

Performance Evaluation of Adaptive Routing Algorithms for achieving Fault Tolerance in NoC Fabrics

Haibo Zhu¹, Partha Pratim Pande¹, Cristian Grecu²

¹*School of Electrical Engineering & Computer Science
Washington State University
PO BOX 642752
Pullman, WA, USA
(haibo, pande)@eecs.wsu.edu*

²*SoC Research Lab
Department of Electrical and Computer Engineering
University of British Columbia
2332 Main Mall Vancouver, BC, V6T 1Z4, Canada
grecuc@ece.ubc.ca*

Abstract

Commercial designs are integrating from 10 to 100 embedded functional and storage blocks in a single system on chip (SoC) currently, and the number is likely to increase significantly in the near future. The communication requirements of these large Multi Processor SoCs (MP-SoCs) are conformed by the emerging network-on-a-chip (NoC) paradigm. In the deep sub-micron (DSM) VLSI processes, it is difficult to guarantee correct fabrication with an acceptable yield without employing design techniques that take into account the intrinsic existence of manufacturing faults. To become a viable alternative IC design methodology the NoC paradigm must address the system-level reliability issues, which is going to be the dominant concern in the DSM and beyond silicon era. By incorporating adaptiveness in the data communication mechanism we are able to tolerate permanent manufacturing faults in the NoC interconnect architectures. The corresponding performance and cost figures must be carefully analyzed and weighted against the specific application requirements. In this paper we explore the performance tradeoffs associated with adaptive routing schemes in NoC fabrics.

1. Introduction and motivation

The idea of integrating numerous components of a computer system into a single chip has led to the miniaturization of many portable devices and an increase in their computational capabilities. The possibility of this higher degree of integration has led to the concept of System on Chip (SoC). Current SoC designs are appearing with very large numbers of embedded processors, ranging between 8 to 32 in communications and network processing, security processors, storage array networks, and wireless base stations; to over 100 processors in recent platforms in consumer image processing, and high-end network processors. As the complexity of SoC increases, communication among the constituent Intellectual Property (IP) blocks becomes the main challenge. In this context, Network-on-Chip (NoC) [1] [2] is emerging as a promising design paradigm to replace the traditional bus-based systems.

For deep sub-micron (DSM) VLSI processes, it is

difficult to guarantee correct fabrication with an acceptable yield without employing design techniques that take into account existence of manufacturing defects. Moreover, the life-time reliability of DSM devices is likely to be compromised by effects such as electromigration and material ageing [3]. In order to improve the reliability of multi-core SoCs, their interconnect infrastructures must be designed such that fabrication and life-time faults can be tolerated. These irrecoverable faults influence the behavior of NoC fabrics and consequently degrade the system performance. Therefore, achieving on-chip fault tolerant communication is becoming increasingly important in presence of such permanent faults.

In the NoC environment initially deterministic routing algorithms were employed due to the ease of implementation. The primary limitation of deterministic routing is that it establishes a fixed path between a pair of source and destination nodes. Consequently it demonstrates poor performance in presence of faults situated on the routing path as it fails to establish alternate routes. A certain level of performance can be maintained in presence of faulty components if adaptive routing algorithms are adopted. One of the principal characteristics of adaptive routing algorithms is the ability to establish alternate routing paths in presence of faults. In this paper, we investigate applicability of adaptive routing method in NoC communication infrastructure, which helps us to avoid permanently faulty nodes/links while communicating between a pair of source and destination nodes. Though the adaptive routing methodologies help in maintaining a certain level of performance in presence of faults, it adds overhead in terms of silicon area and energy dissipation. We explore design trade-offs that characterize performance of NoC architectures in presence of permanent faults by incorporating adaptive routing algorithms.

2. Related work

With technology scaling, fault tolerance of the communication infrastructure is becoming a key challenge for designing NoCs. Though NoC research has gained significant momentum, the aspect of fault tolerance is not addressed adequately. Initial NoC research primarily

concentrated on deterministic routing algorithms, but to make the system fault tolerant, we need to adopt more complex routing mechanisms. In [4], stochastic communication paradigm is proposed to achieve fault tolerance in NoC architectures. The embedded cores communicate using a probabilistic broadcast: data packet is forwarded to a randomly chosen neighboring node until the entire network becomes aware of it. Even though this approach spreads the packet with an exponentially fast broadcast speed, it requires significant energy consumption in order to achieve a higher system performance by increasing the probability of transmission. Furthermore, a packet keeps propagating even if it reaches the destination unless the parameter, Time-To-Live (TTL) goes to zero. As a result, valuable network resources are kept busy in sending a successfully received message repeatedly instead of useful new information. In [5], the authors inherited the randomized gossip protocol from [4] to address fault tolerance in NoCs. Performance of different stochastic routing algorithms, viz. directed flooding, probabilistic flooding and random walk were investigated in NoC scenario. The principal limitation of these algorithms is that they can only sustain a very low traffic injection rate. This arises due to the fact that multiple copies of a single message are injected into the network to improve successful data arrival rate. Design of a low latency router supporting adaptivity for on-chip interconnects is described in [9]. But the authors have not quantified the performance of the NoC with increasing number of faults. The turn model is a well known partial adaptive routing algorithm, widely investigated for multi-processor environments [10] [11]. West-first routing, north-last routing and negative-first routing are three basic types of turn models. Compared to fully adaptive routing algorithms, turn model algorithm is a partially adaptive algorithm because two turns out of eight are forbidden in order to avoid deadlock. In [14], the authors combined deterministic x-y routing and adaptive routing in a single router. The routing scheme switches from deterministic to adaptive routing depending on the network congestion. The partial adaptive routing algorithm adopted in [14] is the odd-even turn model [13]. The odd-even turn model prohibits some types of turns based on the locations of the nodes in order to make itself deadlock free. More specifically, a packet is not allowed to make east-to-north or east-to-south turns at nodes located on even columns, and north-to-west or south-to-west turns at nodes located on odd columns [13]. The performance evaluation in [13] shows that the negative-first and odd-even turn models have very competitive performance depending on the traffic scenario.

In this paper, our aim is to evaluate the performance of a simple partially adaptive routing algorithm, like the negative-first routing algorithm compared to a stochastic method like random walk in presence of permanent faults when applied to NoC architectures

3. Fault tolerant routing algorithms in NoC

The common characteristic of NoC architectures is that the constituent IP cores communicate with each other through switches. Generally wormhole routing is adopted [12]. As shown in Figure 1, one of the widely used NoC topologies is the Mesh architecture. We analyze the performance of a Mesh-based NoC in presence of permanent faults when different adaptive and stochastic routing algorithms are adopted.

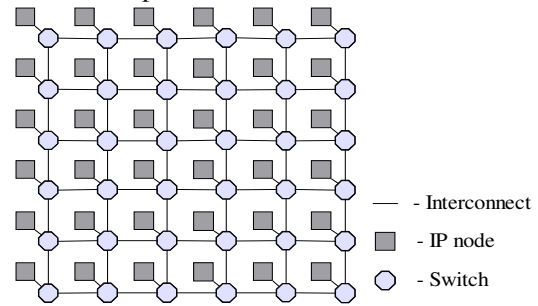


Figure 1: Mesh network

The x-y algorithm is a simple deterministic routing methodology used in Mesh networks [12]. The basic idea of this routing algorithm is that it routes data packets by crossing dimensions in strictly increasing or decreasing order, reducing the offset to zero in one dimension before routing in the next one. Though the x-y routing algorithm is easy to implement and has low overhead, it cannot maintain the desired level of performance in presence of faults. On the other hand fully adaptive algorithm, such as adaptive fault-tolerant wormhole routing algorithm [6] [7] [8], perform better in presence of faults, but due to complexity in implementation has higher overhead in terms of silicon area and energy consumption. Consequently we investigate the applicability of partially adaptive algorithms to achieve a certain degree of fault tolerance in NoC communication fabrics. Turn models are well established partially adaptive routing algorithms used in parallel computing domain [10]. Additionally, random walk is proposed as a suitable fault tolerant routing algorithm for NoC architectures [5]. In this work we evaluate performance of these two fault tolerant algorithms in NoC domain in terms of relevant performance metrics like throughput, energy dissipation, latency and silicon area overhead.

3.1. N random walk

In [5], the authors introduced three different fault tolerant algorithms, viz. probabilistic flooding, directed flooding and N random walk. In this work it is shown that N random walk can provide a better performance than the two flooding algorithms. N random walk allows injection of a fixed number of copies (N) of a message into the network. By using random walk, each node forwards the copies to one of its outgoing channels, and meanwhile makes them follow

non-deterministic paths to destination. The selection of outgoing channel is determined by a set of random probabilities P_N , P_S , P_W and P_E , where the sum of all probabilities is 1. The probabilities are calculated as follows: first, the Manhattan distance between the destination and the current node as well as its neighboring nodes is calculated. The Manhattan Distance for current IP node is given by (1).

$$D_c = |X_{destination} - X_{current}| + |Y_{destination} - Y_{current}| \quad (1)$$

A multiplicative factor M_x is set to 1 for any direction where D_x (x denote either the current or any neighboring node) is

greater than D_c . For the remaining nodes (where $D_x \leq D_c$), the multiplicative factor M_x is equal to $\min(D_x, 4)$. Then the multiplicative factors are normalized to obtain the probabilities P_N , P_S , P_W and P_E . After that, a random number is generated in order to choose one outgoing channel based on the probabilities computed in the previous step. Through the procedure described above, each message is likely to be forwarded towards the same destination, but will take a different path to reach destination [5].

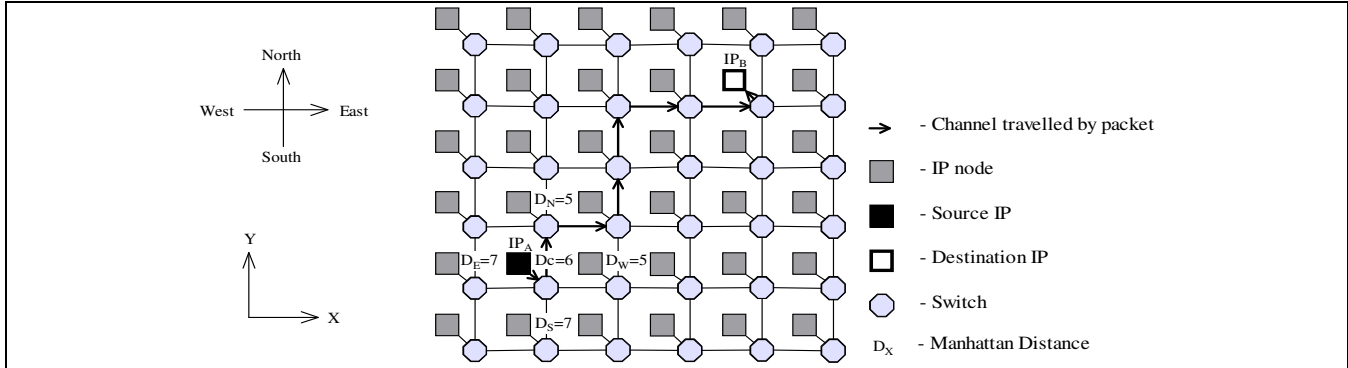


Figure 2: N=1 random walk

A mesh-based network as well as the coordinates is given in Figure 2. The same coordinate system is used throughout the paper. Suppose, source IP_A wants to communicate with the destination IP_B . First, the Manhattan distances D_x of IP_A and its neighboring nodes with respect to IP_B are computed, as shown in Figure 2. Based on D_x , the multiplicative factors are set to 1 for D_E and D_S , and $\min(D_x, 4)$ for D_N and D_W . Then multiplicative factors are normalized to create the probabilities P_N , P_S , P_W and P_E . In this case, the IP_A delivers the message in north direction as the first step because both north and west directions have larger probability than the other directions. Eventually, the message follows the paths shown in Figure 2 from source to destination.

According to [5], though N random walk can sustain a certain level of successful data transfer rate from source to destination node, it dissipates a significant amount of energy because it sends redundant copies of a message into the network.

3.2. Negative first algorithm

The turn models [10] [11] can be used to develop partially adaptive routing algorithms for mesh and torus networks. The turn models make certain turns forbidden so that deadlock can be avoided. These turn models can be applied to handle switch or link failures in a NoC. The *negative-first* algorithm is one of the turn models used in 2-D meshes, in which a packet is routed in the negative direction in each dimensions in the first phase, and then it is routed in the positive direction in the second phase [10]. Specifically, the forwarding message first moves to west or south till the offset is zero and then turns to east or north. The fault-tolerant version of this negative-first algorithm routes adaptively in the negative direction, even further west or south than the destination. The effectiveness of this routing methodology in presence of a fault in the NoC interconnect architecture is explained with the help of Figure 3.

If negative-first routing is adopted then as indicated in Figure 3, the packet is routed adaptively even further south and west than the destination to avoid the faulty switch, and then turn back to north or east to reach the destination. The exception occurs when a packet being routed along the edge of the mesh in the negative direction encounters a faulty switch. As shown in Figure 3b, suppose source IP_C is trying to communicate to the destination IP_D , and the switch S_2 is faulty. In this case the packet is routed one hop perpendicular to the edge, then two hops towards the destination, and one hop back to the edge. The fault-tolerant routing algorithm resulting from the modification of the negative first routing algorithm is summarized as follows.

Let the coordinates of the current node be $(X_{current}, Y_{current})$ and the coordinates of the destination node be (X_{dest}, Y_{dest}) . The distance between the current node and the destination node is expressed as $X_{offset} = X_{dest} - X_{current}$ and $Y_{offset} = Y_{dest} - Y_{current}$.

1. If either $X_{offset} < 0$ or $Y_{offset} < 0$ then route the packet west and south to the destination or further west and south than the destination, avoiding routing the packet to a negative edge for as long as possible. If a faulty node

On the contrary, Figure 5 shows the performance of the NoC with a 5% fault rate by incorporating x-y and negative-first algorithm. It is evident that negative-first outperforms the x-y routing algorithm. It is worth noting that both these algorithms are able to sustain a much better throughput profile than the random walk.

We also compared the performance of the N random walk with x-y and negative-first algorithm in terms of the successful data arrival rate as a relevant metric as suggested in [5]. We considered very low injection load of 0.05%. As shown in Figure 6 negative-first can provide a better performance in presence of faults than x-y routing and N random walk (for $N < 64$). It has almost identical successful data arrival rate in presence of faults as the random walk when $N=64$.

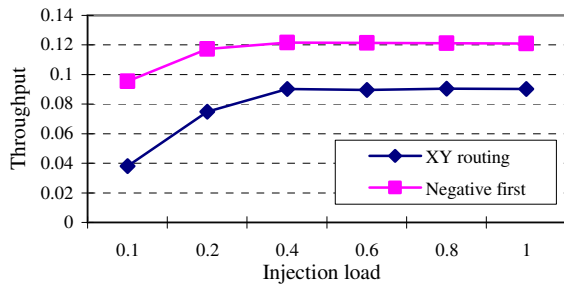


Figure 5: Throughput comparison for negative first and x-y routing algorithm with 5% fault rate

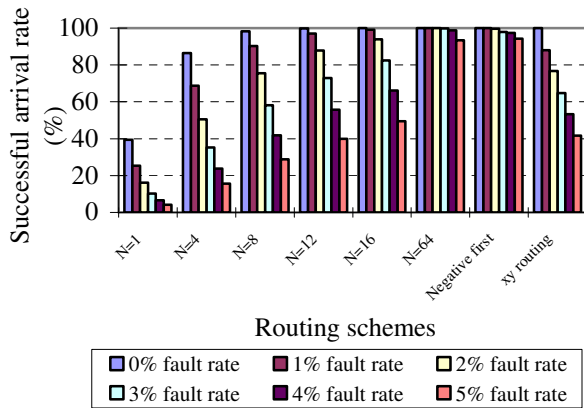


Figure 6: Performance with increasing fault rate

We showed that the negative-first routing algorithm outperforms the x-y routing and N random walk (when $N < 64$) in presence of faults in terms of successful data arrival rate. For a more complete characterization, in the next sub-section we explore the other performance metrics, such as energy dissipation, latency and silicon area overhead.

4.2. Energy dissipation in a NoC-based SoC

When flits travel on the interconnection network, both the inter-switch wires and the logic gates in the switches toggle

and this will result in energy dissipation. The flits from the source nodes need to traverse multiple hops consisting of switches and interconnect segments to reach destinations. Consequently, we determined the energy dissipated in each interconnect and switch hop.

In order to quantify the energy dissipation profile for a NoC interconnect architecture, we determine the energy dissipated in each switch, by running SynopsysTM Prime Power on the gate-level netlist of the switch blocks in the 90 nm technology node. To determine interconnect energy, the capacitance of each interconnect stage is calculated taking into account the specific layout of the topology.

Figure 7 shows the average energy dissipated per hop for different routing algorithms as discussed above. It is evident that the negative-first routing algorithm dissipates almost the same energy as x-y routing. N random walk dissipates much more energy with increasing N, the number of copies of messages in the network. Though the $N=64$ random walk algorithm helps sustaining a higher successful data arrival rate in presence of faults, it comes at the cost of significantly higher energy dissipation.

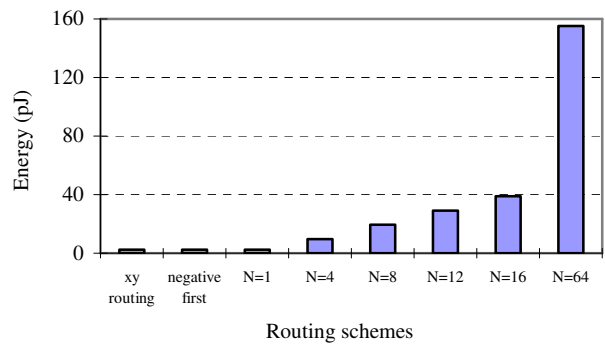


Figure 7: Energy dissipation using different routing schemes

4.3. Latency

Message latency is the time elapsed between the time a message is generated at its source node and the time the message is delivered at its destination node [12]. It is directly related with the average path length, which is given by the number of hops each message is traversing between a pair of source and destination nodes.

Figure 8 shows the average path length for each algorithm in presence of 2% permanent fault, assuming 0.05% injection load. It is evident that negative-first algorithm has the lowest average path length. The average path length for the x-y routing is more than that of negative first. If there is a fault in the path between any particular pair of source and destination nodes, x-y routing can not route the packet successfully. For the random walk algorithm the average path length decreases with increasing N, as the probability of successful arrival increases with it. But even with $N=64$, the average path length is more than that for the

negative first algorithm. This happens due to the fact that random walk does not guarantee forwarding packets in the optimum direction. One point worth noting here is that the injection load was assumed to be very low to be able to get any meaningful results for N-random walk. Consequently this is the best case situation for random walk algorithm. If we increase the injection load then N-random walk will have significantly higher average path length than the negative first algorithm.

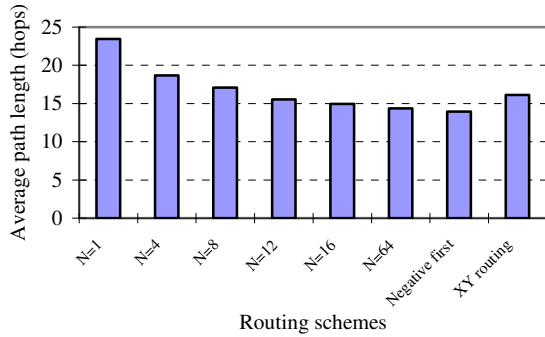


Figure 8: Average path length

4.4. Area overhead

We designed and synthesized the routing blocks incorporating deterministic x-y routing, N random walk, and negative-first routing and synthesized using 90 nm standard cell libraries with Synopsys Design Compiler. We express the silicon area overhead required for a single switch for the different routing schemes discussed in terms of equivalent two-input NAND gates in Table 1.

Table 1: Silicon area overhead of the routing schemes

Routing Method	Silicon Area (2-input NAND gates)
x-y routing	250
N=1 random walk	11160
Negative first	333

From the table, it is evident that x-y routing and negative-first have very comparable area overhead while N random walk needs much more area even for N=1.

5. Conclusions

For deep sub-micron VLSI processes, the life-time reliability of devices is likely to be compromised by effects such as electromigration and material ageing. Consequently, the performance of NoC interconnect architectures will be severely affected due to presence of permanent faults. Though deterministic routing is very easy to implement, it fails to sustain the desired level of performance in presence of permanent faults. Adaptive algorithms need to be incorporated in NoC architectures. In this paper we have

demonstrated that simple partial adaptive routing algorithms, like the turn models, can be effective in sustaining a certain performance level in presence of faults. By adopting the negative-first routing algorithm (one of the popular turn models), we can achieve higher throughput and lower latency in presence of faults compared to a stochastic algorithm like random walk. Moreover, by adopting the negative-first routing algorithm we can effectively dissipate less energy and incur less silicon area overhead compared to stochastic algorithms.

6. References

- [1] L. Benini, G. De Micheli, "Networks on Chips: A New SoC Paradigm", *Computer*, Volume: 35 Issue: 1, Jan. 2002. pp: 70-78.
- [2] P. Pande, C. Grecu, A. Ivanov, R. Saleh, G. De Micheli, "Design, Synthesis and Test of Networks on Chip", *IEEE Design and Test of Computers*, Vol. 22, No. 5, 2005. pp: 404-413.
- [3] D. A. Hodges, H. G. Jackson and R. A. Saleh, "Analysis and Design of Digital Integrated Circuit: In Deep Submicron Technology", Third Edition, McGraw-Hill, 2003.
- [4] T. Dumitras, S. Kerner, R. Marculescu, "Towards on-chip fault-tolerant communication", Proceedings of the 40th Design Automation Conference DAC 2003, 21-24 Jan, 2003. pp: 225 – 232.
- [5] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, "Fault tolerant algorithms for network-on-chip interconnect", IEEE Computer society Annual Symposium on VLSI, 2004. 19-20 Feb. 2004. pp: 46 – 51.
- [6] J. Zhou, F. C. M. Lau, "Fault-tolerant wormhole routing in 2D meshes", International Symposium on Parallel Architectures, Algorithms and Networks, 2000. 7-9 Dec. 2000 pp: 94 – 101.
- [7] J. Zhou, F. C. M. Lau, "Adaptive fault-tolerant wormhole routing in 2D meshes", 15th International Parallel and Distributed Processing Symposium, 23-27 April 2001.
- [8] Jipeng Zhou, F. C. M. Lau, "Adaptive fault-tolerant wormhole routing with two virtual channels in 2D meshes", 7th International Symposium on Parallel Architectures, Algorithms and Networks, 2004. 10-12 May 2004. pp: 142 – 148.
- [9] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, C. R. Das, "A Low Latency Router Supporting Adaptivity for On-Chip Interconnects", Proceedings of the 42nd Design Automation Conference DAC 2005. pp: 559-564.
- [10] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing", Proceedings of the 19th Annual International Symposium on Computer Architecture, May 1992. pp: 278-287.
- [11] C.J. Glass and L.M. Ni, "Adaptive Routing in Mesh-connected Networks", Proceedings of the 12th International Conference on Distributed Computing Systems, June 1992. pp: 12-19.
- [12] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, "Performance Evaluation and Design Trade-offs for Network on Chip Interconnect Architectures", *IEEE Transactions on Computers*, vol. 54, no. 8, August 2005. pp: 1025-1040.
- [13] Ge-Ming Chiu, "The odd-even turn model for adaptive routing", *IEEE Transactions on Parallel and Distributed Systems*, Volume 11, Issue 7, July 2000. pp:729 – 738.
- [14] J. Hu, R. Marculescu, "DyAD - smart routing for networks-on-chip", Proceeding of 41st Design Automation Conference, DAC 2004, pp: 260 – 263.