



A Low Overhead Fault Reporting Scheme for Resilient 3D Network-on-Chip Applications

Michael Opoku Agyeman

Department of Computer Science and Engineering
The Chinese University of Hong Kong

ABSTRACT

Recently three-dimensional Networks-on-Chips (3D NoCs) ranging from regular to highly irregular topologies have been realized as efforts to improve the performance of applications in both general purpose and application-specific multi-core domain. However, faults can cause high contentions in NoCs. As a solution, adaptive routing algorithms are used. On the other hand, these algorithms have high area and timing overheads due to extra logic required for adaptively. We present a novel fault reporting scheme as well as a fault-tolerant routing algorithm for emerging 3D NoCs. The proposed algorithm analyses the condition of the NoC resources and distance of the destination nodes to reroute packets. The algorithm has been evaluated by synthetic and various real-world traffic patterns. Experimental results show that the proposed algorithm has significant reduction in packet delays (over 45%) compared to other algorithms.

Keywords

Networks-on-Chip, Fault-tolerant, Routing algorithms

1. INTRODUCTION

Modern Systems-on-Chip (SoCs) are implemented with billions of transistors with heterogeneous integration of various components and large number of cores to provide enhanced functionality of applications. However, as technology scales with increased complexity, traditional bus-based SoCs fail to meet the communication demands. Network-on-Chip (NoC) has evolved as a viable solution, providing a scalar communication framework with enhanced parallelism and adaptivity [11]. In NoC, the idea of macro network communication is adopted. Here, communication among processing elements is achieved by sending packets of data in a circuit switched or packet switched manner using switches, routers and links but, with a limited amount of resources due to on-chip constraints [2].

Traditional 2D integrated circuit (IC) has larger floorplan, power consumption and a low performance which scales poorly as the number of cores per-chip planer increases [12]. 3D integration technology has been proposed to stack vertically several of 2D silicon layers to optimize the long interconnect wires and larger footprint of SoC design on the traditional 2D IC. The layers in the 3D IC provide shorter inter-layer wires and more resource connectivity with reduced hop-count compared to the long interconnect wires in a 2D SoC implementation with equivalent number of resources [13]. Moreover, 3D technology allows heterogeneous integration of technologies and design disciplines such as RF, digital, analog, etc. on a single chip. Consequently, three-dimensional Networks-on-Chip (3D NoCs) have emerged as a promising technology that maintains the benefits of miniaturization by enabling higher integration density and enhancing system performance while providing a scalable communication platform for multi-core architectures. However, 3D NoC communication infrastructure still depends on wires, and switches which are subject to faults as technology scales. Such conditions can lead to high contentions or even deadlock conditions if optimized recovery algorithms are not employed. Fault-tolerant routing enables packets to be redirected along non-faulty paths at run-time

upon NoC failure conditions. Consequently, it is necessary that 3D NoC routing schemes are made more viable through the realization of efficient routing algorithms. Such algorithms should be effectively implemented to adapt and dynamically restore traffic flows under faulty conditions while maintaining the performance benefits of 3D NoCs with minimum area overhead and complexity. In contrast, most existing fault-tolerant routing algorithms employ multi-hop virtual channels (VC) techniques which require extra buffers in the routers [1, 3, 20]. The main contribution of this paper is a novel fault reporting scheme as well as an adaptive fault-tolerant routing algorithm for 3D NoCs that can efficiently send packets to their destinations even under multiple fault conditions. We achieve this by employing an efficient minimum path sequencer during routing under fault conditions.

The paper is organized as follows. Section 2 presents a brief description of the related work. Section 3 presents details of the proposed routing algorithm. In Sections 4, we present the implementation details and analysis of the experimental results. Finally, we summarize the main findings in Section 5.

2. RELATED WORK

In the evolution of NoCs to support current era of multi-core design, several fault tolerant routing algorithms have emerged in 2D NoC topologies based on various approaches such as turn models, faulty blocks, intermediary nodes, virtual channels etc [3]. Recently, researchers have made some efforts to adopt these algorithms in 3D NoCs [16]. However, the diversity of NoCs in the third dimension provides several opportunities and constraints. A simple extension of 2D fault tolerant routing algorithms is not sufficient in exploring the opportunities provided by the shorter vertical interconnects. Glass et al. [4] proposed turn model which adaptively provides deadlock-free paths for packets by restricting the number turns allowed in the NoC. Similarly Jie et al. [18] proposed the odd-even turn model. Though no virtual channels are required in both techniques, they have strict fault requirements and their performance scale poorly as the number of faults increases. Moreover by restricting the number of turns, such algorithms have high packet delays under non-uniform traffic patterns with low traffic conditions [9].

Other fault tolerant routing algorithm presented in literature employ adaptive routing algorithms to reroute packet when faults are encountered [3, 15, 20]. Here, packets are rerouted along different edges from the main path when a fault is encountered. The algorithms employ two or more virtual channels to ensure deadlock freedom. Prunte et al. [14] also proposed a fault tolerant routing algorithm that require two virtual channels. Dubios et al. [6] presented the *elevator first* distributed routing which require two virtual channels. Fault routing technique proposed by Gomez et al. [8] can handle up to five faulty links. Similarly, the Valinataj et al. [17] proposed the RAFT which can tolerate one faulty link. Due to the multi-hop nature of the error reporting scheme, packet rerouting relies on network status which may be outdated.

Our proposed algorithm supports multiple fault conditions. Contrary to previous work, our algorithm has no particular fault restrictions. Additionally, we present a low traffic overhead fault reporting scheme which can efficiently report single and multi-

Algorithm 1: Fault reporting

```

begin
  /* If Node/switch/link fails */
  for each Core do
    repeat
      if  $core_x = Fault_x$  then
        Report to  $core_{x+1}$  and  $core_{x-1}$ 
        if  $core_y > Fault_y$  then
          Report to  $core_{x,y+1,z}$ 
        else if  $core_y < Fault_y$  then
          Report to  $core_{x,y-1,z}$ 
        end
      else if  $core_x > Fault_x$  then
        Report to  $core_{x+1,y,z}$ 
      end
    else
      Report to  $core_{x-1,y,z}$ 
    end
  until last tile;
end
end
  
```

ple fault conditions at runtime with minimum hops. We also provide hardware implementation results which emphasize on the significance of our proposed algorithm.

3. PROPOSED ALGORITHM

To efficiently route packets under 3D NoCs while avoiding deadlocks and faulty links, the proposed routing algorithm evaluates the cost of sending packets along various alternative routes. Here, the cost is defined as a function of both distance and direction between the current and destination nodes. Hence, the algorithm aims to minimize the average number of hops while providing as many alternate routes as possible. To do this, it is necessary to identify fault conditions at runtime and reroute packets along different routes without introducing deadlocks and livelocks while keeping an optimized NoC performance. Moreover, faults may either occur at runtime (before or after packets are generated), before system start-up or upon system restart or a combination of these scenarios. Hence, we propose an error reporting scheme to efficiently identify faults under the above mentioned conditions.

3.1 Fault Detection

Intuitively, non-faulty nodes connected to a failed node can send failure reports to neighbouring nodes and neighbouring nodes also forward the report to nodes connected to them [16]. After several cycles fault conditions can be reported to all nodes. However, as shown in Figure 1(a) this technique is inefficient and leads to message redundancy which increases the traffic and contention in the NoC. We present an efficient real-time fault reporting algorithm that handles failure conditions with minimum number of hops and traffic overhead. Here we adopt a tagging system initiated by the nearest non-faulty component (*report initiator*) connected to a faulty component. Thus upon a fault, the nodes connected to the fault link/switch send a single flit long error report to their neighbouring nodes. With reference to the Cartesian position of the fault and the current node, the nodes forward the error report to selected nodes. Details of the proposed error reporting for a node $Core_{xyz}$ in a 3D NoC with an error at node $Fault_{xyz}$ are highlighted in Algorithm 1. By adopting this algorithm, errors are sent along paths with minimum hops while avoiding retransmission of error report. The example in Figure 1 shows that the proposed error reporting al-

gorithm works efficiently under both single and multiple fault conditions.

3.2 Fault information propagation network

The above error reporting scheme, although can reach all nodes with minimum packet overhead, relies on inter-node multi-hop communication. Also, the router buffers are shared between the fault reporting flits and the NoC data packets which might introduce contention in the network. Hence, we implement a low-latency fault propagation network which is able to update the faults in the network by a single bit transmission system. Upon the occurrence of a fault, an offline bit information for the faulty node or link is forwarded to the other nodes instantaneously with minimum hop-count, forming a single-bit-minimum-cycle (SBMC) delay per hop propagation network. Hence we call our fault information propagation network an *SBMC network*. Through this network, each a fault can be reported to all nodes via the reporting scheme presented in Algorithm 1. To achieve the minimum hop-count two considerations are made:

- (1) Two set of wires are available in the 3D NoC: intra-layer wires and inter-layer wires which are mostly Through-Silicon-Vias (TSVs)
- (2) TSVs have shorter delays compared with the intra-layer wires

Therefore to avoid complex design and manufacturability issues, we achieve single hop within the layers via the SBMC network with 4 set of wires. For reporting fault to the other layers, first, a single-bit report is sent along an SBMC TSVs network in one-cycle. Another cycle is required to send the report within the layers. Thus a total of two-cycles are required for reporting faults to nodes in other layers. Thus, given $p \in P$ as the number of ports of an *initiator* node (the nearest node connected to the faulty link/switch) which are not directly blocked by the fault, $p = [1, 5]$ set of wires a required for the SBMC error reporting scheme. Each set of wires reports the error instance to the associated unblocked direction. An example of the fault information propagation network for the fault scenario in Figure 1(b) is shown in Figure 2. In this example, the *Down* port of node 16 is blocked and considered faulty, leaving $p = 5$ and $p = 4$ unblocked ports or paths on node 16 and 4, respectively. Figure 2 shows the fault reporting scheme via the SBMC network. Nodes 4 and 16 are the *initiator* nodes in this case as they are the closest NoC components to the fault. In this example, the main network links are omitted for clarification. Hence, all the wires except the faulty link in Figure 2 is one-bit wide. The 5 set of wires reports the fault conditions to all the nodes reachable through node 16 via the unblocked paths in the *North*, *South*, *East*, *West* and *Up* directions. By employing the SBMC network and the relative coordinates of a source node to the *initiator*, faulty nodes/links in the network avoided accordingly. Similarly, node 4 reports the fault to all the nodes in the lowest layer in a single cycle via the SBMC network. Propagating the fault information along the main network requires one flit. When used efficiently, the data width of the fault reporting scheme could be reduced to avoid contention along the propagation network. Consequently, we use 1-bit data to report the fault incidence along. If a fault occurs on any direction of an *initiator* node, the error propagation network corresponding to that node is asserted (or de-asserted otherwise).

3.3 Fault-Tolerant Routing

Under no fault conditions, the algorithm always sends packets along the path with minimum cost between the current core and destination nodes as the main route. If there is a fault on the main route due to a link or switch failure, packets are sent along the path with the second minimum cost as the first alternative route.

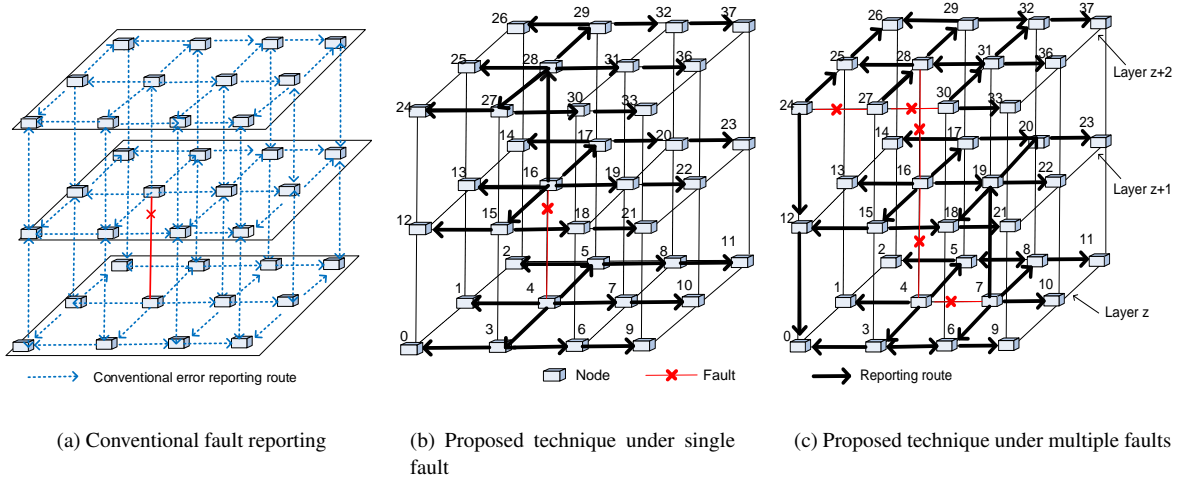


Fig. 1. Traditional vs proposed fault reporting techniques.

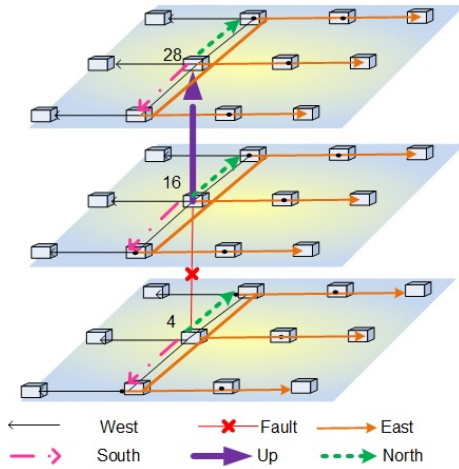


Fig. 2. Proposed fault propagation network. Each line is one-bit wide for single-hop intra- and inter-layer traversal.

However, if both routes are not accessible due to fault conditions, a third alternative route which is the next minimum cost after the previous route is selected. As the number inaccessible routes increases due to increase in the number of faults, the cost in routing packets increase gradually. Using alternate paths with minimum number of hops help in keeping the cost low under multiple fault conditions. However, if the decision is always made at the current node under multiple source conditions, packets may be sent along longer paths. Hence by adopting our efficient error reporting policy to report faults and recovered faults, we consider the minimal paths upon packet generation at the source node when possible. Given R as the set of non-faulty nodes in the XYZ routing path between a source node and a destination node, the proposed fault tolerant routing algorithm is given in Algorithm 2. It should be noted that although the proposed algorithm adapts to faulty conditions, it is deadlock free because it maintains minimal path XY routing for 2D planer communication and unidirectional multi-hop stages along the z axis to the destination layer. Two virtual channels are used per port; main and escape channels. At each routing step a message can visit an available main route making it one hop closer to destination node using main virtual channels. If the path on all the adaptive channels are blocked due to faults, however, the message takes the escape

Algorithm 2: Fault-Tolerant Routing

```

begin
  for all nodes  $d \in R$  do
    repeat
      find the shortest path  $p$  to  $d$ 
      if  $p$  contains at least one Fault then
        update flt flags
        break
      else
        path  $P = p$ 
      end
    until last  $d$ ;
  end
  route using  $P$ 
end
  
```

channel, if not busy, and traverses the rest of its path using XY deterministic routing algorithm. This technique has been extensively studied in the literature and proved to be deadlock free. For comparison purpose, we also implemented another adaptive fault tolerant routing algorithm (VC_XYZ) by adopting dimension-ordered XYZ routing. Here, packets are forwarded along the X dimension, then Y and finally along the Z to their destination nodes. If the X dimension fails, packets are sent along the Y and then continue in the X dimension. Similarly if Y fails, packets are sent along the X and then forwarded to the Y dimension again. However, if there is a fault in the Z dimension, packets are then XY routed to the immediate vertical link.

This algorithm enable messages to explore immediate alternative paths at a node while requiring only two virtual channel per router port to ensure deadlock-freedom. To ease comparison and simplicity, the proposed fault routing algorithm will also be referred as $SBMC_FR$ in the rest of the paper. As shown in the Figure 3, $SBMC_FR$ sends packets along paths with lower hop-counts compared to virtual channel based VC_XYZ . Specifically Figure 3(b) shows that $SBMC_FR$ efficiently selects a path with minimum hop-count under multiple fault conditions.

4. EVALUATION

In order to evaluate the performance of the proposed algorithm, a cycle-accurate NoC simulator is used by extending *Wormsim* [10], an existing 2D NoC simulator. Our extended simulator em-

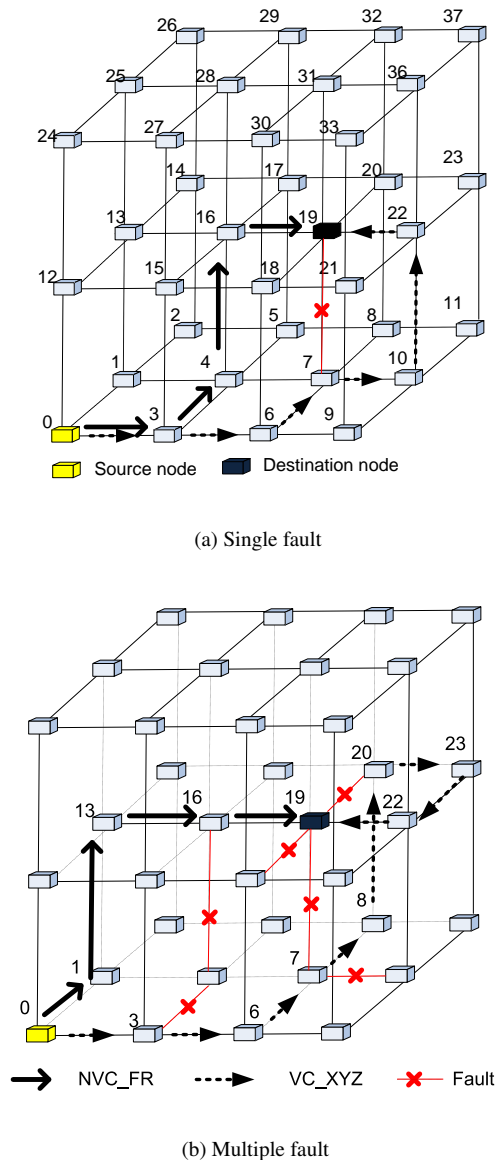


Fig. 3. Comparison between the two routing algorithms under fault conditions.

employs wormhole packet switching flow control to accurately simulate 2D and 3D NoCs with any configuration. We implemented the NoC components at the Register Transfer Level (RTL) in VHDL language and implemented on a 40nm CMOS process technology. Energy consumption of each component is then imported into the simulation platform. Energy consumption of the NoC was estimated using E_{bit} energy model [19]. In the simulation, a fixed packet size of 5 flits is used in the NoC model. In order to evaluate the performance sustainability and saturation points of the network, we performed our analysis with uniform traffic pattern under different injection rates, where a node sends packets with equal probability to other nodes. Moreover, in order to investigate the performance and energy of the NoC in real-world scenarios four realistic traffic patterns have been considered: a complex multimedia traffic (MMS) [10], Atuto-indust and Telecom (from the E3S benchmark suite) [5] and an AV (Audio-visual) benchmark [7]. The setup is run for a warm-up period of 2000 cycles and performance statistics are collected after a simulation length of 20,000

simulation cycles¹. Hence, by introducing different delay and energy models of the NoC components, we have compared the average packet latency and energy consumption.

4.1 Performance and Energy Results

In this section, we first look at the latency and energy of the proposed fault tolerant routing and virtual channel based adaptive XYZ fault tolerant routing under uniform traffic pattern. Figure 4 shows that by reducing the number of hops and avoiding critical paths in the first place, the proposed *SBMC_FR* routing has lower average packet latency and can sustain higher injection rates (an average of about 25%) than virtual channel based *VC_XYZ*. Particularly, Figures 4(a) to 4(c) shows that the performance overhead of the two fault tolerance increases from 20% to over 45% in favour of *SBMC_FR* as the number of faults increases. Moreover as can be clearly seen in Figure 4(d), the energy consumption of NoCs with *SBMC_FR* fault tolerant routing is lower than NoCs with *VC_XYZ* fault routing under fault conditions. Comparatively, the average energy consumption in *SBMC_FR* decreases even further as the number of faults increases. This is expected as redirected packets take shorter routes with minimum hops to their destination nodes in *SBMC_FR*. Additionally this is partly due to the absence of virtual channels with less logic gates and no extra multiplexers in the router architecture of Optimized XYZ.

To validate the performance of our proposed algorithm, Figure 5 compares the average packet latency of the proposed algorithm with *VC_XYZ* algorithm under transpose, hotspot and various realistic traffic patterns by randomly introducing 6 faulty links in 3D NoCs. Figures 5(a) and 5(b) show that even under non-uniform traffic patterns and multiple fault conditions, the proposed *SBMC_FR* algorithm has lower average packet latency compared to *VC_XYZ* which has extra virtual channels. Most importantly, Figure 5(c) shows that though only 6 faults were introduced into the network, our proposed algorithm has superior latency values in all cases under real world applications when compared to that of virtual channel based fault routing algorithm. Particularly, our proposed algorithm has lower average packet latency when compare to RILM ([16]) and Turn model ([6])

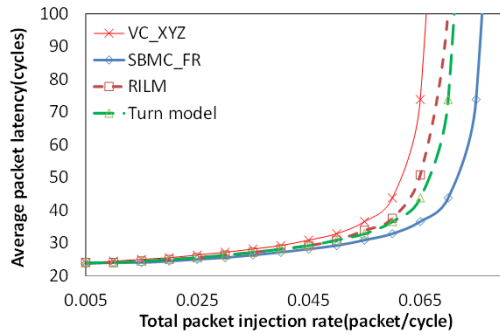
4.2 Area Evaluation

As a final evaluation, we implemented the proposed fault tolerant routing algorithm in hardware at the Register Transfer Level (RTL) in VHDL language on a 40nm CMOS process technology and Synthesized it with Synopsis Design Compiler. The area overhead of the router's routing logic block with the proposed fault tolerant routing algorithm and the error reporting scheme was less than 1% of the gate count of the routing engine block of conventional 3D router which is considered insignificant. It should be noted that the router crossbar architecture and buffer resources which are not affected by our algorithm forms a high percentage of the area overhead of NoC router architecture.

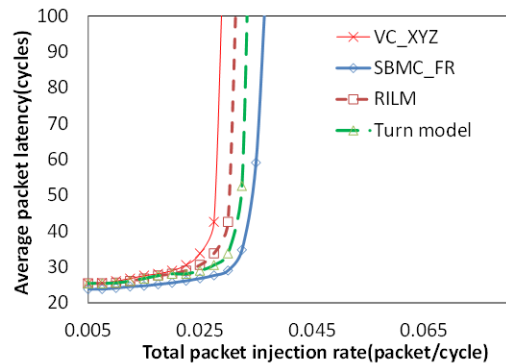
5. CONCLUSION

In this paper, we have proposed an efficient fault reporting and routing scheme for both 2D and 3D NoCs based on deadlock-free minimal path routing. Both the proposed and existing fault tolerant routing schemes have been evaluated with a cycle accurate simulator under synthetic and realistic traffic patterns. Our

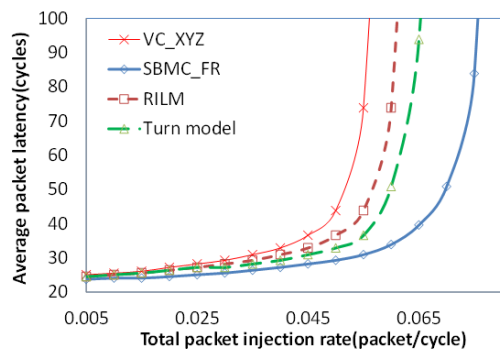
¹Experiments with simulation length greater than 200,000 were conducted in all the applications. However, our experiments show a constant simulation results after a simulation period of 5,000 cycles. Hence, a simulation period of 20,000 is suitable for our experimental setup. A NoC size of $4 \times 4 \times 4$ for uniform and MMS (or $3 \times 4 \times 2$ otherwise) was assumed.



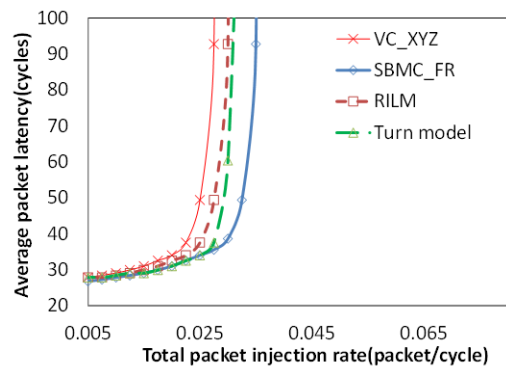
(a) Uniform 2 faults



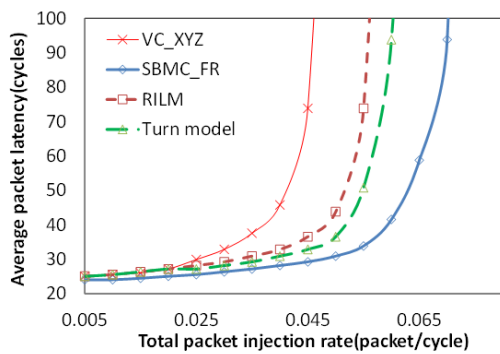
(a) Hotspot



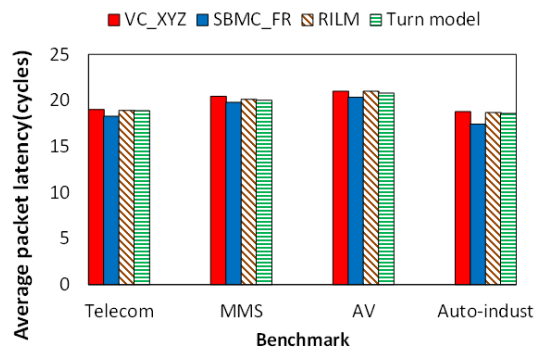
(b) Uniform 6 faults



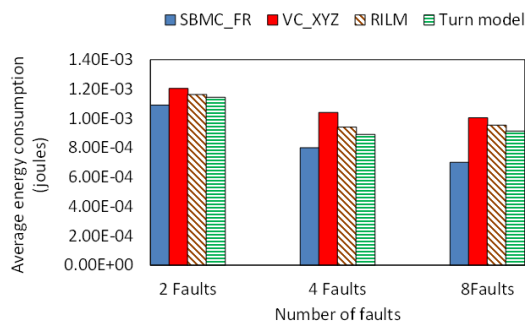
(b) Transpose



(c) Uniform 12 faults



(c) Benchmarks



(d) Average Energy Consumption

Fig. 4. Average packet latency and energy under uniform traffic pattern.

Fig. 5. Average packet latency and energy under various traffic patterns.

evaluation under three different synthetic traffic patterns and four real world applications shows a significant improvement in the average packet latency and energy consumption when our proposed algorithm is compared to other fault tolerant routing algorithms. Particularly, experimental analysis shows that, the proposed routing algorithm can support multiple fault conditions with an improved performance of over 45% in the average packet latency and lower energy consumption when compared to existing virtual channel based routing algorithms under heavy uniform traffic conditions.



6. REFERENCES

- [1] M.A. Al Faruque and J. Henkel. Minimizing virtual channel buffer for routers in on-chip communication architectures. In *Design, Automation and Test in Europe (DATE)*, pages 1238 –1243, march 2008.
- [2] L. Benini and G. De Micheli. Networks on chips: a new SoC paradigm. *Computer*, 35(1):70–78, 2002.
- [3] Jun Chen, Du Xu, and Ling fu Xie. A fault-tolerant routing protocol in 2d torus based on positive-first and negative-first turn models. In *International Conference on Information Engineering and Computer Science (ICIECS)*, pages 1 –5, 2009.
- [4] Glass Christopher J and Lionel M. Ni. Fault-tolerant wormhole routing in meshes without virtual channels. *IEEE Transactions on Parallel and Distributed Systems*, 7(6):620 –636, 1996.
- [5] R. Dick. Embedded system synthesis benchmarks suite (e3s). <http://ziyang.eecs.umich.edu/dickrp/e3s/>. Accessed: 06/2010.
- [6] F. Dubois, A. Sheibanyrad, F. Petrot, and M. Bahmani. Elevator-first: a deadlock-free distributed routing algorithm for vertically partially connected 3d-nocs. *IEEE Transactions on Computers*, PP(99):1, 2011.
- [7] V. Dumitriu and G.N. Khan. Throughput-oriented noc topology generation and analysis for high performance socs. *VLSI*, 17(10):1433 –1446, 2009.
- [8] María Engracia Gómez, José Duato, Jose Flich, Pedro López, Antonio Robles, Nils Agne Nordbotten, Olav Lysne, and Tor Skeie. An efficient fault-tolerant routing methodology for meshes and tori. *Computer Architecture Letters*, 3, 2004.
- [9] Jingcao Hu and R. Marculescu. Dyad - smart routing for networks-on-chip. In *Design Automation Conference (DAC)*, pages 260 – 263, 2004.
- [10] Jingcao Hu and R. Marculescu. Energy- and performance-aware mapping for regular NoC architectures. *Computer-Aided Design of Integrated Circuits and Systems*, 24(4):551–562, 2005.
- [11] G. De Micheli and L. Benini. *Networks on Chips: Technology and Tools*. Morgan Kaufmann, First Edition, 2006.
- [12] Partha Pratim Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. *IEEE Transactions on Computers*, 54(8):1025–1040, 2005.
- [13] Vasilis F. Pavlidis and Eby G. Friedman. Interconnect delay minimization through interlayer via placement. In *in 3-D ICs, in Proc. Great Lakes Symposium on VLSI*, pages 20–25, 2005.
- [14] V. Puente, J.A. Gregorio, F. Vallejo, and R. Beivide. Immunet: a cheap and robust fault-tolerant packet routing mechanism. In *Annual International Symposium on Computer Architecture*, pages 198 – 209, 2004.
- [15] S. Rodrigo, J. Flich, J. Duato, and M. Hummel. Efficient unicast and multicast support for cmps. In *IEEE/ACM International Symposium on Microarchitecture (MICRO-41)*, pages 364 –375, 2008.
- [16] Claudia Rusu, Lorena Anghel, and Dimiter Avresky. Rilm: Reconfigurable inter-layer routing mechanism for 3d multi-layer networks-on-chip. *IEEE International On-Line Testing Symposium*, 0:121–126, 2010.
- [17] M. Valinataj and S. Mohammadi. A fault-aware, reconfigurable and adaptive routing algorithm for noc applications. In *IEEE/IFIP VLSI System on Chip Conference (VLSI-SoC)*, pages 13 –18, sept. 2010.
- [18] Jie Wu. A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model. *IEEE Transactions on Computers*, 52(9):1154 – 1169, 2003.
- [19] Terry Tao Ye, Giovanni De Micheli, and Luca Benini. Analysis of power consumption on switch fabrics in network routers. In *DAC*, pages 524 – 529, 2002.
- [20] Jipeng Zhou and Francis C.M. Lau. Multi-phase minimal fault-tolerant wormhole routing in meshes. *Parallel Computing*, 30(3):423 – 442.