Fault-tolerant Method with Distributed Monitoring and Management Technique for 3D Stacked Meshes

Masoumeh Ebrahimi, Masoud Daneshtalab, Pasi Liljeberg, Hannu Tenhunen
Department of Information Technology, University of Turku, Finland
{masebr, masdan, pakrli, hanten}@utu.fi

Abstract— In this paper, we present a fully adaptive routing algorithm for 3D stacked mesh, called 3D-FAR. This algorithm utilizes two, two, and four virtual channels along the X, Y, and Z dimensions, respectively. It allows packets to take any shortest paths between the source and destination routers. 3D-FAR divides the network into four disjoint subnetworks. To improve the fault-tolerant capability of the network, packets are able to switch between subnetworks in an ascending order. In this paper, we also propose a fault-tolerant algorithm for 3D mesh network, called 3D-FT. This method is discussed both for tolerating faulty routers and links in the network. For tolerating faulty routers, only the shortest paths are taken while for tolerating faulty links, the non-minimal paths are used only when the source and destination routers are located in the same dimension with a faulty link between them. 3D-FT utilizes a distributed monitoring and management technique to distribute the fault statuses among the surrounding routers.

Keywords—3D stacked mesh; Fault-tolerant method; Distributed monitoring and management technique; Fully adaptive routing algorithm; Inter-layer and intra-layer faults;

I. INTRODUCTION

Traditionally, System-on-Chip (SoC) designers employ buses or hierarchical bus structures to interconnect Intellectual Property (IP) blocks. The advances in semiconductor technologies make it possible to integrate billions of gates and hundreds of processing units into a single chip [1]. This technology trend implies the need for a structured, scalable, reusable, and high performance communication platform which cannot be offered by the bus infrastructures. Therefore, Networks-on-Chip (NoCs) have emerged as a solution to address the communication demands of future SoC designs. Scalability and reusability characteristics of NoC result in reducing design time and shortening time to market for new products. NoC consists of an interconnection of routers to enable a large number of cores to communicate with each other. In a mesh-based NoC, each core is connected to a router by a local network interface. Cores communicate with each other by propagating packets through routers in the network. Routers are connected to their neighbors through bidirectional links. Typically in NoCs, the resources are scarce regarding the ever increasing demand for a high performance communication among cores. Therefore, a major challenge in this domain is to achieve a high performance system using the limited available resources [1].

As the size of the network is scaled up, the transmission delay between distant routers is significantly increased in 2D networks which results in lower performance and higher power consumption. In addition, a 2D Integrated Circuit (IC) design imposes a very large chip area as the number of cores increases. Considering 2D design challenges, the technology is moving towards the concept of 3D integrated circuits where multiple active silicon layers are vertically stacked. Combining the benefits of 3D ICs and NoCs schemes provides a significant performance gain for SoCs.

Routing algorithms can be classified as deterministic and adaptive approaches. The simplest deterministic routing approach is dimension-order routing which is known as the XY and XYZ algorithms in 2D and 3D networks, respectively. Implementations of deterministic routing algorithms are simple but they are not able to balance the load across the links in a non-uniform or bursty traffic [3].

Adaptive routing has been used in interconnection networks to improve network performance and to tolerate link or router failures. In adaptive routing algorithms, a packet can traverse from a source to a destination through multiple paths. Specifically, adaptive routing algorithms can be used to avoid congestion by adapting the routing decision to the network status. Adaptive routing algorithms can be either partially adaptive or fully adaptive. In partially adaptive routing algorithms, packets are limited to choose among some shortest paths, while in fully adaptive methods, packets are allowed to take any shortest paths available between the source and destination pair [4,11]. Obviously, fully adaptive routing algorithms can distribute packets more efficiently over the network, and thus alleviating congestion. One of the contributions of this paper is to propose a fully adaptive routing algorithm in 3D NoCs. This method requires two, two, and four virtual channels along X, Y, and Z dimensions, respectively.

The reminder of this paper is organized as follows. In Section II, the related work is given. In Section III, the fully adaptive routing algorithm in 3D NoCs is introduced. Section IV discusses the fault-tolerant method along with the monitoring and management technique. The results are reported in Section V while the summary and conclusion are given in the last section.

II. RELATED WORK

Several methods are presented in the realm of 2D NoCs in order to balance the traffic load over the network. DyXY [15] is a fully adaptive routing algorithm using one and two virtual channels along the X and Y dimensions. There are few partially and fully adaptive algorithms in a 3D mesh network. MAR [16] is a partially adaptive routing algorithm in 3D NoCs which is based on the Hamiltonian path. It is a simple approach providing the adaptivity without using virtual channels. The first fully adaptive routing algorithm in a 3D mesh network is presented in [17], called DyXYZ. Using this algorithm, packets are able to take any shortest paths between the source and destination routers. DyXYZ requires four, four, and two virtual channels along the X, Y, and, Z dimensions to provide fully adaptiveness.

A number of studies presented solutions to tolerate faulty links or routers in a 2D mesh network. All traditional methods are based on taking unnecessary non-minimal routes to tolerate faults, which increases the latency of packets significantly. Recently, four high performance fault-tolerant methods have been presented in a 2D mesh network, which are based on using the shortest path. These algorithms maintain the performance of the network while tolerating faults. Two of these algorithms tolerate faulty links (MD [23] and MAFA [24]) and two others tolerate faulty routers (HiPFaR [25] and MiCoF [26]). Although MAFA attempts to choose the shortest paths, in some cases non-minimal paths might be taken even if a non-faulty minimal path exists between the source and destination pair. MD and HiPFAR use the non-minimal paths only when the source and destination are located in the same dimension and there is a fault between them. MiCoF, however, does not use any non-minimal path by a simple modification in the router architecture.

978-1-4799-0565-2/13/$31.00 ©2013 IEEE

Table 1. Completed pairs within each subnetwork

Subnetworks	Pair (X^+, X^-)	Pair (Y^+, Y^-)	Pair (Z^+, Z^-)	Complete pair
$(X1^+)(Y1^+)(Z1^*)$	X^- is missing	Y^- is missing	Pair exists	Z
$(X2^+)(Y1^-)(Z2^*)$	X^- is missing	Y^+ is missing	Pair exists	Z
$(X2^-)(Y2^-)(Z3^*)$	X^+ is missing	Y^+ is missing	Pair exists	Z
$(X1^-)(Y2^+)(Z4^*)$	X^+ is missing	Y^- is missing	Pair exists	Z

Table 2. Different subnetworks are disjoint from each other

Subnetwork A	Subnetwork B	X dimension	Y dimension	Z dimension
$(X1^+)(Y1^+)(Z1^*)$	$(X1^-)(Y2^+)(Z4^*)$	Differs in vc direction	Differs in vc number	Differs in vc number
$(X1^+)(Y1^+)(Z1^*)$	$(X2^+)(Y1^-)(Z2^*)$	Differs in vc number	Differs in vc direction	Differs in vc number
$(X1^+)(Y1^+)(Z1^*)$	$(X2^-)(Y2^-)(Z3^*)$	Differs in vc number	Differs in vc number	Differs in vc number
$(X1^-)(Y2^+)(Z4^*)$	$(X2^+)(Y1^-)(Z2^*)$	Differs in vc number	Differs in vc number	Differs in vc number
$(X1^-)(Y2^+)(Z4^*)$	$(X2^-)(Y2^-)(Z3^*)$	Differs in vc number	Differs in vc direction	Differs in vc number
$(X2^+)(Y1^-)(Z2^*)$	$(X2^-)(Y2^-)(Z3^*)$	Differs in vc direction	Differs in vc number	Differs in vc number

There are a number of fault-tolerant algorithms investigated in a 3D mesh network. The 4NP-First routing algorithm [27] utilizes two separate virtual channels, one dedicated to 4N-First routing algorithm and another one 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 mesh networks. The planar-adaptive routing algorithm [28] is a well-known method presented in the realm of interconnection networks. This algorithm requires one, three, and two virtual channels along the X, Y, and Z dimensions. The adaptivity of this method is limited to a fully adaptive routing algorithm inside a sequence of 2D planes. HamFA [29] is a Hamiltonian-based fault-tolerant algorithm presented for 3D mesh network. This algorithm tolerates almost all one-faulty links either in horizontal or vertical links without using any virtual channels.

In this paper, we present a method to tolerate faulty links and routers in 3D stacked mesh. At first, we introduce a fully adaptive routing algorithm for 3D network which requires two virtual channels less than the DyXYZ method [17]. To the best of our knowledge, this is the minimum number of virtual channels presented for a fully adaptive routing algorithm in a 3D mesh network. Then, we present a fault-tolerant routing algorithm for tolerating faults (either inter-layer or intra-later faults). To maintain the performance of the network, the proposed method uses only the shortest paths in the presence of faults as long as a path exists. If a minimal path does not exist between the source and destination routers, the use of non-minimal paths is necessitated. We present an innovative method to support these cases. The idea behind this method is to allow packets to switch between disjoint subnetworks in an ascending order to provide a high fault-tolerant capability. For distributing the fault information, the proposed fault-tolerant algorithm employs an efficient distributed monitoring and management approach, diminishing the negative impacts of centralized approaches.

III. FULLY ADAPTIVE ROUTING ALGORITHM (3D-FAR)

DyXYZ [17] is a fully adaptive routing algorithm, requiring four, four, and two virtual channels along the X, Y, and Z dimensions. Using 3D-FAR, the number of virtual channels reduces to two, two, and four virtual channels along the X, Y, and Z dimensions, respectively (i.e. two virtual channels less than DyXYZ). In the 3D-FAR algorithm, the network is partitioned into four subnetworks (Fig. 1) as: $((X^+)(Y^+)(Z^*)$, $(X^-)(Y^+)(Z^*)$, $(X^+)(Y^-)(Z^*)$, and $(X^-)(Y^-)(Z^*))$ where "+", "-" represent the channels along the positive and negative directions, respectively, and "*" stands for both positive and negative directions (i.e. a bidirectional channel). The deadlock freeness can be proved by guaranteeing that each subnetwork uses a separate set of virtual channels. The virtual channel assignment of 3D-FAR is as follow: $((X1^+)(Y1^+)(Z1^*)$, $(X2^+)(Y1^-)(Z2^*)$, $(X2^-)(Y2^-)(Z3^*)$, and $(X1^-)(Y2^+)(Z4^*))$ where the numbers indicate the virtual channel numbers assigned to a dimension.

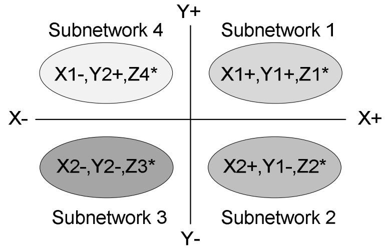


Fig. 1. Four different regions in 3D-FAR

Within each subnetwork, a cycle can be formed if packets are able to take both positive and negative directions along at least two dimensions. For example, to form a cycle in a XY plane, it is necessary to take X^+, X^-, Y^+, and Y^- directions. Similarly, to form a cycle in a XZ plane, there should be a possibility of taking X^+, X^-, Z^+, and Z^- directions. Finally, to form a cycle in an YZ plane, Y^+, Y^-, Z^+, and Z^- directions should be taken by packets. As can be obtained from four different subnetworks in 3D-FAR, only one pair along the Z dimension (Z^*: Z^-,Z^+) is completed in each of four subnetworks, and thus there is no possibility of forming a cycle in each subnetwork (Table 2). To prove that the network is deadlock-free between subnetworks, it is enough to show that different subnetworks are disjoint from each other. By a pairwise comparison among each two subnetworks, it can be easily obtained that either the direction or the virtual channel number differs along each dimension. For example, the subnetwork 1 $(X1^+)(Y1^+)(Z1^*)$ and the subnetwork 4 $(X1^-)(Y2^+)(Z4^*)$ are disjoint along the X dimension since the subnetwork 1 covers the positive direction of the virtual channel 1 ($X1^+$) and the subnetwork 4 covers the negative direction of virtual channel 1 ($X1^-$). The two subnetworks are also disjoint along the Y and Z dimensions since the virtual channel 1 is employed in the subnetwork 1 $(Y1^+)(Z1^*)$ and the virtual channel 2 and 4 is used in the subnetwork 4 $(Y2^+)(Z4^*)$. This comparison can be done for any other pair of subnetworks (Table 2). As a cycle cannot be formed within each subnetwork and on the other hand different subnetworks are disjoint from each other, it is proved that the whole network is deadlock-free.

According to this proof, subnetworks are disjoint from each other. However, to improve the fault-tolerant capability, we enable the switching between subnetworks without creating cycles. As shown in Fig. 2, packets can traverse between subnetworks in a strictly ascending order such that the cycle cannot be formed. It means that packets in the subnetwork 1 can switch to the subnetwork 2, but after switching, they can no more use any channels from the subnetwork 1. Similarly, packets in the subnetwork 1 or the subnetwork 2 can switch to the subnetwork 3, losing the possibility of using the channels of the subnetwork 1 or the subnetwork 2. Finally, packets in the subnetwork 1, subnetwork 2, or subnetwork 3 can switch to the subnetworks 4. In general, after using a channel in a higher subnetwork, switching to the lower subnetwork is not allowed.

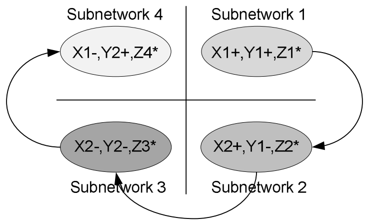


Fig. 2: Switching between subnetworks is possible only in an ascending order

IV. FAULT-TOLERANT METHOD FOR 3D MESH NETWORKS

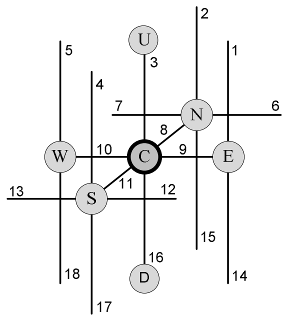
In this section, we present a fault-tolerant routing algorithm for 3D stacked mesh, called 3D-FT. We investigate this method both for tolerating faulty routers and faulty links. At first, we explain the fault monitoring and management technique and then we discuss the approach of tolerating faults.

A. *Fault Monitoring and Management Technique*

Fault-tolerant routing algorithms require to know about the location of faults in, at least, their surrounding routers. To be able to tolerate faults, some fault information should be provided at each routing unit. We assume that faults are detected by fault-detection techniques and they are monitored at each router. 3D-FT needs to monitor the faulty/healthy statuses of its 6 neighboring routers and 18 surrounding links as shown in Fig. 3 at a given router (i.e. router C). The links include the instant links connected to the neighboring routers (i.e. 3, 8, 9, 10, 11, and 16). The information of these instant links is already provided for the router C using a fault-detection technique. However, the fault information of the other links should be sent to the router C. These links are as: four links connected to the north neighboring router (i.e. 2, 6, 7, and 15), four links connected to the south neighboring router (i.e. 4, 12, 13, and 17), two links connected to the east neighboring router (i.e. 1 and 14), and two links connected to the west neighboring router (i.e. 5 and 18). Moreover, using a fault-detection technique, a router is informed about the fault status of itself while the information about the neighboring routers (i.e. north, south, east, west, up, and down) should be transferred to the router C. The fault management unit is responsible to combine the fault information at each router and transfers it to the neighboring routers. In this example, the north and south neighboring routers combine the fault information of the router itself and four connected links and transfer a 5-bit information to the router C. East and west neighboring routers transfer 3 bits to the router C; 1 bit for the fault status of the router and 2 bits for the fault statuses of the links. From the up and down directions, 1-bit information is transferred to the router C which indicates the fault status of the connected router. In reverse, the fault status of the router C and its four links 3, 9, 10, and 16 are transferred to the north and south neighboring routers. The fault status of the router C and its two links 3 and 16 are delivered to the east and west neighboring routers. Finally, only the fault status of the router C is sent to the up and down neighboring routers. 3D-FT uses a distributed management technique and it does not utilize any centralized approach for this purpose.

B. *Tolerating Faulty Routers*

Commonly, when a router is faulty, the fault-tolerant algorithms are used to reroute packets around the faulty region. It introduces the non-minimal paths, increasing the latency of packets considerably. The MICOF [26] approach addresses this inefficiency in a 2D mesh network by a simple modification in the router architecture. Using the MiCoF approach, all packets use only the shortest paths in the presence of faults. In the router architecture of MiCoF and in the case of faults, the east input channel is directly connected to the west output channel while the west input channel is connected to the east output channel. Similarly, the packets coming from the north or south input channels are directly connected to the south or north output channels, respectively. In other words, the whole router acts as a wire, connecting the input channels to output channels in specific directions. Compared with normal router architecture, this architecture needs a few amounts of multiplexers and de-multiplexers at input and output ports plus a small wiring overhead. 3D-FT uses the same idea and provides the ability to connect the input and output links of the faulty routers in appropriate directions. The router architecture of 3D-FT is shown in Fig. 4 where in the case of fault the following connections are made: the east input channel is connected to the west output channel; the west input channel is connected to the east output channel; the north input channel is connected to the south output channel; the south input channel is connected to the north output channel; the up input channel is connected to the down output channel; and the down input channel is connected to the up output channel.



Neighbor	Bits
North	5
South	5
East	3
West	3
Up	1
Down	1

Fig. 3: Fault monitoring and management of 3D-FT

Fig. 5(a) shows a 3×3×3 mesh topology with four faulty routers. A faulty router itself and the core connected to it are disconnected from the network while the links are used to connect the neighboring routers in appropriate directions. Using the 3D-FT architecture, the resulted network is illustrated in Fig. 5(b).

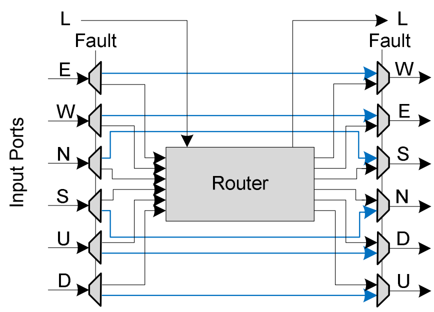


Fig. 4: Router architecture of 3D-FT

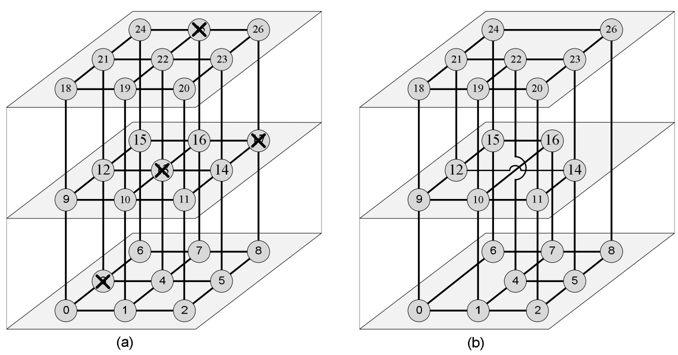
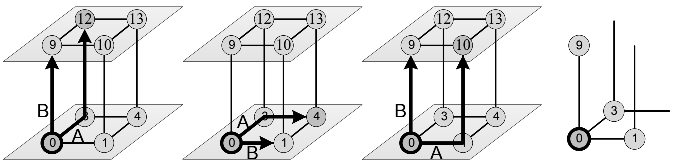


Fig. 5: (a) Four faulty routers in a 3×3×3 mesh network (b) The resulted network using the 3D-FT architecture

By using the 3D-FT, the network is fault-tolerant against any single faulty routers. When the source and destination are located in the same row or column, the faulty router can be seen as a wire, maintaining the connection between the source and destination routers. In other cases, the packet has at least two neighboring routers, so that if one of them is faulty, the packet is sent through the other one. In sum, for tolerating faulty routers, only the shortest paths are used.

C. *Tolerating Faulty Links*

Now, we show how faulty links can be tolerated using 3D-FT. When packets have three minimal choices, they are sent through a non-faulty link. When the distance reaches one along two dimensions while it is zero along the third dimension, it is important to send a packet through a non-faulty path as the wrong decision results in dropping the packet or taking non-minimal routes. In Fig. 6(a), the packet can be sent to the north or up direction. If only the neighboring links are checked, the packet might be sent to the up direction. However, if the link between the router 9 and the destination router 12 is faulty, a non-minimal path should be taken. In order to avoid this situation, at first, the fault status of the north-up route is checked. If the path is non-faulty, the packet is sent to the north direction; otherwise the up direction is selected. Similarly, in Fig. 6(b), the status of the north-east path is checked and if the path is non-faulty, the packet is sent to the north direction; otherwise the east direction is selected. Finally, in Fig. 6(c), the packet is delivered to the east direction if the east-up path is non-faulty; otherwise the up direction is selected. As was already described in the fault monitoring and management section, the current router knows about the fault statuses of some surrounding links shown in Fig. 6(d). When the current and destination routers are located in the same dimension and the link between them is faulty, the non-minimal route is necessitated. In order to avoid taking unnecessary longer paths, 3D-FT always avoids reducing the distance to zero along two dimensions when the distance along the third dimension is greater than one.



(a) (b) (c) (d)

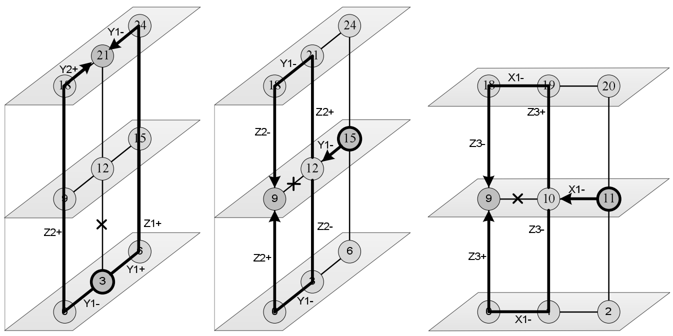
Fig. 6: 3D-FT when the distance reaches one along two dimensions

When the source and destination are located along a same dimension while there is a faulty link between them, packets should take non-minimal paths. To tolerate these faults, 3D-FT takes advantage of switching between subnetworks. For this purpose, packets are started routing in the lowest possible subnetwork and then they can switch to a higher subnetwork in an ascending order. The rules of the 3D-FT algorithm are as follows: 1- If a link in the Z dimension is faulty, packets are rerouted through the Y dimension. 2- If a link in the X or Y dimension is faulty, packets are rerouted through the Z dimension.

For performing these rules, packets may require to change their subnetworks. Let us consider the example of Fig. 7(a) where the source and destination are located at router 4 and 22 and the link 4-13 (i.e. connecting the router 4 and router 13 together) is faulty. Since the link along the Z dimension is faulty, according to Rule1, packets are rerouted through the Y dimension which can be in its positive or negative direction. Let us assume that rerouting takes place in the positive direction of the Y dimension. Since the subnetwork 1 covers $Y1^+$, the packet uses this channel. Then the packet should be routed along the positive direction of the Z dimension. The subnetwork 1 also covers $Z1^+$, so the packet is still routed using the channels of the subnetwork 1. The packet continues along this direction until it reaches the same layer as the destination router where it should be sent through the negative direction of the Y dimension. However, this channel is not included in the subnetwork 1 while the subnetwork 2 covers $Y1^-$ so that the packet uses this channel to reach the destination router. Now, we explain the situation when the packet sends along the negative direction of the Y dimension at the router 4. The subnetwork 1 does not cover the negative direction of the Y dimension while the subnetwork 2 covers it, so the packet uses $Y1^-$ from this subnetwork. The packet can be routed along the Z dimension using $Z2^+$ from the same subnetwork. Finally, the positive direction of the Y dimension is not covered by the subnetwork 2 and the $Y2^+$ is taken from the subnetwork 4 and the packet reaches the destination. As every router has at least one neighbor along the Y dimension, faults on vertical connections can be tolerated by rerouting packets through the Y dimension.

Fig. 7(b) shows the case where the link in the Y dimension is faulty and it is tolerated by rerouting packets through the Z dimension. We investigate the example where the source and destination are located at routers 15 and 9, respectively, and the link 12-9 is faulty. At first the negative direction of the Y dimension should be taken. Since the subnetwork 1 does not cover the negative direction of the Y dimension the next subnetwork is checked. The subnetwork 2 covers $Y1^-$ and thus the packet uses this channel. For routing along the Z dimension either in the positive or negative direction, the channels of the subnetwork 2 are used ($Z2^*$). Then the packet should be routed along the negative direction of the Y dimension which is covered by the subnetwork 2 ($Y1^-$) and the positive directions of the Z from the same subnetwork ($Z2^*$).

In Fig. 7(c), a faulty link in the X dimension is tolerated by rerouting packets along the Z dimension. Let us assume the case when the source and destination are located at the router 11 and 9, respectively and the link 10-9 is faulty. The negative direction of the X dimension is not covered by the subnetworks 1 and 2, and thus the $X2^-$ from the subnetwork 3 is used. The packet is rerouted along the Z dimension using $Z3^+$ or $Z3^-$ from the same subnetwork. The packet needs to take the $X2^-$ and then $Z3^+$ or $Z3^-$ to reach the destination router. All of these channels are covered by the subnetwork 3. As every router has at least one neighbor in the Z dimension, faults on the X or Y dimension can be tolerated by rerouting packets along the Z dimension.



(a) (b) (c)

Fig. 7: Source and destination routers are located in the same row or column and there is a faulty link between them

3D-FT is able to tolerate all single faulty routers and links in the network using the shortest paths as possible. However, in many cases, 3D-FT tolerates multiple faults as well. To achieve the maximum adaptiveness, 3D-FT can be designed in a way that all packets use the channels of the subnetwork 1 if there is no possibility of routing packets through this subnetwork and they switch to the subnetwork 2 and so on.

V. RESULTS AND DISCUSSION

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. An input buffer can accommodate 8 flits in

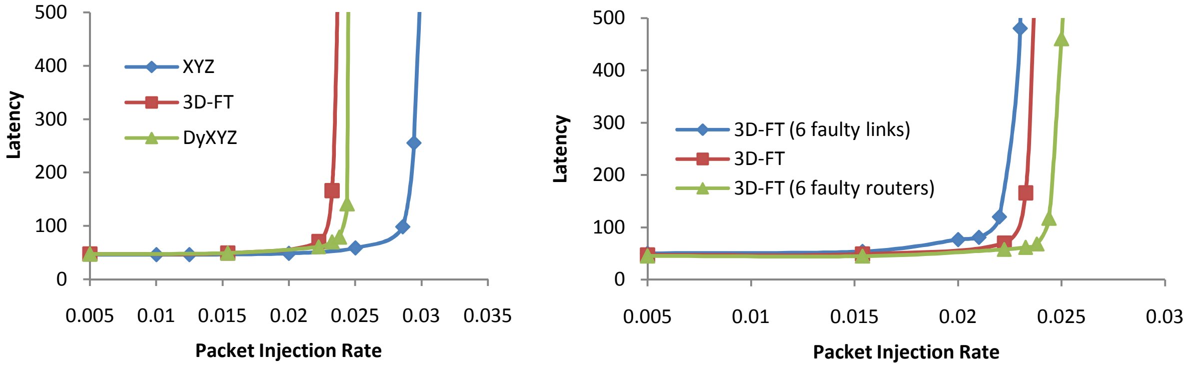


Fig. 8: Performance under uniform traffic profile in (a) fault-free network (b) under the presence of faults.

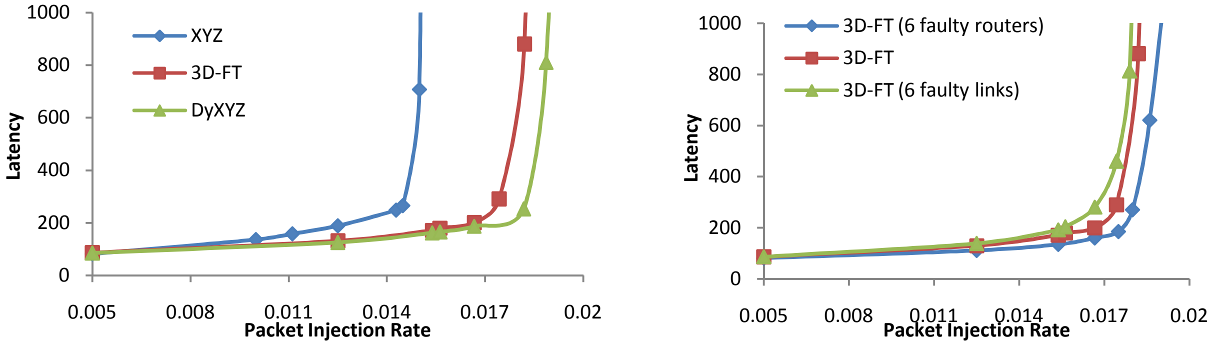


Fig. 9: Performance under hotspot traffic profile in (a) fault-free network (b) under the presence of faults

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 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 two sets of simulations; 1- to measure the efficiency of the 3D-FT method over the deterministic routing algorithm XYZ and adaptive routing algorithm DyXYZ. 2- to measure the reliability of 3D-FT over HamFA [29]. In both sets of simulation, we perform the experiments on a 4×4×4 3D mesh network. For performance analysis, the simulator is warmed up for 12,000 cycles and then the average performance is measured over another 200,000 cycles. Before investigating the experimental results, we make a simple comparison between HamFA and 3D-FT methods. HamFA is a fault-tolerant method supporting almost all one-faulty unidirectional links. It is able to tolerate faults either on vertical or horizontal links without using virtual channels. On the other hand, 3D-FT can tolerate both faulty links and routers while guaranteeing to tolerate all single faults wherever in the network. 3D-FT is built upon a fully adaptive method requiring two, two, and four virtual channels.

Uniform Traffic Profile: In the uniform traffic profile, each processing element (PE) generates data packets and sends them to another PE using a uniform distribution. In Fig. 9(a), the average communication delay as a function of the average packet injection rate is plotted for the XYZ, DyXYZ, and 3D-FT schemes. As observed from the results, XYZ leads to a lower latency than the 3D-FAR method. The reason is that, under uniform traffic, dimension-order routing is best for evenly distributing traffic over the network. DyXYZ performs better than 3D-FAR which is because of using two more virtual channels than 3D-FT. Note that XYZ uses the same number of virtual channels as 3D-FT. In Fig. 9(a), we injected six random faulty links and routers into the network and measured the latency of the arrived packets. As can be seen in this figure, the latency is slightly increased since 3D-FT avoids using non-minimal routes as possible. On the other hand, under six faulty routers, the performance improves. This is due to the fact that the faulty router performs as a wire, meaning that no routing takes place at these routers and thus the total number of hops decreases.

Hotspot Traffic Profile: Under the hotspot traffic pattern, some nodes are chosen as hotspots receiving an extra portion of the traffic in addition to the regular uniform traffic. In simulations, given a hotspot percentage of *H*, a newly generated message is directed to each hotspot node with an additional *H* percent probability. We simulate the hotspot traffic with four hotspot nodes at positions (2,1) and (3,1) in the layer 2 and the same positions in the layer 3. The performance of the XYZ, DyXYZ, and 3D-FT methods with *H=10%* is illustrated in Fig. 10(a). As observed from this figure, DyXYZ and XYZ leads to the best and worst performance while the performance of 3D-FT is slightly lower than DyXYZ which is mainly because of the lower number of virtual channels in 3D-FT than DyXYZ. Fig. 10(b) compares the performance of 3D-FT in a fault-free case with the situations when there are six faulty links and routers in the network. The obtained result is similar to the uniform traffic such that the faulty links decrease the performance of the network while the faulty routers result in a better network performance.

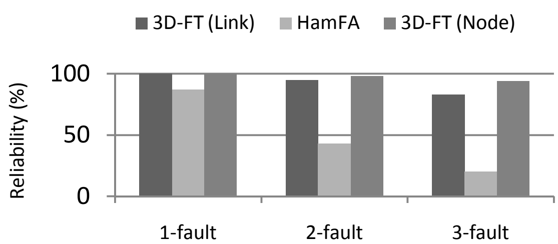


Fig. 10: Reliability measurment

Reliability Evaluation under Uniform Traffic Profile: We compared the reliability of 3D-FT with HamFA. For this purpose, we injected three faulty links and routers in the network when using 3D-FT and three unidirectional faulty links when using HamFA. All faults 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. 10, 3D-FT tolerating faulty routers obtains the best reliability because of its innovative router architecture. The reliability of HamFA decreases significantly in comparison with the 3D-FT when tolerating faulty links.

Hardware Analysis: To assess the area overhead of the proposed method, the whole platforms of the HamFA and 3D-FT methods 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 3D-FT schemes are 25.3 mm^2 and 26.2 mm^2, respectively. The area overhead of 3D-FT is slightly larger than HamFA because of employing the monitoring and management technique which does not exist in the HamFA method.

VI. Summary and Conclusion

In this paper, we described a faulty adaptive routing algorithm in details and proved its deadlock freeness. This algorithm requires only two, two, and four virtual channels along the X, Y, and Z dimensions, respectively. It is the minimum number of virtual channels among the traditional methods. Moreover, in this paper a fault-tolerant algorithm is proposed on top of this fully adaptive method to be able to tolerate both faulty routers and links. The proposed fault-tolerant algorithm takes only the shortest paths to tolerate faults as long as the shortest path exists. By a simple modification in the router architecture, no rerouting takes place to tolerate faulty routers. For tolerating faulty links, non-minimal paths are necessitated when the source and destination are located in the same dimension and there is a faulty link between them. These kinds of faults are tolerated by allowing packets to switch between subnetworks in strictly ascending order.

References

[1] M. Moadeli et al., "An Analytical Performance Model for the Spidergon NoC," in *Proceedings of 21st Annual Conference on Advanced Networking and Applications*, 2007, pp. 1014–1021.

[2] Y. Xie et al., "Design space exploration for 3D architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 2, no. 2, pp. 65–103, 2006.

[3] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, and H. Tenhunen, "High-Performance TSV Architecture for 3-D ICs," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2010, pp. 467–468.

[4] A. Y. Weldezion et al., "Scalability of network-on-chip communication architecture for 3-D meshes," in *Proceedings of 3rd ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, 2009, pp. 114 –123.

[5] J. Hu, L. Wang, L. Jin, and H. Z. JiangNan, "Electrical modeling and characterization of through silicon vias (TSV)," in *Proceedings of International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, 2012, vol. 2, pp. 1 –4.

[6] B. S. Feero and P. P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation," *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 32 –45, 2009.

[7] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "SunFloor 3D: A tool for Networks On Chip topology synthesis for 3D systems on chips," in *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, 2009, pp. 9 –14.

[8] H. Matsutani, M. Koibuchi, and H. Amano, "Tightly-Coupled Multi-Layer Topologies for 3-D NoCs," in *Proceedings of 41st International Conference on Parallel Processing*, 2007, p. 75.

[9] V. F. Pavlidis and E. G. Friedman, "3-D Topologies for Networks-on-Chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1081 –1090, 2007.

[10] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. K, "Design and Management of 3D Chip Multiprocessors Using Network-in-Memory," in *Proceedings of ISCA-33*, 2006, pp. 130–141.

[11] M. Daneshtalab et al., "High-performance on-chip network platform for memory-on-processor architectures," in *Proceedings of 6th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, 2011, pp. 1–6.

[12] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 138–149, 2007.

[13] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2003.

[14] L. M. Ni, P. K. McKinley, "A survey of wormhole routing techniques in direct networks," *Computer*, vol. 26, no. 2, pp. 62 –76, 1993.

[15] M. Li et al., "DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip," in *Proceedings of 43rd ACM/IEEE Design Automation Conference*, 2006, pp. 849 –852.

[16] M. Ebrahimi et al., "Path-based Partitioning Methods for 3D Networks-on-Chip with Minimal Adaptive Routing," *IEEE Transactions on Computers*, 2012.

[17] M. Ebrahimi et al., "DyXYZ: Fully Adaptive Routing Algorithm for 3D NoCs," in *Proceedings of 21th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Computing (PDP)*, 2013, pp. 499–503.

[18] D. Fick et al., "A highly resilient routing algorithm for fault-tolerant NoCs," in *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, 2009, pp. 21 –26.

[19] D. A. Fabien Chaix, "Fault-Tolerant Deadlock-Free Adaptive Routing for Any Set of Link and Node Failures in Multi-cores Systems.," pp. 52–59, 2010.

[20] W.-C. Tsai et al., "A Fault-Tolerant NoC Scheme using bidirectional channel," in *Proceedings of 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011, pp. 918 –923.

[21] Z. Zhang et al., "A reconfigurable routing algorithm for a fault-tolerant 2D-Mesh Network-on-Chip," in *Proceedings of 45th ACM/IEEE Design Automation Conference*, 2008, pp. 441 –446.

[22] M. Koibuchi, H. Matsutani, H. Amano, and T. Mark Pinkston, "A Lightweight Fault-Tolerant Mechanism for Network-on-Chip," in *Second ACM/IEEE International Symposium on Networks-on-Chip (NoCS)*, 2008, pp. 13 –22.

[23] M. Ebrahimi, M. Daneshtalab, J. Plosila, and Mehdipour, Farhad, "MD: Minimal path-based Fault-Tolerant Routing in On-Chip Networks," in *Proceedings of IEEE/ACM 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2013, pp. 35–40.

[24] M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen, "MAFA: Adaptive Fault-Tolerant Routing Algorithm for Networks-on-Chip," in *Proceedings of 15th Euromicro Conference on Digital System Design (DSD)*, 2012, pp. 201 –207.

[25] M. Ebrahimi, M. Daneshtalab, and J. Plosila, "High Performance Fault-Tolerant Routing Algorithm for NoC-based Many-Core Systems," in *Proceedings of 21th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Computing (PDP)*, 2013, pp. 463–469.

[26] M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen, "Minimal-Path Fault-Tolerant Approach Using Connection-Retaining Structure in Networks-on-Chip," in *Proceedings of 7th International Symposium on Networks-on-Chip (NOCS)*, 2013.

[27] S. Pasricha and Y. Zou, "A low overhead fault tolerant routing scheme for 3D Networks-on-Chip," in *2011 12th International Symposium on Quality Electronic Design (ISQED)*, 2011, pp. 1 –8.

[28] A. A. Chien and J. H. Kim, "Planar-Adaptive Routing: Low-cost Adaptive Networks for Multiprocessors," in *, The 19th Annual International Symposium on Computer Architecture, 1992. Proceedings*, 1992, pp. 268 –277.

[29] M. Ebrahimi, M. Daneshtalab, and J. Plosila, "Fault-Tolerant Routing Algorithm for 3D NoC Using Hamiltonian Path Strategy," in *Proceedings of 16th ACM/IEEE Design, Automation, and Test in Europe (DATE)*, 2013, pp. 1601–1605.