Traffic-aware Reconfigurable Architecture for Fault-tolerant 2D Mesh NoCs

Poona Bahrebar
Department of Electronics and Information Systems (ELIS), Ghent University, Belgium
poona.bahrebar@ugent.be

Dirk Stroobandt
Department of Electronics and Information Systems (ELIS), Ghent University, Belgium
dirk.stroobandt@ugent.be

ABSTRACT

With the aggressive scaling of the VLSI technology, Networks-on-Chip (NoCs) are becoming more susceptible to faults. Therefore, designing reliable and efficient NoCs is of significant importance. The rerouting approach which is employed in most of the fault-tolerant methods causes the network performance to degrade considerably due to taking longer paths and creating hotspots around the faults. Moreover, they cannot adapt to the dynamic traffic distribution in the network. Considering the increasing demands for real-time systems, the necessity for designing reconfigurable and robust NoCs is even more pronounced. In this paper, a dynamically reconfigurable technique is proposed to address fault-tolerance and minimal routing in mesh NoCs. To accomplish this goal, the router architecture is modified to enable the frequently communicating nodes to bypass the faulty router and communicate through shorter paths. Thus, not only the rerouting is minimized, the connectivity of the network is maintained in the vicinity of faults. The experimental results validate the performance and reliability of the proposed technique with a small hardware overhead.

CCS Concepts

•Computer systems organization → Interconnection architectures; •Networks → Physical topologies; Network reliability; •Hardware → Network on chip;

Keywords

Network-on-Chip (NoC), fault-tolerant routing methods, minimal paths, dynamic reconfiguration, deadlock

1. INTRODUCTION

With the aggressive device scaling in the semiconductor technology, two salient problems emerge: (1) efficient connection of the increasing number of on-chip resources, and (2) effective management of the decreasing reliability [1]. The first problem concerns the limitation of the conventional interconnects which cannot scale well with the complexity of the chip multiprocessors. The *Network-on-Chip (NoC)* paradigm has emerged as a promising solution for such platforms [2]. The second problem arises from the extreme device scaling which threatens the reliability by low yield and wear-out. Thus, the interconnect must be able to tolerate partial failure due to its critical role in holding all of the components of the system together [1, 3, 4].

The failures are generally classified as either *permanent* or *transient*. Moreover, failures can occur in the switches[1] or links connecting the switches. Permanent switch failures are addressed in this paper.

The incorporation of fault-tolerance into NoC design mainly revolves around two intertwined concepts: (1) *routing method*, and (2) *network architecture*. While the focus is often slanted toward the routing methods, the network architecture plays an equally important role in delivering a reliable system.

Realizing the significance of network connectivity in the presence of faulty routers, the current research was carried out to propose a **T**raffic-**A**ware **R**econfigurable **A**rchitecture, aptly called *TARA*, for Fault-tolerant 2D mesh NoCs. The underlying principle of the proposed technique is inspired by the Minimal-path Connection-retaining Fault-tolerant (MiCoF) approach [4] to maintain the connectivity of a network by modifying the router architecture. Once a router becomes faulty, its associated links are normally discarded. However, by connecting the incoming links of a faulty router to its outgoing links in the desired directions, the packets are able to bypass the faulty switch and reach the destination through surviving routers. The main features and contributions of the proposed technique are:

(1) Unlike the static MiCoF which is oblivious to the traffic distribution, the network topology in TARA is dynamically tailored according to the location of faults and the current traffic pattern in the network. This is especially important for the real-time systems since their dynamically changing requirements can be met effectively and a high performance communication can be achieved. Moreover, the corner paths can be survived using TARA.

(2) The proposed method supports *adaptivity* (path diversity) in the network. Furthermore, rerouting is minimized by sending the packets through the available *minimal* (shortest) paths. It does not require routing tables, disable healthy routers, or set exceptional rules for the borderlines.

(3) TARA uses a purely local *fault look-ahead information* which is the minimum knowledge required for making reliable routing decisions. Thus, the hardware overhead is kept to a minimum.

(4) Only one and two Virtual Channels (VCs) [2] along the X and Y dimensions are required to ensure deadlock-freedom for a single faulty router. It is livelock-free, as well.

The rest of the paper is organized as follows. The related works are studied in Section 2 followed by a description of MiCoF in Section 3. Section 4 details the mechanism of TARA. Section 5 is devoted to the simulation results and discussion. Finally, the conclusions are drawn in Section 6.

[1] "switch" and "router" are used interchangeably in this paper.

2. RELATED WORK

Most of the existing fault-tolerant routing methods rely on detour strategies in order to *reroute* the packets around the faulty region [4]. However, rerouting may introduce *deadlock* or *livelock* which can paralyze the network operations. The methods using detour can be divided into three main groups, depending on the mechanism they use to create a deadlock-free path around the fault: some approaches [1, 6] opt for a more restricted set of path selection heuristics such as the *turn model* [2], some [3, 7] use VCs, and some [1, 7, 8, 9] require off-line reconfiguration of routing tables. Regardless of the mechanism being used, the detour-based algorithms suffer from two major drawbacks: (1) The hotspots are more likely to be created around the faults when rerouting is performed. Furthermore, the aggregated congestion by taking longer paths degrades the performance, significantly [4]. (2) Strict restrictions on the number and location of faults do not permit arbitrary fault patterns within the network.

Besides routing algorithms, several studies [8, 4, 3] look into router design to improve the reliability through microarchitectural modifications. For instance, the baseline router is extended in [4, 3] to *bypass* the faulty components within the routers. The mechanism introduced in [3] connects the input and output router ports directly by adding redundant wires, called Default Backup Paths (DBPs), to maintain the connectivity of the network in the vicinity of faults. However, it is difficult to design a deadlock-free routing algorithm for the resulting irregular topology. The MiCoF approach [4] combats failures similarly by allocating alternate paths to bypass the faulty router. The nominal hardware of DBP and MiCoF is much simpler compared to that of the original router datapath. Thus, the packets are transmitted with shorter latency and lower energy. However, both techniques are statically hardwired and do not take the traffic scenarios into account.

Reconfiguration techniques have also been widely used [9, 8] to design resilient NoCs. Most of the fault-tolerant algorithms studied earlier perform route reconfiguration to bypass faults. In general, reconfiguration solutions require some extra logic to enable alternative paths. Besides, it is necessary to avoid the reconfiguration-induced deadlocks that may occur as a result of transitioning from an old routing algorithm to a new one while the network is still being reconfigured [8].

Considering the literature, it is likely that a combination of different techniques may be required to address fault-tolerance, given the various fault situations that may occur in a network.

3. PRELIMINARIES

In order to reduce rerouting and maintain a connected mesh, Ebrahimi et al. [4] modified the router architecture to function as a bridge by coupling the horizontal and vertical adjacent routers. As shown in Fig. 1, once the MiCoF router fails, the East (respectively, West) input channel is directly connected to the West (resp. East) output channel. Similarly, the North (resp. South) input is directly linked to the South (resp. North) output channel [4].

Figure 2(a) presents a 4×4 mesh with five faulty routers. The resulting network after applying the MiCoF approach is illustrated in Fig. 2(b). As seen in the figure, the network connectivity is restored by replacing the broken datapaths of routers 6 and 9 with two bypassing paths. The faulty routers on the border of the network (1 and 11) can serve merely as horizontal or vertical bridges. The failed corner router 12 is completely discarded as it cannot assist to the network connectivity according to the MiCoF architecture. As an example, consider the NE-ward packet sent from source node 4 to the destination node 10. The packet can bypass the faulty router and reach the destination through a minimal path (2 hops, shown with the red arrow) without rerouting.

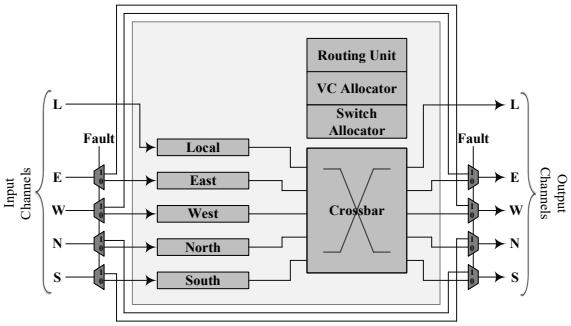

Figure 1: The MiCoF router architecture [4].

4. PROPOSED TECHNIQUE

According to [10], the topology determines the ideal performance of the NoC whereas the routing algorithm determines how much of this potential could be realized. As seen previously, the only concern in MiCoF is to restore the network connectivity oblivious to the existing traffic pattern. As a result, the bypassing paths in MiCoF are static and cannot be changed at run-time. Thus, while some traffic scenarios may enjoy the routes provided by MiCoF, the resulting topology may put a heavy burden on some other cores. This can be better explained through an example. Consider the network in Fig. 2(b) where router 6 is bypassed through a direct route between 5 and 7. However, this particular link may be of little benefit if there exists no significant communication transaction between 5 and 7. Assume that node 5 is expected to send packets to 10 for a long interval. The current topology implies rerouting the packets through a longer (at least 3-hops) path to reach 10. However, it is of great importance to meet the applications' requirements and avoid congestion in a faulty network. To accomplish this goal, we inherit the core idea of MiCoF and equip that with reconfiguration logic to provide on-demand minimal (and

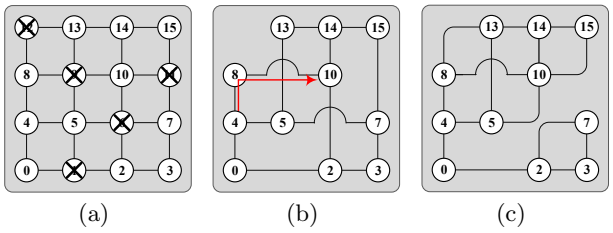

Figure 2: A 4×4 mesh network (a) with five faulty routers [4], (b) using the MiCoF approach [4], and (c) using the proposed TARA technique.

adaptive) routes that match the communication pattern of the currently running application.

4.1 The Router Architecture of TARA

Figure 3(a) illustrates a faulty router and its four adjacent neighbors. Our aim is to enable the possibility of reconfiguration such that not only the network connectivity is sustained, but also the deficient router can be adjusted to reduce the the hop count between the high-volume communicating nodes. As depicted in the figure, the broken router can be bypassed in three different modes: *Mode 1* which is similar to MiCoF provides straight connections. *Mode 2* implies connecting the West neighbor to the South, and the East neighbor to the North. The connected pairs in *Mode 3* are East-South and West-North.

The proposed router architecture is depicted in Fig. 3(b). As can be seen, we modified the MiCoF switch (Fig. 1) by adding a Reconfiguration Control Unit (RCU), one 4-to-2 encoder, and larger demultiplexers and multiplexers at the input and output ports plus the required wiring. Similar to MiCoF, the routers employ a Built-In Self-Test (BIST) mechanism to evaluate their internal components. Once a failure is detected in a router, the *Fault* signal is activated.

The functionality of the TARA switch is explained as follows. Consider the same example discussed earlier with the initial state as Fig. 2(a) where a heavy communication flow is created from node 5 to 10. However, switch 6 is broken and needs to be bypassed according to *Mode 3*. Thus, node 5 makes a complaint to the RCU of node 6 about the unavailable EN route whenever it tries to transmit packets. The RCU of node 6 collects all of the complaints from its neighbors, evaluates the requirements, and eventually determines which mode is suitable for the current network state. The RCU has one output port for each bypassing mode (labeled as $M1$, $M2$, and $M3$). Depending on the mode chosen by RCU, only one output signal has the value of 1, while the rest are 0. Together with $\overline{Fault}$, the RCU output signals are connected to the inputs of the encoder. The truth table of the encoder is also illustrated in Fig. 3(b). As depicted in the figure, two bits are required to configure the TARA router to operate in four different modes (i.e. original datapath, $M1$, $M2$, and $M3$). Thus, the outputs of the encoder are connected to the select lines of the demultiplexers and multiplexers allowing the desired physical route to be established. For instance, if *Mode 3* is eventually selected by RCU, a new bitstream ("11") is generated at the output of the encoder such that the East input port will be directed to the North, West to the South, North to the East, and South to the West.

Once a failure occurs, the proposed reconfigurable architecture adjusts the affected paths dynamically to reduce network blocking while restoring the connectivity. Therefore, the network may have different topologies at different times depending on the communication patterns. For instance, the network in Fig. 2(c) fulfills heavy communication demands between the 5-13, 5-10, 0-2, 2-7, and 10-15 pairs. Moreover, unlike MiCoF, TARA can always rescue the corner routes.

Note that bypassing multiple adjacent faulty routers may create long links that may decrease the clock frequency since the flits cannot pass through them during a single cycle. To solve this problem, we can borrow the idea of [11] where registers (1-flit buffers) are inserted along the alternative links such that the flits can be latched after each cycle.

4.2 The Routing Algorithm of TARA

In order to avoid traversing unnecessary longer paths, each router needs to know about the fault status of its surrounding neighbors (i.e. fault information). In TARA, we assume that each router is equipped with a 1-hop look-ahead fault information, similar to MiCoF.

Fault scenarios degrade the regular NoCs to irregular (arbitrary) topologies. Thus, most of the designers resort to topology-agnostic routing algorithms such as Up*/Down* which rely on lookup tables to ensure the deadlock-freedom. These schemes suffer from two major drawbacks [12]: (1) The cost of the lookup tables (in terms of silicon area and power consumption) badly affects the scalability of the algorithm; (2) Heavy computation is required to maintain the correct

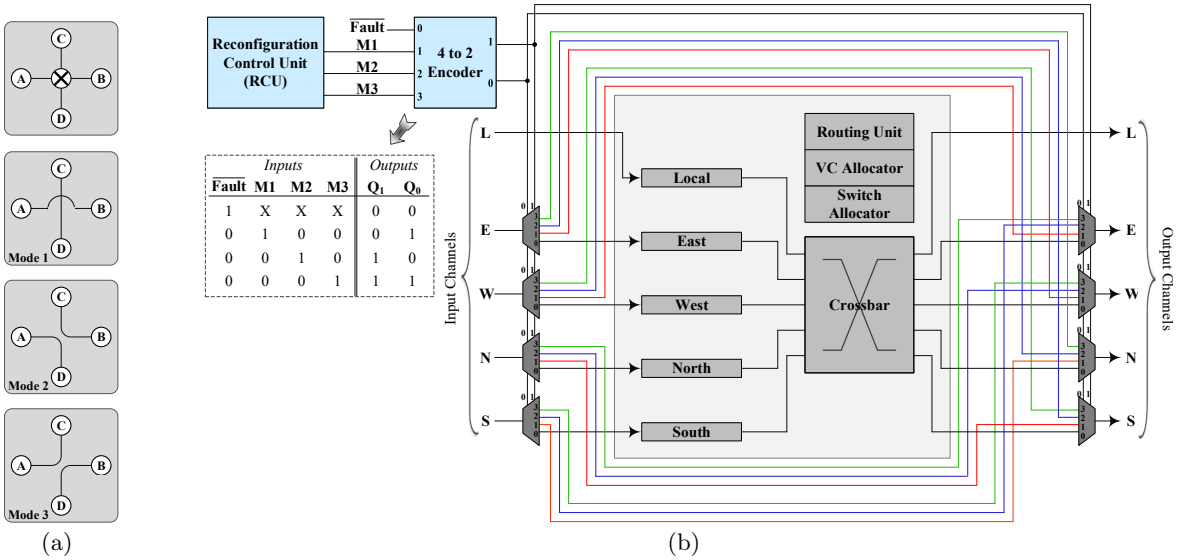

Figure 3: (a) Bypassing a faulty router in three different modes. (b) The proposed router architecture.

routing paths in the network.

In order to avoid costly routing tables, the TARA routing algorithm is designed upon the Dynamic XY (DyXY) routing algorithm [13] where the packet is adaptively sent to the X or Y direction depending on the congestion condition. In order to ensure deadlock-freedom, one and two VCs are utilized along the X and Y dimensions, respectively. Further details about DyXY can be found in [4].

TARA is capable of tolerating any single faulty switch with 100% reliability while ensuring the deadlock-freedom. Figures 4(a) and (b) illustrate the cases where a northeast-ward packet detects a broken switch using the look-ahead information and the destination node, D, is one hop away from the current node, C, along both X and Y dimensions. As discussed previously, the faulty switch may be configured in three different modes which are illustrated under the original state. If the switch operates in *Mode 1* or *Mode 2*, using the bypassing routes implies a non-minimal path to the destination. Thus, the packet takes the 2-hop NE path (Fig. 4(a)) or the EN path (Fig. 4(b)) to reach the destination. In *Mode 3*, the packet simply takes the direct minimal path to reach the destination. To ensure the consistency with the DyXY routing algorithm, bypassing the paths formed by *Mode 2* or *Mode 3* in TARA is treated as taking turns from X to Y dimension and vice versa. Note that a *turn* involves a 90-degree change of the traveling direction.

Figures 4(c)-(j) show all of the cases for a northeast-ward packet where the distance between C and D is one or two hops along the X and Y dimensions. Depending on the bypassing mode, the packets are forwarded through different paths as depicted in the figures. Finally, Fig. 4(k) and (l) show the routing policy of TARA for an east- and northward packet, respectively. As can be seen, TARA prefers to transmit the packets through the minimal paths. More-

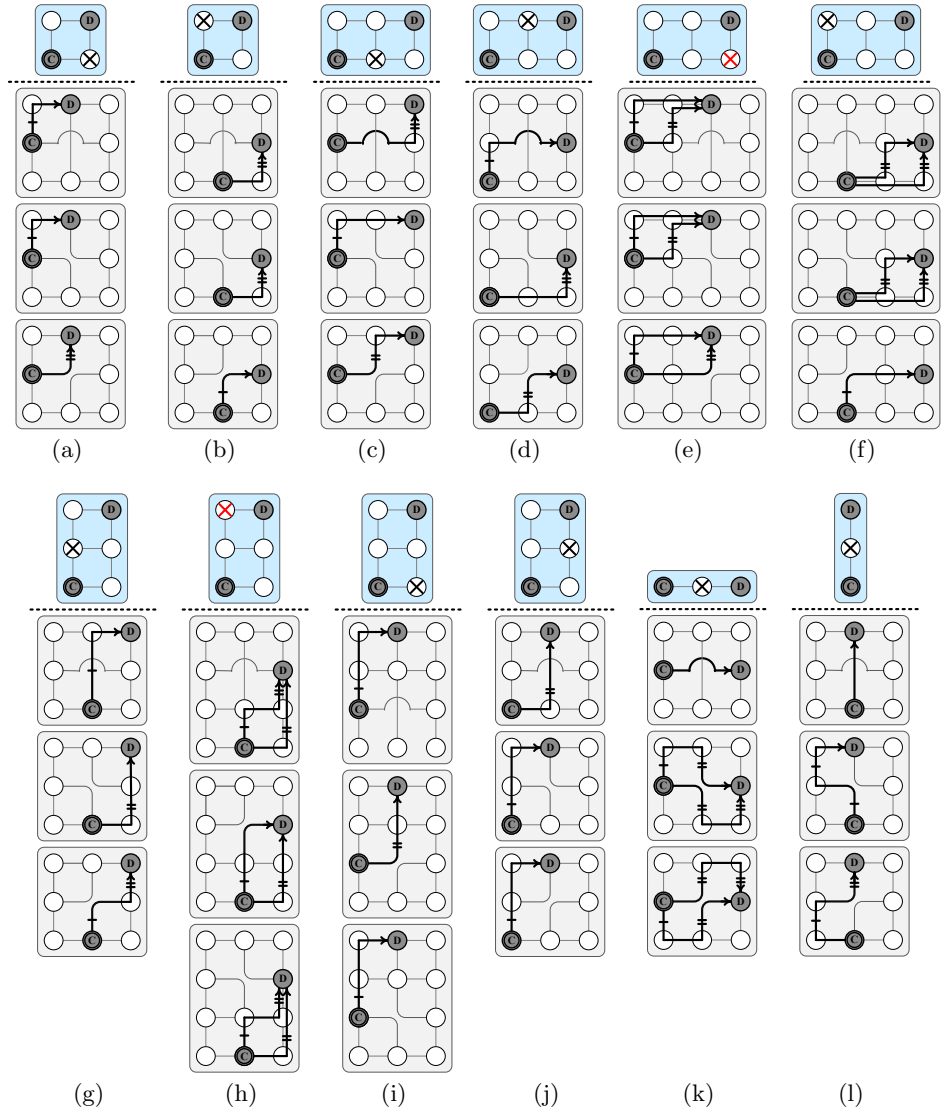

Figure 4: Tolerating single faulty switch using TARA when the destination is located on the (a)-(j) northeast, (k) east, and (l) north of the current node. The faulty switch which cannot be detected by the look-ahead information of the current node is marked in red.

over, it is adaptive and takes the congestion condition of the receiving routers into account whenever multiple paths are available. Since the number of reroute operations is limited in TARA, it is livelock-free as well.

When the distance from C to D is two hops or more along both dimensions, C forwards the packet to the healthy output channel. If both of the neighbors are healthy, the less congested neighbor is selected to send the packet. Besides northeast, routing the packets in other directions is performed in a similar fashion and is not discussed here because of the limited space.

4.3 Reconfiguration Issues

4.3.1 Reconfiguration Mechanism

In order to keep the complexity of RCU as low as possible, TARA utilizes a simple reconfiguration methodology which is performed as follows:

Step 1. We define *reconfiguration time frame windows* (R_w) during which the requests from the neighbors are collected. Three counters collect the statistics for each bypassing mode. Once the *Fault* signal is activated, the control mechanism of RCU is triggered such that the corresponding counter is updated for each complaint. At the end of the R_w, the counters are compared using a majority gate to determine which mode has the highest demand and the output bitstream is generated by RCU. R_w equals 100 cycles in this paper.

Step 2. Once a pattern change is observed, the faulty router configures itself to a new bypassing mode that best suits the traffic pattern of the current application.

4.3.2 Deadlock-freedom

Techniques employing dynamic reconfiguration [14, 8] have to pay a special attention to the *reconfiguration-induced* deadlock in addition to the *routing-induced* deadlock. Since we use a single deadlock-free routing algorithm (i.e. DyXY) which does not change during the topology reconfiguration, reconfiguration-induced deadlock cannot occur.

4.3.3 In-flight Packets

In order to avoid lost and dropped flits in a connected network, we need to make sure that each packet sits entirely in one switch along the route that is affected by the reconfiguration. Thus, before switching to a new topology, the following conditions should hold:

1) The Routing Unit (RU) of the upstream router should be temporarily halted to stop sending packets to the output ports. The reason is that the reconfiguration affects the inter-router connections and the output port assigned to a given packet may not be valid for the new bypassing mode.

2) The flits which have passed the RU, should successfully follow the remaining stages and leave the upstream router.

3) The reception of the tail flit of the in-flight packets should be acknowledged by the downstream router.

5. SIMULATION RESULTS

For appraising the efficiency of TARA, simulations were conducted for an 8×8 mesh using the BookSim 2.0 cycle-based network simulator [2]. The data width is assumed as 32 bits and each input channel has a buffer size of 16 flits. The congestion threshold is set to 62.5% of the buffer capacity, and the packet size is uniformly distributed between 5 and 10 flits. After the warm-up period, the results were averaged over 200,000 cycles. TARA was implemented using the universal Logic-Based Distributed Routing (uLBDR) [15] due to its efficiency to support irregular topologies. TARA was compared with MiCoF as the most closely relevant technique. We assumed that the reconfiguration takes two extra cycles, one cycle for making a decision and another cycle for configuring the router in the desired mode.

5.1 Performance Analysis

5.1.1 Synthetic Traffic

The average latency curves of TARA and MiCoF as a function of the network's message injection rate are plotted in Fig. 5, for different number of switch failures (indicated in the parentheses) under the uniform and hotspot traffic patterns. The faulty nodes were selected using a random function to ensure a fair comparison.

As shown by simulations, employing MiCoF results in a lower average latency compared to TARA under the uniform traffic. This is due to the reconfiguration strategy in TARA where the bypassing mode is selected according to the traffic history. In fact, this kind of reconfiguration is mostly wrong under the uniform traffic pattern. For instance, consider an interval where data is sent to the SW direction. TARA will configure the topology to provide direct access to that direction. However, SW has the lowest possibility to carry high traffic loads in the next interval, based on the uniform traffic model [14]. Therefore, the configuration in TARA is counterproductive and escalates the latency experienced by the packets. On the contrary, MiCoF which provides static routing paths oblivious to the underlying traffic flows, can achieve better performance under the uniform traffic.

For the hotspot traffic which represents a more realistic traffic model, four hotspot nodes at the center of the network were selected with an extra 25% traffic. Under the hotspot traffic, utilizing TARA results in a lower average latency compared to MiCoF. This can be explained by considering the fact that there exists a consistent traffic flow in the network such that the reconfiguration remains effective throughout the simulation. Thus, TARA is able to outperform the static MiCoF in tackling the congested areas by providing customized minimal paths at the occurrence of fault, and thereby reducing the average communication delay. Compared to TARA, the performance of MiCoF drops significantly because of the fixed paths that the packets have

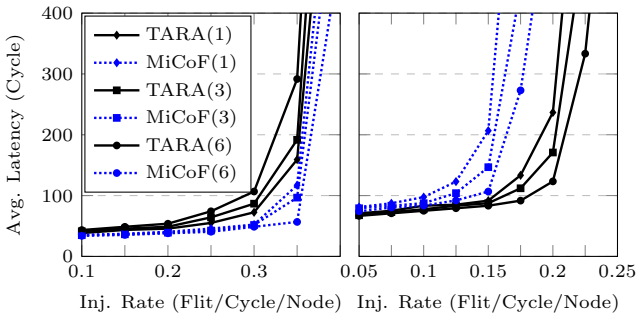

Figure 5: Performance comparison of an 8×8 mesh under uniform (left) and hotspot (right) traffic.

to follow to reach the destinations without considering the traffic flows. Moreover, both MiCoF and TARA provide better latency results as the number of failures increases in the network. The reason is bypassing the pipeline of the original datapath which can be translated into smaller hop count and faster transmission in the network.

5.1.2 *Trace-driven Traffic*

For a more realistic traffic analysis, we carried out the trace-driven simulations from SPLASH-2 benchmarks across an 8×8 NoC with 3 switch failures, which is configured with 20 processors and 44 shared L2 cache memory modules. Figure 6 shows the average packet latency across five benchmarks, normalized to MiCoF. TARA provides lower latency than MiCoF and it shows the greatest performance gain for *cholesky* with 22% reduction in latency. The average performance gain of TARA across all benchmarks is up to 16%.

5.2 Reliability Analysis

In order to assess the reliability of TARA, the number of faulty routers was increased from 1 to 10 under the uniform traffic profile. The reliability is measured as [4]:

$$\frac{\textit{No. of successful packet arrivals at the destination nodes}}{\textit{Total no. of delivered packets}}$$

The results are shown in Fig. 7. As can be seen, both methods are 100% reliable when there is a single faulty router in the network. TARA can tolerate up to ten faulty routers by more than 99.99% reliability whereas the reliability of MiCoF is around 76%. Note that MiCoF can tolerate multiple number of faulty switches in the network provided that they are not located in the diagonal positions [4].

5.3 Area Analysis

To assess the hardware cost of TARA, the routers were modeled with VHDL and synthesized by Synopsys Design Compiler using the TSMC 32nm standard cell library. Both routers use 2 VCs along the Y dimension. Compared with the MiCoF router, TARA increases the area by 24% which is mainly due to the reconfiguration circuitry. Assuming that the router area occupies 11% area of a tile as reported by Intel [14], the increase in the tile area using TARA will be around 2.6% which can be considered negligible.

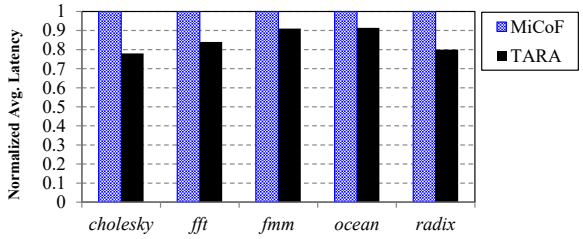

Figure 6: Performance for application traces.

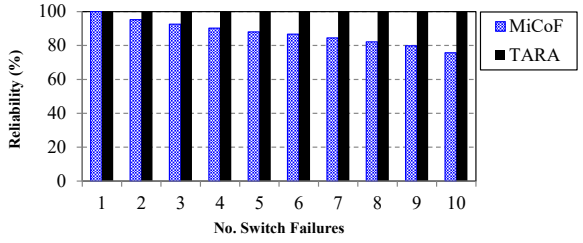

Figure 7: Reliability evaluation in an 8×8 mesh under the uniform traffic.

6. CONCLUSION

We have proposed a dynamically reconfigurable architecture (TARA) for wormhole-switched 2D mesh NoCs. The main advantages of TARA over the conventional schemes is its ability to dynamically tailor the failed switch to the current traffic pattern such that not only the connectivity of the network is maintained, but also the performance can be improved. Moreover, TARA supports adaptive, fully distributed, minimal, and algorithmic routing without using routing tables.

7. REFERENCES

[1] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw. A highly resilient routing algorithm for fault-tolerant NoCs. In *Proc. ACM/IEEE Design Automat. Test in Europe (DATE)*, pages 21–26, 2009.

[2] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, San Francisco, USA, 2004.

[3] M. Koibuchi, H. Matsutani, H. Amano, and T. M. Pinkston. A lightweight fault-tolerant mechanism for Network-on-Chip. In *Proc. ACM/IEEE Int. Symp. Networks-on-Chip (NOCS)*, pages 13–22, 2008.

[4] M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen. Minimal-path fault-tolerant approach using connection-retaining structure in Networks-on-Chip. In *Proc. ACM/IEEE Int. Symp. Networks-on-Chip (NOCS)*, pages 1–8, 2013.

[5] A. Prodromou, A. Panteli, C. Nicopoulos, and Y. Sazeides. NoCAlert: An on-line and real-time fault detection mechanism for Network-on-Chip architectures. In *Proc. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, pages 60–71, 2012.

[6] P. Ren, Q. Meng, X. Ren, and N. Zheng. Fault-tolerant routing for on-chip network without using virtual channels. In *Proc. ACM/IEEE Design Automat. Conf. (DAC)*, pages 1–6, 2014.

[7] D. Lee, R. Parikh, and V. Bertacco. Highly fault-tolerant NoC routing with application-aware congestion management. In *Proc. ACM/IEEE Int. Symp. Networks-on-Chip (NOCS)*, pages 1–8, 2015.

[8] R. Parikh and V. Bertacco. uDIREC: Unified diagnosis and reconfiguration for frugal bypass of NoC faults. In *Proc. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, pages 148–159, 2013.

[9] C. Iordanou, V. Soteriou, and K. Aisopos. Hermes: Architecting a top-performing fault-tolerant routing algorithm for Networks-on-Chips. In *Proc. IEEE Int. Conf. Comp. Design (ICCD)*, pages 424–431, 2014.

[10] L. Zhang, Y. Han, Q. Xu, X. w. Li, and H. Li. On topology reconfiguration for defect-tolerant NoC-based homogeneous manycore systems. *IEEE Trans. VLSI Syst.*, 17(9):1173–1186, 2009.

[11] M. Modarresi, A. Tavakkol, and H. Sarbazi-Azad. Application-aware topology reconfiguration for on-chip networks. *IEEE Trans. VLSI Syst.*, 19(11):2010–2022, 2011.

[12] F. Dubois, A. Sheibanyrad, F. Pétrot, and M. Bahmani. Elevator-First: A deadlock-free distributed routing algorithm for vertically partially connected 3D-NoCs. *IEEE Trans. Comput.*, 62(3):609–615, 2013.

[13] M. Li, Q.-A. Zeng, and W.-B. Jone. DyXY - A proximity congestion-aware deadlock-free dynamic routing method for Network on chip. In *Proc. ACM/IEEE Design Automat. Conf. (DAC)*, pages 849–852, 2006.

[14] B. Fu, Y. Han, J. Ma, H. Li, and X. Li. An abacus turn model for time/space-efficient reconfigurable routing. In *Proc. Int. Symp. Comput. Arch. (ISCA)*, pages 259–270, 2011.

[15] S. Rodrigo, J. Flich, A. Roca, S. Medardoni, D. Bertozzi, J. Camacho, F. Silla, and J. Duato. Cost-efficient on-chip routing implementations for CMP and MPSoC systems. *IEEE Trans. Comp.-Aided Design Integr. Circ. Syst.*, 30(4):534–547, 2011.