Fault-Tolerant Routing Algorithm for 3D NoC Using Hamiltonian Path Strategy

Masoumeh Ebrahimi, Masoud Daneshtalab, Juha Plosila
Department of Information Technology, University of Turku
{masebr, masdan, juplos}@utu.fi

Abstract— **While Networks-on-Chip (NoC) have been increasing in popularity with industry and academia, it is threatened by the decreasing reliability of aggressively scaled transistors. In this paper, we address the problem of faulty elements by the means of routing algorithms. Commonly, fault-tolerant algorithms are complex due to supporting different fault models while preventing deadlock. When moving from 2D to 3D network, the complexity increases significantly due to the possibility of creating cycles within and between layers. In this paper, we take advantages of the Hamiltonian path to tolerate faults in the network. The presented approach is not only very simple but also able to support almost all one-faulty unidirectional links in 2D and 3D NoCs.**

I. INTRODUCTION

Networks-on-Chip (NoCs) has become a promising solution for on-chip interconnection in many-core systems-on-chip due to its reusability and scalability [1][2]. Routing techniques provide some degrees of fault tolerance in NoCs because of the alternative paths between nodes in the network. On-chip interconnects implemented with deep submicron semiconductor technology, running at GHz clock frequencies are prone to failures [1][3]. Due to this extreme device scaling, the likelihood of failures increases [5][6].

Faults can be discussed at link level [4] or router level [7][8]. The fault at a router level means that a router is counted as faulty when either a single component or entire router fails. Similarly, the faulty links can be discussed in a general form of bidirectional faulty links in which a link is counted as faulty if only one unidirectional link fails or both unidirectional links stop working. In this paper, we take into consideration the special case of unidirectional faulty links. Based on this assumption, we propose an algorithm which is able to support almost all unidirectional faulty links in 3D mesh network. We utilize the concept of the Hamiltonian path to tolerate faulty links in the network. We called this method HamFA, the Hamiltonian-based Fault-tolerant routing Algorithm.

Implementing deadlock free fault-tolerant algorithms are usually complicated in a 2D mesh network while the complexity increases with 3D mesh network. This is due to the fact that in a 2D mesh network, the algorithms were concerned with two abstract cycles in a XY plane while in 3D network the cycles should be prevented within each layer (i.e XY, XZ, and YZ) and between layers. There are few fault-tolerant algorithms investigated in 3D mesh network. The 4NP-First routing algorithm [9] utilizes two separate virtual channels, one dedicated to 4N-First routing algorithm and the other to the 4P-First routing algorithm. The 4N-First and 4P-First routing algorithms are resulted from the straight forward extension of the negative-first turn model from 2D to 3D NoCs. In some cases, 4NP method sends two packets for each pair of source and destination node through different routes. In sum, this algorithm is able to support all one-faulty cases using two separate virtual channels and routing algorithms. Our proposed method supports almost all one-faulty links either in horizontal or vertical links without using any virtual channels or packet redundancy. AFRA [10] is another algorithm presented for 3D mesh network. This algorithm supports single faults only in TSV links, whereas faults might has occurred in horizontal links as well. In addition, every node has to know the faulty information on all vertical links of the same row. For instance, in a 4×4×4 mesh network, the information on 12 links is required at each node. Besides the chance of occurring faults on these links, the fault distribution mechanism imposes an additional overhead on the system. Our proposed method does not require any extra link for distributing fault information and hence no distribution mechanism is needed. In this paper, we also look for the simplicity of the fault-tolerant approach to 3D NoCs. To obtain this goal, we use the Hamiltonian path for tolerating faults in the network. It is worth mentioning that this method only requires knowing the fault information on its adjacent links which is already available in any NoC architecture. Moreover, HamFA does not require any routing table at routers or carrying additional information in the packet header.

This paper is organized as follows. In Section II, the Hamiltonian Path is introduced while the proposed approach is presented in Section III. The results are explored in Section IV and the paper is concluded in the last Section.

II. HAMILTONIAN PATH

The Hamiltonian path is used for NoCs to support collective communications [11][12] when a packet has several destinations to be delivered. The Hamiltonian path strategy [13] guarantees that the network will be remained free of deadlocks when routing packets. The Hamiltonian path visits each node exactly once along the path. As shown in Fig. 1(a), for each node a label is assigned from 0 to N-1 in which N is the number of nodes in the network. Several Hamiltonian paths can be considered in the mesh topology. In 3D a×b×c mesh (symmetric mesh), each node is presented by a ordered triple (x,y,z). The following equations show one possibility of assigning the labels which we utilize in this paper:

$$L(x,y,z)=\{(a\times b\times z)+(a\times y)+(x)\} \qquad where\, z: even, y: even$$

$$L(x,y,z)=\{(a\times b\times z)+(a\times y)+(a-x-1)\} \qquad where\, z: even, y: odd$$

$$L(x,y,z)=\{(a\times b\times z)+(a\times(b-y-1))+(a-x-1)\} \qquad where\, z: odd, y: even$$

$$L(x,y,z)=\{(a\times b\times z)+(a\times(b-y-1))+(x)\} \qquad where\, z: odd, y: odd$$

As exhibited in Fig. 1, two directed Hamiltonian paths (or two sub-networks) are constructed by the labeling. The high channel sub-network starts at node 0 (Fig. 1(b)), and the low channel sub-network ends at node 0 (Fig. 1(c)). In a case when the label of the destination node is greater than the label of the source node, the routing takes place in the high channel sub-network; otherwise it takes place in the low channel sub-network. In order to guarantee the minimal paths for each couple of source and destination nodes, the channels that are not part of the Hamiltonian path (the dashed lines in Fig. 1) could be used in appropriate directions. These channels are called short-cut channels.

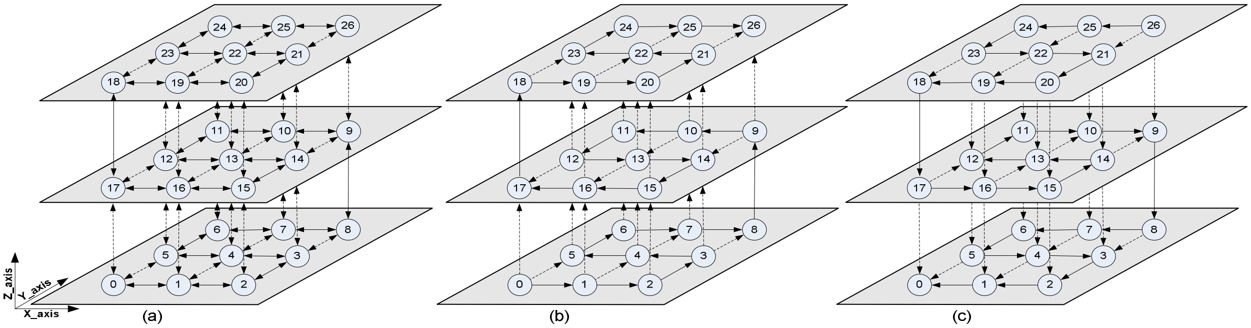

Fig. 1. (a) A 3×3×3 mesh physical network with the label assignment (b) high channel and (c) low channel sub-networks. The solid lines indicate the Hamiltonian path and dashed lines indicate short-cut channels that could be used to reduce the path length.

To show that the proposed algorithm is deadlock free, we need to show that the channel dependency graph (CDG) is acyclic. Since in the high channel sub-network, all packets pass through switches only in ascending order, no cycle can be created in the CDG graph. Similar perspective can be applied to the low channel sub-network [14]. Moreover, high and low channel subnetworks are disjoint and thus packets moving in each subnetwork utilize different sets of resources. Therefore, there is no possibility of deadlock as long as packets are strictly belonging to either subnetwork.

III. THE PROPOSED APPROACH

A. *Hamiltonian-based Fault-Tolerant Method for 2D Mesh*

Using Hamiltonian path and short-cut links, obviously, there is always a minimal path between each source and destination node within a layer. When source and destination nodes are located in different layers and there is no minimal option in a current layer, for sure, it exists in the next layer. The reason is that the direction of the Hamiltonian path in two adjacent layers is reversed. For example, as can be obtained from Fig. 1, packets should be routed through the north and east directions from the node 0 to node 8 in the first layer while the south and west directions are taken to deliver packets from the node 9 to node 17 in the second layer. To clarify, we follow the paths of packets travelling from the source node 4 to different destinations in Fig. 1 to show the availability of at least one minimal path in all situations.

Path {4,9}={4,7,8,9}

Path {4,18}={4,5,12,17,18} or {4,13,16,17,18}

Path {4,24}={4,5,12,23,24} or {4,7,10,11,24} or {4,13,22,23,24}

Regarding the position of current and destination nodes, all the nodes in 3D mesh network are divided into 26 groups as E, W, N, S, U, D, EN, ES, EU, ED, WN, WS, WU, WD, NU, ND, SU, SD, ENU, END, ESU, ESD, WNU, WND, WSU, and WSD; where E, W, N, S, U, and D stands for East, West, North, South, Up, and Down directions. For ease of understanding we start introducing the HamFA routing algorithm in a 2D layer. Let us consider the examples of Fig. 2 where the bold and shaded nodes are the current and destination nodes, respectively. In Fig. 2(a), when the current node is located in an odd row (e.g. node 5) and the destination is to the NE, N, or NW position of the current node, there are two choices for delivering packets: north and west output ports. As can be seen in this figure, the labels of the nodes connected to these output ports (i.e. neighboring nodes 6 and 10) are greater than the label of the current node and smaller than the labels of the destination nodes (i.e. nodes 12, 13, 14, and 15). In this condition by sending packets through west or north direction, the routing takes place in ascending order and thus the network remains deadlock free. Since there are two choices of output ports, whenever one of the links becomes faulty, the other link can be used to deliver the packet. However, paths are not necessarily minimal while non-minimal paths may be taken when a link is faulty. This is the case when the packet is currently at node 5 and it is going to be forwarded to node 12 while the north link of node 5 is faulty. The packet has to take the west output port, which imposes a non-minimal path. When the destination is in the northwest direction, the path is minimal even in the presence of fault.

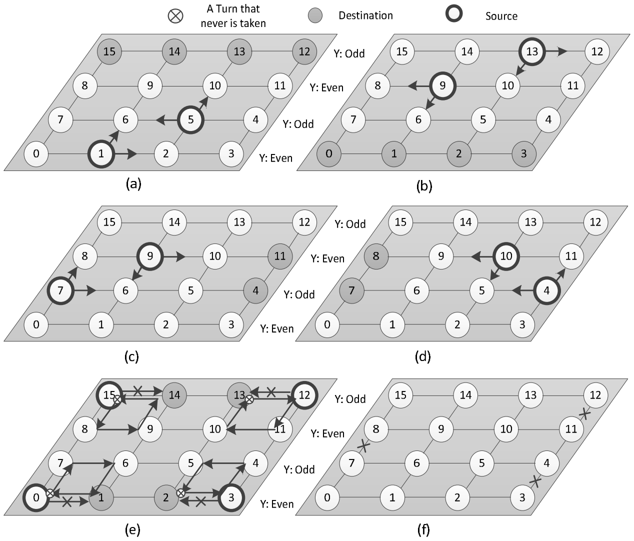

Fig. 2. A 2D layer (e.g. the first layer of a 4×4×4 mesh network)

The choices of output ports are different when the current node is in the even row such as the node 1 (i.e. north and east output ports can be selected). The network remains deadlock free by choosing a neighboring node with a greater label than the current node and smaller label than the destination. There is an exception when the current node is one hop away from the destination node along the Y direction (e.g. current node: 10 and destination node: 12). The neighboring node in the north direction (i.e. node 13) has a higher label than the destination node 12. However, the network remains deadlock free as a cycle cannot be formed by a single transaction from the high channel to the low channel sub-network (i.e. a cycle might be completed if a packet switches to the high channel subnetwork again). In sum, all packets have two output choices when delivering to the destination nodes in N, NE, or NW position. Similarly, in Fig. 2(b), packets are allowed to take the east or south output port in odd rows and the west or south output port in even rows when travelling to a destination node in S, SE or SW position. Fig. 2(c) and Fig. 2(d) show two possible choices of output ports when the destination is to the east and west of the current node, respectively. Faults in borderline links are investigated in Fig. 2(e) and Fig. 2(f). When an eastward link of a router located in the first row is faulty (e.g. the unidirectional link from the node 0 to node 1), the packet has to be routed around it (through nodes 7, 6, 1, and 0). Since the turn connecting the node 1 to the node 7 through the node 0 never takes place, no cycle can be formed and thus the network remains deadlock free. The forbidden turns in other situations are shown with a cycle-cross sign. Fig. 2(f) illustrates the only cases in which HamFA is unable to remove the cycles and thus by occurring faults on these links, the network would not be guaranteed to be deadlock free.

B. Hamiltonian-based Fault-Tolerant Method for 3D Mesh

As been already mentioned, the complexity of fault-tolerant algorithms in 3D network increases due to facing a larger number of turns in the network. However, this complexity can be easily addressed by using a Hamiltonian path. On the other hand, by moving to the 3D mesh network, the ability of tolerating faults is improved because of providing more output port choices per node compared with 2D mesh network. For example, middle layers are completely resilient against any one-faulty cases (within layers) by taking advantages of upper and lower layers. The reason is that every node in middle layers has at least two neighboring nodes with greater labels than the current node and two neighbors with smaller labels. Consider an example in Fig. 3 when the packet is routing in the high channel sub-network toward the destination 44 and it is already at node 43 while the link (43, 44) is faulty. The packet cannot be routed to the west direction due to the probability of creating cycle as illustrated in Fig. 2(f). However, the packet can continue routing in the high channel sub-network by selecting the upward channel (i.e. node 52). This packet is able to reach the destination by switching to the low channel sub-network and travelling though this sub-network until it reaches the destination. The horizontal links that cannot be supported by the HamFA method are indicated in Fig. 3. Among vertical links, only the faults on three links are not supported by the HamFA method, which are the links between nodes (31 and 32), (47 and 48), and (15 and 16). All the other nodes in the network have an alternative option to deliver the packet. In the HamFA method, the possibility of sending packets through the vertical link is checked before those of horizontal links. Therefore, if the label of the neighboring node in the vertical direction is between the current and destination node, the packet is sent to the vertical link. If the condition is not met, the horizontal links are examined. In the case that horizontal links are faulty or do not exist (e.g. in borderline routers), the packet is forced to send through the vertical link. Table 1 shows the order of selecting among output ports regarding the destination position. As can be seen in this table, due to symmetry inherent in the HamFA method, the output port choices are the same for different destination positions. It makes the routing algorithm quite simple.

Table 1. The options of output ports provided by HamFA in 3D NoCs

Destination Position		Y=Even	Y=Odd
N, D, ND, SD, WN, EN, END, WND	Z=Odd	D,N,E,D	D,N,W,D
N, U, NU, SU, WN, EN, ENU, WNU	Z=Even	U,N,E,U	U,N,W,U
U, EU, WU, ENU, WNU	Z=Odd	U,W,S,U	U,E,S,U
D, ED, WD, END, WND	Z=Even	D,W,S,D	D,E,S,D
ED, WD, ESD, WSD	Z=Odd	D,E,N,D	D,W,N,D
EU, WU, ESU, WSU	Z=Even	U,E,N,U	U,W,N,U
S, NU, SU, ES, WS, ESU, WSU	Z=Odd	U,S,W,U	U,S,E,U
S, ND, SD, ES, WS, ESD, WSD	Z=Even	D,S,W,D	D,S,E,D
W	Z=Odd	W,S,U,N	W,N,D,S
W	Z=Even	W,S,D,N	W,N,U,S
E	Z=Even	E,N	E,S
E	Z=Odd	E,N	E,S

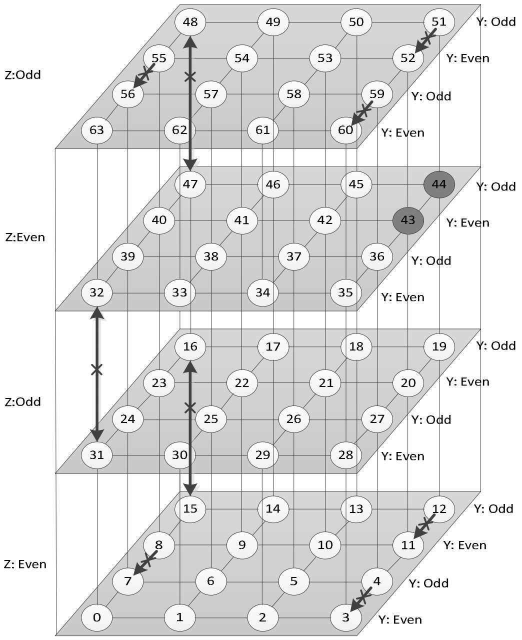

Fig. 3. 4×4×4 mesh network with the Hamiltonian path labeling

IV. EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed routing scheme, a NoC simulator is developed with VHDL to model all major components of the on-chip network. For all the routers, the data width is set to 32 bits. Each input buffer can accommodate 8 flits in each channel. Moreover, the packet length is uniformly distributed between 5 and 10 flits. As a performance metric, we use latency defined as the number of cycles between the initiation of a message issued by a Processing Element (PE) and the time when the message is completely delivered to the destination. The request rate is defined as the ratio of the successful message injections into the network over the total number of injection attempts. We perform the experiments on a 4×4×4 3D mesh network. The simulator is warmed up for 12,000 cycles and then the average performance is measured over another 200,000 cycles. AFRA [10] is considered as the baseline scheme. Before investigating the experimental results, we make a simple comparison between HamFA and AFRA methods. AFRA is a fault tolerant method supporting all one-faulty unidirectional links only in vertical channels. So, it is reliable by 33,3% over all links while HamFA is reliable by more than 95% in a 4×4×4 mesh network. AFRA requires a fault distribution mechanism to know about the fault information on all vertical links along each row while HamFA does not require any extra information as it normally knows about the fault on its unidirectional links towards the adjacent neighbors. AFRA claims that is able to support many one-faulty links if the vertical links are faulty in the same direction. HamFA, however, is able to support many one-faulty links either on vertical or horizontal links if the faulty links are belonging to the same sub-network.

A. Performance Analysis

In the uniform traffic profile, each processing element (PE) generates data packets and sends them to another PE using a uniform distribution [15]. In Fig. 6(a), the average communication latencies of the HamFA and AFRA routing algorithms are measured for fault-free and one-faulty link cases. As observed from the results, under the hotspot traffic profile, HamFA performs better than the AFRA routing algorithm in both fault-free and one-faulty link cases.

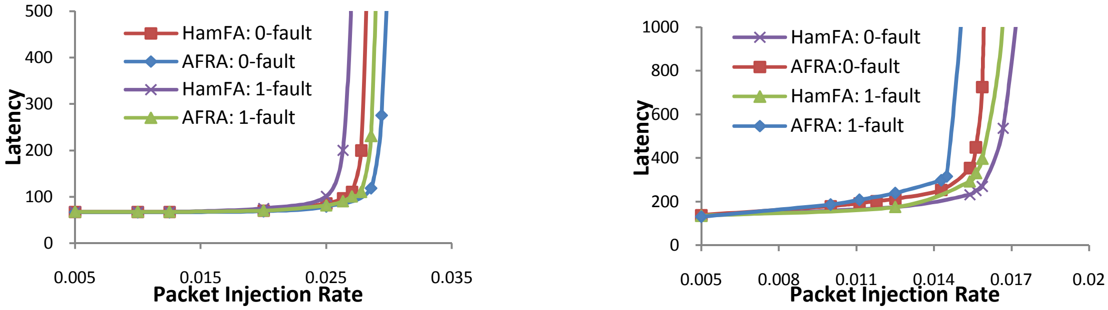

Fig. 6. Performance analysis of HamFA and baseline methods in 4×4×4 3D-mesh network (a) under uniform traffic profile (b) hotspot traffic profile (h=10%) in fault-free, and 1-faulty cases.

The reason is that, HamFA is an adaptive method (similar to MAR routing algorithm [16]) and it has alternative choices to route packets while AFRA is a deterministic routing algorithm. However, under the uniform traffic profile in fault-free cases, AFRA leads to a better performance due to the compatibility between dimension-order routing and uniform traffic.

B. Reliability Evaluation under Uniform Traffic Profile

To evaluate and compare the reliability of HamFA and AFRA, the number of faulty links increases from 1 to 3. All faulty links are selected using a random function. The results are obtained using 10,000 iterations in a 4×4×4 3D mesh network when the traffic is uniform. A network is reliable if all the injected packets reach their destinations. In other words, the network is counted as unreliable even if all packets reach the destinations except one packet. As shown in Fig. 4, HamFA can tolerate one, two, and three faulty links by 95%, 44%, 20% reliability, respectively.

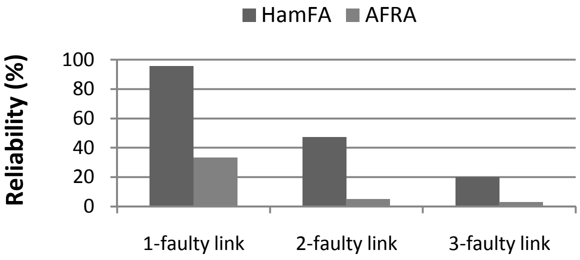

Fig. 4. Reliability analysis

C. Hardware Analysis

To assess the area overhead of the proposed method, the whole platforms of the HamFA and AFRA routing algorithms are synthesized by Synopsys Design Compiler. Each scheme includes network interfaces, routers, and communication channels. For synthesis, we use the UMC 90nm technology at the operating frequency of 1GHz and supply voltage of 1V. Depending on the technology and manufacturing process, the pitch of TSVs can range from 1μm to 10μm. In this work, the pad size for TSVs is assumed to be 5μm square with pitch of around 8μ. We perform place-and-route, using Cadence Encounter, to have precise power and area estimations. The layout areas of the HamFA and AFRA schemes are 25.3 mm^2 and 27.2 mm^2, respectively. This shows that AFRA has a larger area overhead (%6) because of using longer scan chains for each row in each layer.

V. CONCLUSION

In this paper, we proposed a general fault-tolerant mechanism based on the Hamiltonian path. In the basic form of the Hamiltonian path, packets are not able to switch between high channel and low channel sub-networks. To address this problem, we modify the usage of the Hamiltonian path in a network in order to be more efficient in tolerating faults. Our proposed algorithm based on this modification is able to tolerate almost all unidirectional one-faulty links by more than 95% reliability in 3D NoCs without using any virtual channel. This algorithm does not require any fault distribution mechanism or additional information in the packet header. Finally, the algorithm is simple that can be implemented with negligible hardware overhead. In addition to one-faulty links, this algorithm is able to support a wide range of multiple unidirectional faulty links as long as the faults occur either in the high channel or low channel sub-networks.

REFERENCES

[1] W. Tsai et al., “A fault-tolerant NoC scheme using bidirectional channel,” in Proc. of DAC, pp.918-923, 2011.

[2] E. Rijpkema et al., “Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip,” in Proc. of DATE, pp. 350-355, 2003.

[3] M. Cuviello et al., “Fault modeling and simulation for crosstalk in system-on-chip interconnects,” in Proc. of ICCAD, pp. 297-303, 1999.

[4] M. Ebrahimi et al., “MD: minimal path-based fault-tolerant routing in on-chip networks,” in Proc. of ASP-DAC, 2013.

[5] M. Koibuchi et al., “A lightweight fault-tolerant mechanism for network-on-chip”, in Proc. of NOCS, pp.13-22, 2008.

[6] M.H Neishaburi et al., “HW/SW architecture for soft-error cancellation in real-time operating system,” Journal of the Institute of Electronics, Information and Communication Engineers, v. 4, no. 23, pp. 755-761, 2007.

[7] M. Ebrahimi et al., “High performance fault-tolerant routing algorithm for NoC-based many-core systems,” in Proc. of PDP, 2013.

[8] M.H. Neishaburi, Z. Zilic, “ERAVC: Enhanced reliability aware NoC router,” in Proc. of ISQED, pp. 1-6, 2011.

[9] S. Pasricha, Y.Zou, “A low overhead fault tolerant routing scheme for 3D Networks-on-Chip,” In Proc. of ISQED, pp. 204-211, 2011.

[10] S. Akbari et al., “AFRA: A low cost high performance reliable routing for 3D mesh NoCs,” In Proc. of DATE, pp.332-337, 2012.

[11] E. O. Amnah, W.L. Zuo, “Hamiltonian paths for designing deadlock-free multicasting wormhole-routing algorithms in 3-D meshes,” Journal of Applied Sciences, pp. 3410-3419, 2007.

[12] M. Ebrahimi et.al., “HAMUM – A Novel Routing Protocol for Unicast and Multicast Traffic in MPSoCs,” in Proc. of PDP, pp. 525-532, 2010.

[13] X. Lin, L.M. Ni, “Multicast communication in multicomputer networks,” IEEE Trans. Parallel Distrib. Syst., v.4, pp. 1105-1117, 1993.

[14] M. Ebrahimi et.al.,“Path-based partitioning methods for 3D networks-on-chip with minimal adaptive routing,” IEEE Transaction on Computers (IEEE TC), 2012.

[15] M. Daneshtalab et al., “Adaptive input-output selection based on-chip router architecture,” Journal of Low Power Electronics, v. 8, no. 1, pp. 11-29, 2012.

[16] M. Ebrahimi et al., “Exploring Partitioning Methods for 3D Networks-on-Chip Utilizing Adaptive Routing Model,” in Proc. of NOCS, pp. 73-80, 2011.