Minimal Path Fault Tolerant Routing Algorithm and its Performance Evaluation

Abir J Mondal
Department of Electronics and Computer Engineering
NIT Arunachal Pradesh
Yupia, India
abir_jm@hotmail.com

Pinaki Chakraborty
Department of Basic and Applied Science
NIT Arunachal Pradesh
Yupia, India
pinakichk@gmail.com

***Abstract*— There has been increasing demand for Networks on Chip (NoC) with academia and industry, but it is threatened by decreasing reliability of aggressively scaled transistors. In many core embedded systems the reliability issue associated with on chip communication is a crucial factor. The emergence of NoC paradigm is to address the said issue. The use of traditional fault tolerant routing algorithms to reroute packets around faulty regions will increase the packet latency and create congestion around the faulty region. This results in an increase in time to traverse a packet from a source to a destination, so it should be avoided. In this paper, a unique fault tolerant approach is described that is able to route packets through shortest paths in the presence of faulty nodes as long as a path exists. To avoid congestion, next node can be adaptively chosen when the distance from the current to destination node is equal in both directions. In addition, an analytical model is presented to evaluate the performance in case of single and multiple faulty nodes.**

Keywords—NoC; fault tolerance; routing algorithm; minimal; non-minimal

I. Introduction

According to Moore's law billions of transistors could be integrated on a single chip in the near future [1]. In these chips, hundreds of functional intellectual property (IP) blocks and embedded memory modules could be placed together to form a Multi-Processor Systems-on-Chip (MPSoCs). The notion of integrating numerous components into a single chip has led to the reduction of many portable devices and a rise in their computational capabilities. By incrementing the number of processing elements on a single chip, traditional bus based architectures in MPSoCs are not useful any longer and a new communication infrastructure is needed. In this context, Network-on-Chip (NoC) [2][3] is emerging as a promising design paradigm to replace traditional bus based systems.

In deep sub-micron (DSM) VLSI processes, it is challenging to guarantee correct fabrication with an acceptable output without exploiting design techniques that take into account manufacturing defects [4-6]. To improve the reliability of multi-core SoCs, their interconnect framework must be designed such that the fabrication faults can be sustained. These irrecoverable faults influence the behavior of NoC fabrics and consequently demean the system performance. Consequently, achieving on-chip fault tolerant communication is becoming increasingly important in presence of such permanent faults.

Initially deterministic routing algorithms [7] were employed due to the ease of implementation. The constraint of implementing deterministic routing is that it establishes a fixed path from 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 nodes if adaptive routing algorithms [8-10] are adopted. One of the characteristics of adaptive routing algorithms is the ability to establish alternate routing paths in presence of faults.

In NoC communication framework the option of adaptive routing method is further explored so as to avoid faulty nodes 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 faulty nodes, it includes non-minimal paths under certain fault distribution, thereby increasing the time to destination. Further, under certain fault distribution the adaptivityness causes rerouting of packet and in some situation packet drop also. So, there is a need to develop fault tolerant routing algorithm that figures out minimal path under any fault distribution if it exists. The purpose of this paper is to find out a fault tolerant routing algorithm under any fault distribution situations and to evaluate its performance such that it always results in minimal paths.

II. Related Work

Fault tolerant routing algorithms can be separated into two groups: one that uses convex or concave regions [11-14] and the other utilize contour strategy for addressing faults [15-16]. It can also be classified into two classes: the methods using virtual channels [16-18] and those without using virtual channels [19-20]. It is also possible to implement routing algorithms as either table-based or in algorithmic form [21-22]. In algorithmic routing mechanism, an algorithm is executed using hardware circuits using FSM to compute appropriate router port. It is generally suitable for one topology. Table based mechanism is used to deal with regular as well as irregular topologies. The table based methods cannot scale well since the table size increases with the size of network and may become impractical. In the application specific platforms where communication transactions among IP cores are known in advance, it is quite possible to use compression techniques [21-23] to reduce the size of tables instead of straight forward table based implementation. In [24], authors discussed efficient implementation of distributed routing algorithms for partial 2D meshes without using routing tables. Most of the fault tolerant routings use either virtual channels [18] or turn models [25] based strategies to achieve deadlock freedom. In this paper, we present a reconfigurable, deadlock free, cost efficient routing algorithm without using virtual channels.

III. Proposed Methodology

The common characteristic of NoC architectures is that the constituent IP cores communicate with other through switches. Generally wormhole routing is adopted [17]. One of the most widely used NoC topologies is the Mesh architecture. The x-y algorithm is a simple deterministic routing methodology used in Mesh networks [17-18]. 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 faulty nodes. Fully adaptive fault tolerant routing algorithms [19][21] perform better in presence of faulty nodes, but due to complexity in implementation has higher overhead in terms of silicon area. Turn models are well established partially adaptive routing algorithms used in parallel computing domain [22]. But under certain fault distribution situations, fully adaptive as well as partially adaptive routing algorithms either result in non-minimal path or in worst case packet-drop. Consequently, we investigate the applicability of fault tolerant routing algorithm under any fault distribution scenario and to evaluate performance so as to maintain minimal path if it exists.

A. Fault Information Distribution Method

As shown in Fig. 1, fault information is distributed in such a way that each router is informed about the faulty nodes of its direct neighboring routers. For this purpose, each router transfers faulty information to the neighbors. If E, W, N and S stand for the packet direction in the east, west, north and south directions respectively, then each router has the information about the following nodes: E, W, N and S. For routing a packet in the northeast direction, a router uses the information on the nodes N and E. Similarly, for a northward packet, the fault's information on the nodes N, E and W is beneficial for making a reliable routing. Using this information, packets are possibly routed through either minimal or non-minimal paths, but rerouting is avoided.

We have assumed a mesh network where the locations of current and destination nodes are pre-defined and the neighboring nodes of the current node are explored to figure

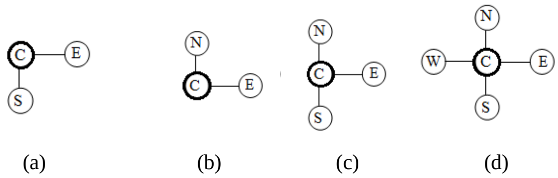


Fig. 1. *One hops fault distribution methodology*

out busy nodes. The busy nodes are assumed taking into consideration the failure of partially adaptive algorithm and contour strategy. We model faults as busy nodes which mean either the node is in communication with other processing element or act as an intermediate node to an existing process. At first, the current and destination addresses are determined. Then the neighbors of current node are checked to determine whether they are busy or not. The first step is to determine X_{offset} and $Y_{offset.}$ If X_{offset} is less than zero, the following conditions are verified to determine next node:

a) $if\left(\left(X_{busy} \neq \left(X_n - 1\right)\right) or \left(Y_{busy} \neq Y_n\right)\right)$

b) $if\left(\left(Y_{busy} \neq \left(Y_n + 1\right)\right) or \left(X_{busy} \neq X_n\right)\right)$

c) $if\left(\left(Y_{busy} \neq \left(Y_n - 1\right)\right) or \left(X_{busy} \neq X_n\right)\right)$

In the same manner a set of conditions are derived to determine next node if $X_{offset} > 0$, $X_{offset} = 0$ and $Y_{offset} < 0$, and $X_{offset} = 0$ and $Y_{offset} > 0$. Each of the cases is analyzed as shown above to figure out whether the next node is a busy node or the direction towards which packet shall be forwarded. The fault distribution technique describing the one hops scenario is expressed in algorithmic form as shown in Fig. 2.

B. Improved Fault Information Distribution Method

An improved fault distribution methodology is now described which not only avoids non-minimal paths but always chooses minimal path if it exists. Moreover, rerouting is also avoided. The unique methodology is obtained by improving the fault distribution described in section A. It was observed that if the two hop faulty nodes information are available to the current router, non-minimal paths can always be avoided. The fault information is shared in such a way that each router is informed about the fault condition in its immediate neighbor and also at multiple hops through its neighbor. Using this information, unnecessary paths are avoided to prevent packet drop in case destination is unreachable and deadlock. Fig. 3 depict the proposed fault distribution methodology. The current router is aware of the faulty nodes in one hop distance. E, W, N and S stand for the East, West, North and South directions. In Fig. 3(a), the neighboring nodes share their condition with current node (C). In Fig. 3(b), the node in the East direction is aware of the faulty nodes in its neighbor (NE, EE and SE). Similarly, in Fig. 3(c), the neighboring nodes (SE, SS, SW) share their condition with South directed node. Finally, in Fig. 3(d), the node in the West direction has knowledge about its neighboring nodes (SW, WW and NW). In this manner the current node is not only aware about its immediate neighbor,

```
Algorithm: Random Routing for 2-D Meshes
Inputs: Coordinate of current node (Xcurrent , Ycurrent)
        Destination node (Xdest , Ydest ).
Output: Selected output Next_Node (Xn,Yn)
Definition : ngbr : neighboring node.
             Node: router
Procedure:

     Xoffset =Xdest – Xcurrent;
     Yoffset =Ydest – Ycurrent;
//check the neighbor node, if busy then the co-ordinate
//of that node will be (Xbusy,Ybusy)
if Xoffset < 0 then
     {
     If (ngbr== healthy) then
     Select=X- (South)
     else
     Select= East or West or North; //select preferably if nodes are healthy
     }
     endif
if Xoffset > 0 then
     {
     If (ngbr== healthy) then
     Select=X+ (North)
     else
     Select= East or West or South; //select preferably if nodes are healthy
     }
     endif
if Xoffset = 0 and Yoffset < 0 then
     {
     If (ngbr== healthy) then
     Select=Y- (West)
     else
     Select= North or South or East; //select preferably if nodes are healthy
     }
     endif
if Xoffset = 0 and Yoffset > 0 then
     {
     If (ngbr== healthy) then
     Select=Y- (East)
     else
     Select= North or South or East; //select preferably if nodes are healthy
     }
     endif
if Xoffset = 0 and Yoffset = 0 then
        Select = Internal;
     endif
```

Fig. 2. Algorithm illustrating one hops visibility

but also possesses information about nodes in two hop distances through its immediate neighbor. For routing a packet in the northeast direction, the router uses information about faulty nodes from its immediate neighboring nodes in the north and east directions because they are aware about the condition of their neighbors (NE). Similarly, for a southwest packet, the information on south and west directed nodes is beneficial for making a reliable routing. Using this information, packets are routed through minimal paths which avoids making unnecessary routing around faulty nodes. Further the improved fault distribution methodology is expressed in algorithmic form as illustrated in *Fig. 4*.

C. Comparison of Fault Distribution Methodologies

In Fig. 5 we have considered a 2D mesh and the performance of the proposed one hops fault tolerant routing algorithm and its improved version are compared while traversing packet from current node to destination node. The current and destination nodes are marked with C and D respectively. Both the algorithms select next node based on fault information. Moreover, in Fig. 5 packet is sent from current node to destination node with nodes marked cross

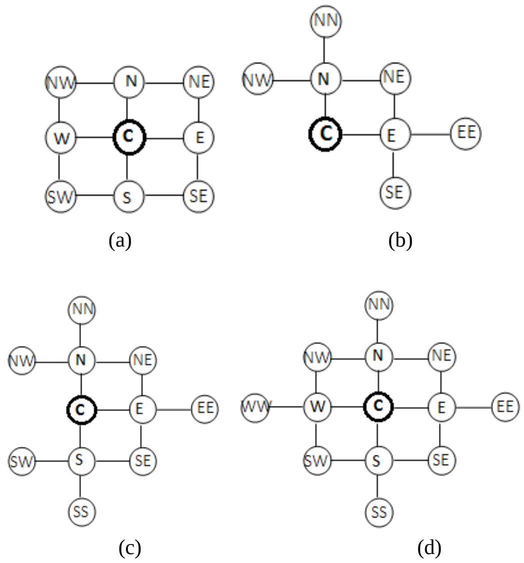


Fig. 3. Improved fault distribution methodology

denote faults. In Fig. 5a, the packet has equal probability to traverse either to north or east directions. Using one hops fault tolerant routing the packet is first directed towards node N. On reaching node N it has again equal probability to travel either north or east. If east node, NE is non faulty the packet is directed towards NE. Further, the node, NE is informed about its neighbors NNE and NEE. If either one is busy, the packet is routed through the other path. If north bound node, NNE is non-faulty, packet is routed north. Again, the node NNE is aware about its neighbor and routed packet towards node NNNE because the other neighbor is faulty. While reaching node NNNE, packet has to traverse along the dotted path so as to reach destination, because of a faulty node in its neighbor. Similarly, if the packet was initially traversed along east node, it was observed that using the one hops fault tolerant routing algorithms the path to destinations turns out to be a non-minimal path.

The non-minimal path traversed using the one hops fault tolerant algorithm can be avoided using the improved fault tolerant methodology described above. Using the improved technique, the current router is not only informed about its immediate neighbor but also at two hops distances using the immediate neighbor. The node C has equal probability to travel either north or east. If the node N is non-faulty, packet is directed towards N. Again, the node N is aware about its neighbor, NN and NE. If the node NE is non-faulty, packet is directed towards node NE. Further, the node NE is also informed about its neighbors. Using this information packet is routed to destination using the only minimal path available. Similar arguments are possible for the mesh shown in Fig. 5b. It was observed that if the one hops fault tolerant routing algorithm is used to traverse packet from node C to D, the non-minimal path along the dotted line is availed to reach destination. But the improved fault distribution technique results in the only available minimal path.

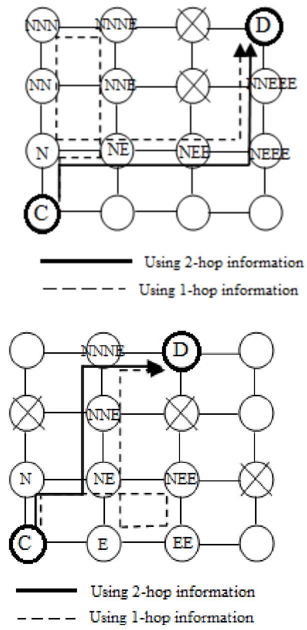
```
Algorithm: Proposed Routing for 2-D Meshes
Inputs: Coordinate of current node (X_current , Y_current)
        Destination node (X_dest , Y_dest ).
Output: Selected output Next_Node (X_n,Y_n)
Definition : ngbr : neighboring node.
             Node: router
Procedure:
    X_offset =X_dest – X_current;
    Y_offset =Y_dest – Y_current;
//check the neighbor node, if faulty then the co-ordinate
//of that node will be (X_faulty,Y_faulty )
if X_offset < 0 then
    {
    If (ngbr== healthy) then
    Select=X- (South)
    else
    Select= E,ES,W,WS,EE,WW,N; //select preferably if nodes are healthy
    }
    endif
if X_offset > 0 and Y_offset < 0 then
    {
    If (ngbr== healthy) then
    Select=(X+,Y-) (North, West)
    else
    Select= W,WN,WW,E,EN,S,SW,EE,SS ; //select preferably if nodes are healt
    }
    endif
if X_offset > 0 and Y_offset > 0 then
   {
    If (ngbr== healthy) then
    Select= (X+,Y-) (North, East)
    else
    Select= E,EN,EE,W,WN,WW,S,SE,SS; //select preferably if nodes are health
    }
    endif
if X_offset > 0 and Y_offset = 0 then
    {
    If (ngbr== healthy) then
    Select=X+ (North)
    else
    Select= E,EN,W,WN,EE,WW,S; //select preferably if nodes are healthy
    }
    endif
if X_offset = 0 and Y_offset < 0 then
    {
    If (ngbr== healthy) then
    Select=Y- (West)
    else
    Select= N,NW,S,SW,NN,SS,E; //select preferably if nodes are healthy
    }
    endif
if X_offset = 0 and Y_offset > 0 then
    {
    If (ngbr== healthy) then
    Select=Y+ (East)
    else
    Select= N,NE,S,SE,NN,SS,W; //select preferably if nodes are healthy
    }
    endif
if X_offset = 0 and Y_offset = 0 then
    Select = Internal;
    endif
```

Fig. 4. Algorithm illustrating two hops visibility

Further to understand the reliability of the 2 hops routing algorithm, faulty nodes are adaptively chosen and the performance is verified in traversing a packet to destination. The reliability is verified both by increasing the mesh size and faulty nodes. It is observed from Fig. 6 that the 2 hops fault tolerant routing algorithm succeeds in traversing packet even when mesh size increases along with fault distribution.

IV. Results

The one hops fault tolerant routing algorithm and its improved version is evaluated in terms of complexity, path length and reliability. The term complexity refers to number of comparisons required by routing algorithm to reach destination. To determine complexity we have assumed two cases; mesh without busy nodes and with busy



(a) (b)

Fig. 5. *Mesh showing comparisons of one hops and two hops routing algorithm*

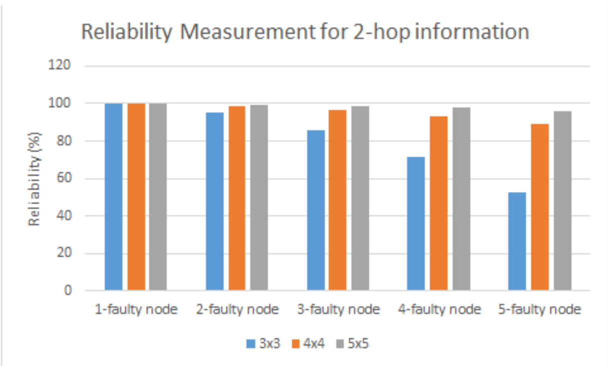


Fig. 6. *Reliability of improved fault tolerant routing algorithm*

nodes. We have considered a 2D mesh with predefined current and destination locations. The performance of routing algorithms is measured as the number of nodes between source and destination are increased. It is observed from Fig. 7 that the one hops fault tolerant routing algorithm has better complexity than its improved version and the traditional algorithms. This is because as the number of nodes between current and destination are increased, the one hops algorithm simply checks its neighbor and makes routing decisions. The improved version has to maintain the described fault distribution so as to figure out next node.

But the scenario is changed as the busy nodes are introduced into the mesh. It is observed from Fig. 8 that the inclusion of busy nodes has resulted in better performance in case of improved fault tolerant routing algorithm. In addition, the busy nodes location has made it impossible for turn models to deliver packet to destination. Using the improved fault distribution

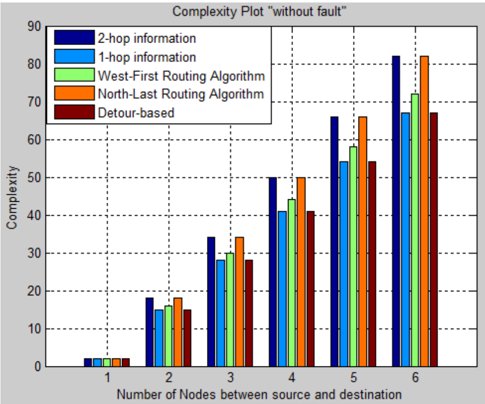


Fig. 7. *Plot of Complexity vs. Number of Nodes between Source and Destination*

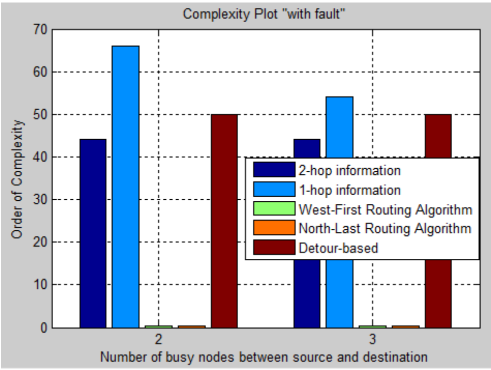


Fig. 8. *Plot of Complexity vs. Number of Busy Nodes between Source and Destination*

technique the current node is always aware about nodes at two hops distances. Consequently, it can easily figure out the faulty node locations and make appropriate decisions to find out the next node. In this manner, the only available minimal path is always chosen thereby avoiding non-minimal path.

The performance of the fault tolerant routing algorithm is evaluated in terms of path length. By path length we mean the total distance traversed to reach destination. For the 2D mesh we derived path weights using breadth first search. Once the path weights are available we have estimated the path length from current to destination taking into account the presence of faulty nodes. The faulty nodes are changed randomly and the corresponding path length is obtained. Initially, two busy nodes are considered and their locations are varied. The corresponding plot of path length is given in Fig. 9. Next three busy nodes are taken and their positions are varied to obtain desired path length. It is observed from Fig. 9 that for a particular position of faulty nodes the two hops algorithm succeeded in traversing packet to destination using minimal path. The turn model failed to reach destination, whereas the one hops algorithm results in non-minimal path.

Lastly, the performance of the one hops and its improved version is validated by increasing the mesh size and faulty nodes. Fig. 9 illustrates the reliability scenario as the faulty nodes are progressively increased in a 3x3 mesh. The reliability of the network

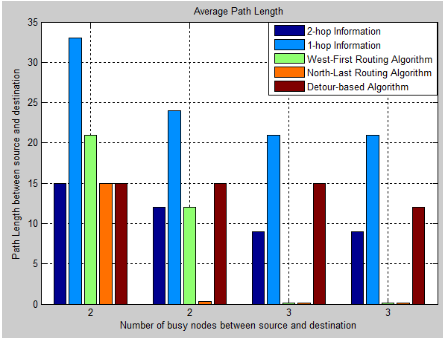


Fig. 9. *Plot of path length for different positions of faulty nodes*

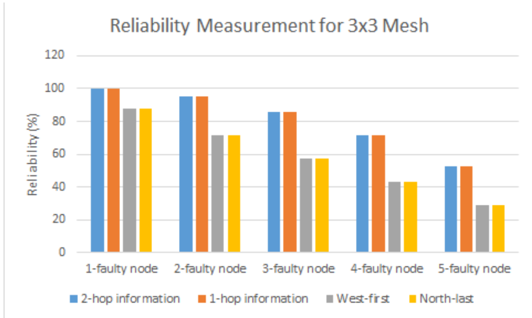


Fig. 10. *Reliability analysis for 3x3 mesh in presence of faulty nodes*

is 100% if one node is faulty. When the number of faulty nodes is 2, there are 7 possible locations for faults to be present. So the number of two faulty nodes combination is 21. Out of these 21 combinations network fails in only one condition. Consequently, reliability of the one hop and two hops routing algorithm is 95.24%. Similarly, when the number of faulty nodes is 3, there are 4 possible locations for faults to be available. Further, there are 35 possible combinations for 3 faulty nodes to be present and from there 5 combinations cause network failure. So the reliability turns out to be 85.71%. When the numbers of faulty nodes are 4 and 5, there are 4 and 5 available locations for faulty nodes to be present. So the total numbers of possible combinations are 35 and 21 respectively. From these combinations 10 combinations cause network failure. Consequently, the reliability is 71.4% and 52.4% respectively.

Similarly, in the case of 4x4 and 5x5 mesh the available locations for faulty nodes are first determined and then the possible combinations are figured out. From there, the combinations resulting in network failure are obtained and finally the reliability percentage is calculated. Fig. 11-12 depicts the reliability plot for both 4x4 and 5x5 mesh. Compared to turn model, one hop and its improved version is more reliable even if the mesh size and faulty nodes are increased.

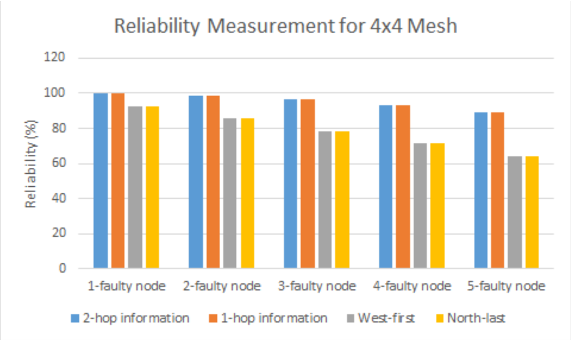


Fig. 11. *Reliability analysis for 4x4 mesh in presence of faulty nodes*

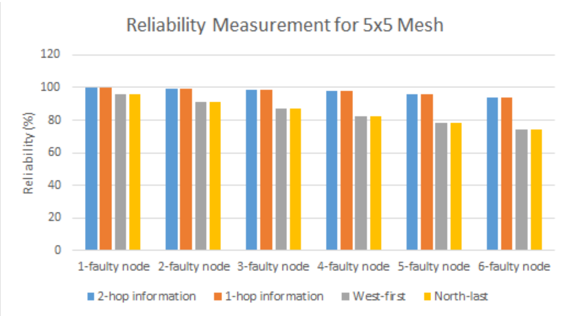


Fig. 12. *Reliability analysis for 5x5 mesh in presence of faulty nodes*

V. Conclusion

The one hop fault distribution scenario is a fault tolerant algorithm, but results in non-minimal path. While analyzing the one hop fault distribution technique, a unique fault distribution methodology is proposed that always result in the only minimal available path, if it exists. The introduction of busy nodes has shown that two hops fault distribution scenario have better complexity and path length compared to its initial version. But, both the one hop and its improved version turn out to be reliable even when mesh size was increased along with faulty nodes. Finally, we are working on router micro-architecture to verify the performance of the proposed routing algorithms in terms of area and power metrics.

References

[1] Xu, Jing et al. ,2005, "A methodology for design, modeling and analysis for network-on-chip, " in Proc. IEEE International Symposium on Circuits and Systems, pp.1778-1781.
[2] W. Tsai et al. , 2011, "A fault-tolerant NoC scheme using bidirectional channel", in Proc. DAC, pp.918-923.
[3] E. Rijpkema et al. , 2003, "Trade offs in the design of a router with both guarantee and best-effort services for networks on chip", in Proc. DATE'03, pp. 350-355.
[4] M. Cuviello et al. , 1991, "Fault modeling and simulation for crosstalk in system-on-chip interconnects", in Proc. ICCAD, pp.297-303.
[5] M.H Neishaburi et al. , 2007, "HW/SW architecture for soft-eError calculation in real-time operating system", Journal of the Institute of Electronics, Information and Communication Engineers, Vol. 4, No. 23, pp. 755-761.
[6] J. Duato et al. , 2003, "Interconnectin networks: an engineering approach", Morgan Kaufmann publishers.
[7] L.M. Ni et al. , 1993, "A surway of wormhole routing techniques in direct networks", in Proc. IEEE Computer , V. 26, I.2, pp. 2-7.
[8] D. Fick et al. , 2009, "Vicis: a reliable network for unreliable silicon", in Proc. of Design Automation Conference, pp. 812-81.
[9] S. Chalasani et al. , 1995, "Fault-tolerant worhmhole routing algorithms for mesh networks", IEEE Trans on Computers, 44(7):848-64.
[10] PH. Sui et al. , 2011, "An improved algorithm for fault-tolerant wormhle routing in meshes", IEEE Trans on Computers x;46(9): 1040-2.
[11] S. Park et al. , 2008, "Fault-tolerant wormhole routing algorithms in meshes in the presence of concave faults", in roc. of International Parallel and Distributed Processing Symposium (IPDPS), pp. 441-446.
[12] Z. Zhang et al. ,2008, "A reconfigurable routing algorithm for fault-tolerant 2D-mesh Network-on-chip", in Proc. DAC, pp. 441-446.
[13] M. Valinataja et al. , 2011, "A reconfigurable and adaptive routing method for fault-tolerant mesh-based Network-on-Chip", in Proc. International Journal of Electronics and communications (AEU), V. 65, I.7, pp. 630-640.
[14] M. Koibuchi et al. , 2008, "A lightweight fault-tolerant mechanism for Network-on-Chip", in Proc. NOCS, pp. 13-22.
[15] J. Wu, 2003, "A fault-tolerant and deadlock-free routing protocol in 2D meshs based on dd-even turn model", IEEE Transation on computers, V. 52, pp. 1154-119.
[16] D. Fick et al. , 2009, "A highly resilient routing algorithm for fault-tolerant NoCs", in Proc. DATE, pp. 21-26.
[17] W. J. Dally and B. P. Toweles, 2004, "Principles and Practices of Interconnection Netwrks", Morgan Kaufmann.
[18] J. Duato, S. Yalamanchili, and L. Ni, 2003, "Intercnnection Networks – An Engineering Approach", Morgan Kaufmann.
[19] M. Palesi, R. Holsmark, S. Kumar, and V. Catania, 2009 ,"Application specific routing algorithms for network-on-chip", Prallel and Distributed Systems, IEEE Transactions on, vol. 20, no. 3, pp. 316-330.
[20] J. Flich and J. Duato, 2008, "Logic-based distributed routing for NoCs", IEEE Computer Architecture Letter, Vol. 7, No. 1, pp. 13-16.
[21] C. Glass and L. Ni, 1992, "The turn model for adaptive routing", in Proc. 19th Internatinal Symposium on Computer Architecture, pp. 278-287.
[22] Z . Zhang, A. Grener, and S. Taktak ,2008, "A reconfigurable routing algorithm for fault-tolerant 2D-mesh Network-on-chip", in Proc. 45th DAC, pp. 441-446.
[23] G. M. Chiu, 2000, "The odd-even turn model for adaptive routing", IEEE Transactions on, vol. 11, no. 7, pp. 729-738.

[24] B. Fu, Y. Han, J. Ma, H. Li, and X. Li, 2011, “An abacus turn model for time/space-efficient reconfigurable routing”, in Proc. 38th International Symposium on Comuter Architecture, pp. 249-270.

[25] C. J. Glass and L. Ni, 1993, “Fault-tolerant wormhole routing in meshes”, in Proc. 23rd Internatinal Symposium on Fault-tolerant Computing, pp. 240-249.