899–914, 2013.
[23] SourceForge, Noxim: Network-on-Chip Simulator, 2008. [Online]. Available: http://sourceforge.net/projects/noxim
[24] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," in *Proc. ASP Des. Autom. Conf.*, Jan. 2003, pp. 233–239.
[25] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Burlington, MA, USA: Morgan Kaufmann, 2004.
[26] K.-C. Chen, S.-Y. Lin, W.-C. Shen, and A.-Y. Wu, "A scalable built-in self-recovery (BISR) VLSI architecture and design methodology for 2D-mesh based on-chip networks," *Des. Autom. Embedded Syst.*, vol. 15, pp. 111–132, Apr. 2011.
[27] J. Hu and R. Marculescu, "Application-specific buffer space allocation for networks-on-chip router design," in *Proc. IEEE/ACM Int. Conf. Comp. Aided Design*, Nov. 2004, pp. 354–361.
[28] IEEE 802 Standard [Online]. Available: www.ieee802.org/11

Yu-Yin Chen received the BS degree in electrical engineering from National Taiwan University in 2013. She is currently working toward the MS degree in Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan. Her research interests include fault tolerance algorithms and architectures in Network-on-Chip (NoC) systems. He is currently an engineer at MediaTek Inc., Hsinchu, Taiwan.

En-Jui Chang received the BS degree in electrical engineering from National Central University, Zhongli, Taiwan, in 2008, and the PhD degree from Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, in 2014. His research interests include Network-on-Chip (NoC) algorithms/architectures, bio-inspired algorithms/architectures, and fault tolerance algorithms/architectures in VLSI systems. He is currently an engineer at MediaTek Inc., Hsinchu, Taiwan.

Hsien-Kai Hsin received the BS degree in electrical engineering and the PhD degree from Graduate Institution of Electronic Engineering, National Taiwan University, Taiwan, in 2009 and 2014, respectively. His research interests include Biologically-inspired systems and high-speed interconnects. In particular, his research focuses on swarm intelligence design methodologies for multicore SoCs, with special interest in Network-on-Chip communication architecture. He is currently an engineer at MediaTek Inc., Hsinchu, Taiwan.

Kun-Chih (Jimmy) Chen (IEEE S'10-M'14) received the BS degree in computer science and engineering from National Taiwan Ocean University, Taiwan, in 2007, the MS degree in computer science and engineering from National Sun Yat-sen University, Taiwan, in 2009, and the PhD degree from Graduate Institute of Electronics Engineering, Nation Taiwan University, Taiwan, in 2013. From October 2014 to January 2015, he served as a postdoctoral fellow in Intel-NTU Connected Context Computing Center working on the development of Green Sensing Platform for Internet of Things (IoTs), Reliable Thermoelectric Converter, and Power-aware Software Defined Network (SDN). From February 2015 to July 2016, he joined the faculty of Electronic Engineering Department of Feng Chia University. He is currently an assistance professor of Computer Science and Engineering Department of National Sun Yat-Sen University. His research interests include algorithm development, VLSI architecture design, and implementation for three-dimensional Networks-on-Chip (3D NoC), advanced arithmetic unit design, and fault-tolerant system designs. He served on the technical program committees of some major IEEE International Conferences, such as ICCE-TW and SiPS. He received the Best Paper Award of 2014 International Symposium on VLSI Design, Automation and Test (VLSI-DAT'14), the Best Paper Award of 2016 International Joint Conference on Convergence (IJCC'16), and the Best PhD Dissertation Award of IEEE Taipei Section in 2014. Besides, he was also invited to publish a book chapter in "Routing Algorithm in Network-on-Chip," which was published by Springer in November 2013. He is a member of the IEEE.

An-Yeu (Andy) Wu (IEEE M'96-SM'12-F'15) received the BS degree from National Taiwan University in 1987, the MS and PhD degrees from the University of Maryland, College Park, in 1992 and 1995, respectively, all in Electrical Engineering. In 2000, he joined the faculty of the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), where he is currently a professor. His research interests include low-power/ high-performance VLSI architectures for DSP and communication applications, adaptive/multi-rate signal processing, reconfigurable broadband access systems and architectures, and System-on-Chip (SoC)/ Network-on-Chip (NoC) platform for software/hardware co-design. He had served as the associate editors of leading IEEE Transactions in the circuits and systems area and signal processing area, such as the *IEEE Transactions on Vert Large Scale Integration (VLSI) Systems*, the *IEEE Transactions on Circuits and Systems I: Regular Papers*, the *IEEE Transactions on Circuits and Systems II: Express Briefs*, and the *IEEE Transactions on Signal Processing*. He is now serving an associate editor for the *Journal of Signal Processing Systems (JSPS)* and acted as the lead guest editor of the Special Issue of 2010 the *IEEE Workshop on Signal Processing Systems (SiPS)* in JSPS, which was published in Nov. 2011. He also served on the technical program committees of many major IEEE International Conferences, such as SiPS, AP-ASIC, ISCAS, ISPACS, ICME, SOCC, and A-SSCC. He is currently serving as the chair of VLSI Systems and Architectures (VSA) Technical Committee in IEEE Circuits and Systems (CAS) Society. In 2015 he became a fellow member of the IEEE.