MAFA: Adaptive Fault-Tolerant Routing Algorithm for Networks-on-Chip

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

***Abstract*— while Networks-on-Chip have been increasing in popularity with industry and academia, it is threatened by the decreasing reliability of aggressively scaled transistors. This level of failure has architectural level ramifications, as it may cause an entire on-chip network to fail. Traditional fault-tolerant routing algorithms can overcome the faulty links or routers by rerouting packets around faulty regions. These approaches increase the packet latency and create congestion around the faulty region. In this paper, we present a novel fault-tolerant method that is able to route packets through shortest paths in the presence of faulty links, as long as a path exists. Although the same idea can be applied to a network with any number of virtual channels, we utilize two virtual channels to tolerate all one and two faulty links. Finally, the method is extended to support multiple faulty links by fully utilizing all allowable turns in the network.**

Keywords-component: fault-tolerant; minimal and adaptive routing algorithm.

I. Introduction

Networks-on-Chip (NoC) has become a promising solution for on-chip interconnection in many-core Systems-on-Chip (SoC) due to its reusability and scalability [1][2][3][4]. On-chip interconnects implemented with deep submicron semiconductor technology, running at GHz clock frequencies are prone to failures [1][5][6]. Routing techniques provide some degrees of fault tolerance in NoCs. Routing algorithms are mainly categorized into deterministic and adaptive [7][8][9][10]. A deterministic routing algorithm uses a fixed path for each pair of nodes resulting in increased packet latency especially in congested networks. In contrast, in adaptive routing algorithms, a packet is not restricted to a single path when traveling from a source node to its destination. Adaptive routing algorithms could achieve better fault-tolerant capabilities utilizing alternative routing paths. In wormhole routings, messages are divided into small flits traveling through the network in a pipelined fashion. This approach eliminates the need to allocate large buffers in intermediate switches along the path [11]. Moreover, in wormhole routing, message latency is less sensible to distance. However, it should be used with special care to avoid deadlock in the network. Deadlock is a situation when the network resources continuously wait for each other to be released. Routing algorithms are required to be deadlock-free and break all cyclic dependencies among channels. Virtual channels are usually used in the network both to avoid deadlock and increase fault tolerance, but it is an expensive solution. Traditional fault-tolerant schemes in NoCs have focused on rerouting packets around faulty regions, either convex or concave, so that the selected paths are not always the shortest one. However, detour strategy is a costly solution and considerably increases packet's latency and router's complexity. The inefficiency of these methods is mainly due to the fact that the information about faulty components is insufficient or the way of utilizing them is inefficient. In this paper, we present a fault-tolerant method which has several characteristics such as: 1- tolerating all one and two faulty links by using two virtual channels. It can also tolerate a large group of multiple faulty links; 2- reducing the packet latency by selecting a shortest path between each pair of source and destination nodes, if a shortest path exists. The rest of this paper is organized as follows: Section II reviews the related work. The turn model, fault distribution mechanism, and the proposed fault-tolerant algorithm are explained in Section III. The results are given in Section IV while we summarize and conclude in the last section.

II. Related Work

Fault-tolerant routing algorithms can be classified into two main groups: one can handle convex or concave shapes [12][13][14][15] and the other utilizes the contour strategy for addressing faults [16][17]. The basic assumptions in all of these methods are the permanent faulty cases. The methods in the first group are based on defining fault ring (f-ring) or fault chain (f-chain) around faulty regions where healthy nodes are disabled in order to form a specific shape. A reconfigurable routing algorithm using the contour strategy provides the possibility of routing packets through a cycle free surrounding a faulty component. The contour concept is firstly presented in [16] and the algorithm is able to tolerate all one-faulty routers in 2D mesh network without using virtual channels and disabling healthy nodes. However, to support more faulty routers, the contours must not be overlapped and thus faulty routers should be located far away from each other. This idea has been extended in [17] to tolerate two faulty links. The contour concept is efficient with a small number of faults while it becomes very complicated to support more faulty components.

In another classification, fault-tolerant routing algorithms could be divided into two classes: the methods using virtual channels [17][18] and those without using virtual channels [19][20]. In general, different methods define a new tradeoff between the numbers of virtual channels, the ability to handle different fault models, and the degree of adaptiveness.

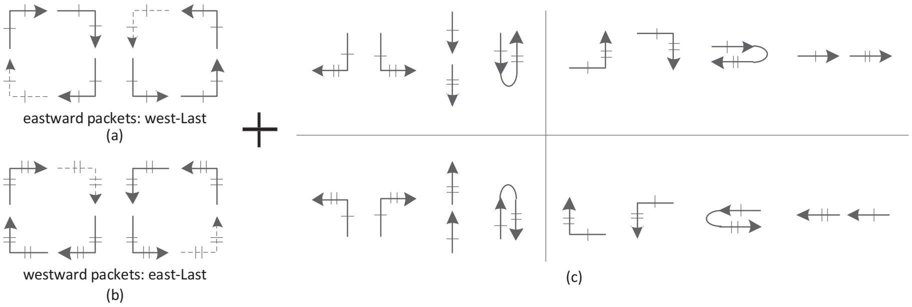

Fig. 1. (a) west-last (b) east-last (c) all permitted transactions between vc1 and vc2

The virtual channel-based fault-tolerant routing algorithms provide better fault-tolerant characteristics than those without virtual channels. The methods that do not use any virtual channel are mainly based on the turn models [21]. In turn models, some turns are eliminated in order to guarantee the deadlock freeness in the network and then the remaining turns are used both for routing packets and tolerating faults. In this paper, the fault information is distributed and utilized in such a way that packets can be routed through shortest paths in the presence of faults. This method can be used with any number of virtual channels in the network. However, in order to keep the area overhead low, we take use of two virtual channels. The basic version of the proposed method not only is able to tolerate all one and two faulty links but also all packets can be routed through shortest paths as long as any path exists. Then we improve the method to support multiple faulty links by taking advantage of non-minimal paths. To find non-minimal choices, we find out all possible transactions between two virtual channels, such that a packet being routed in one virtual channel can switch to the next virtual channel. It is worth mentioning that the presented method can operate in the presence of temporal or dynamic faults without creating any cycles in the network but tolerating less faulty cases compared with static cases.

III. The Proposed Approach

Fault-tolerant models with a small number of virtual channels are either deterministic or unable to handle different fault models. On the other hand, using a large number of virtual channels is not a cost efficient solution for on-chip networks. Traditional fault-tolerant algorithms are relatively complex due to considering different fault models and locations of faults. Our approach is general and can be applied to a network using any number of virtual channels. In this work, the proposed algorithm is based on using two virtual channels in each direction.

A. *The Turn Model and Deadlock Freedom*

To determine all allowable turns in the proposed method, at first, we assume that the network is divided into two separate subnetworks. All eastward packets are routed through the first subnetwork using the virtual channel 1 (vc1) while the westward packets are propagated in the second subnetwork utilizing the virtual channel 2 (vc2). Since the subnetworks are disjointed, any deadlock free routing algorithm can be applied to each of them and the network remains deadlock free. However, using the same routing algorithm for both channels leads to the prohibition of certain turns in the whole network. In order to avoid this situation, different turns are defined on each virtual channel, such that the prohibited turns in one virtual channel are permitted in the other one.

The turns to be prohibited in each virtual channel are inspired by the method in [23]. We have modified this method allowing the transaction between two virtual channels without forming any cycles. The allowable and unallowable turns in vc1 and vc2, similar to the method in [23], are shown in Fig. 1(a) and Fig. 1(b), respectively. The routing method used in the vc1 is west-last where the west-to-north and west-to-south turns cannot be taken by the packets. In the vc2, the east-last method is utilized where the east direction cannot be taken earlier than the other directions.

Eastward packets are routed in the first subnetwork to be able to use all minimal routes in the east direction. Moreover, the remaining turns are used for non-minimal routing if needed. Similarly, westward packets are routed in the second subnetwork to utilize all alternative paths to send packets in the west direction and the remaining turns are utilized for non-minimal purposes. Since the algorithms are deadlock free within each subnetwork, so that it is deadlock free in the whole network. However, many other turns (0-degree, 90-degree, 180-degree) can be also added into the list of allowable turns as shown in Fig. 1 (c). In fact, a cycle has not occurred if packets could switch from vc1 to vc2 but not vice versa. By using these additional turns, the network remains deadlock free since the starting and ending points of a set of packets always happens on different virtual channels. Now, we have an extensive set of allowable turns in the network to be used for tolerating faults.

B. *Fault Information Distribution Method*

Another idea behind our method relies on a new distribution mechanism of fault information and the method of utilization. As shown in Fig. 2, fault information is distributed in a way that each router is informed about the faulty links of its direct neighboring routers. For this purpose, each router transfers faulty information on its links to the neighbors. If E, W, N, and S stands for the packet direction in the East, West, North, and South directions, respectively, then each router has the information about the following links: E, W, N, S, EE, EN, ES, WW, WN, WS, NN, NE, NW, SS, SE, SW, ENW, ESW, WNE, WSE, NES, NWS, SEN, and SWN. For routing a packet in the northeast direction, a router uses the fault information on the links located in either minimal paths (i.e. EE, EN, NE, and NN) or non-minimal paths (e.g. SE, WN, WW, and SS). Similarly, for a northward packet, the fault's information on some links (e.g. N, ENW, and WNE, EE, and WW) is beneficial for making a reliable routing decision. Using this information, packets are possibly routed through minimal and non-faulty links which avoids facing with faulty links and making unnecessary routing around them. In sum, the idea is based on a common wise saying that “prevention is better than cure”.

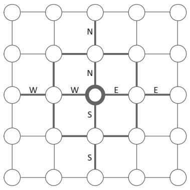

Fig. 2. Fault distribution mechanism

C. *Minimal and Adaptive Fault-Tolerant Algorithm*

We present a routing algorithm named Minimal and Adaptive Fault-Tolerant Algorithm (MAFA). The proposed method is not only adaptive but also is able to route packets through shortest paths, if any available. Then we improve the method and present Enhanced-MAFA to tolerate multiple faulty links. Using the explained turn model, all shortest paths in the east direction are valid for eastward packets. Similarly, westward packets can utilize all shortest paths in the west direction. Moreover, since the transaction from vc1 to vc2 is possible by MAFA, all the packets on vc1 can switch to vc2 whenever needed. It is worth mentioning that, using MAFA, all packets start routing in vc1.

According to MAFA if the distance from the current to destination node is greater than two hops along each direction, packets can adaptively choose among the non-faulty links. As shown in Fig. 5(a), when the destination is in the northeast of the current node, the packet can be delivered in either the north or east direction considering the availability of EE, EN, NE, and NN links (the pseudo code is illustrated in Fig. 3). If the distance of a packet reaches zero in one direction, the packet has to be routed along the other direction to reach the destination. Therefore, the packet must take a non-minimal path when it faces a faulty link. To overcome this situation, the presented algorithm avoids reducing the distance into zero in one direction when the distance along the other direction is greater than one. Generally, the method attempts to maintain the distances on the X and Y dimensions as equally as possible when routing packets. When the distance between the current and destination nodes reaches 1 in at least one dimension, at first all the possible shortest paths on the greater-distance dimension are checked. The packet is sent along the greater-distance dimension if any minimal and non-faulty paths exist; otherwise the links on the smaller-distance dimension are examined. The examples in Fig. 5(b) and Fig. 5(c), show a northeast packet where the distance over one dimension reaches one. In Fig. 5(b) the availability of NN and NE links is checked before those of EE and EN links while in Fig. 5(c) the conditions of EE and EN links are examined earlier than NN and NE links (the pseudo codes are illustrated in Fig. 3). Finally, as shown in Fig. 5(d), the distances along both directions are one. A packet can be delivered through one of shortest paths considering the faulty situation of NE and EN links. Otherwise it is sent in the north or east direction considering NN and EE links. Two examples of northward and eastward packets are illustrated in Fig. 5(e) and Fig. 5(f), which are supported by pseudo codes in Fig. 4. When there are several non-faulty minimal paths, the congestion value metric is used to select among them. The congestion value is defined as the number of free slots in the input buffers of the neighboring routers.

```
--northeast packet
IF (Dx=1 and Dy=1) THEN
     IF NE=’1’ and EN=’1’ THEN
         sel <= select north or east
         according to congestion values;
     ELSIF NE=’1’ THEN sel <= north;
     ELSIF EN=’1’ THEN sel <= east;
     ELSIF NN=‘1’ THEN sel <= north;
     ELSIF EE=‘1’ THEN sel <= east;
     END IF;
ELSIF (Dx=1 and Dy≥2) THEN
     IF (NE=’1’ or NN=’1’) THEN
          sel <= north;
     ELSIF (EN=’1’ or EE=’1’) THEN
          sel <= east;
     END IF;
ELSIF (Dx≥2 and Dy=1) THEN
     IF (EN=’1’ or EE=’1’) THEN
          sel <= east;
     ELSIF (NE=’1’ or NN=’1’) THEN
          sel <= north;
     END IF;
ELSIF (Dx≥2 and Dy ≥2) THEN
     IF (NN=’1’ or NE=’1’) AND (
         EN=’1’ or EE=’1’) THEN
          sel <= select north or east
          according to congestion values;
     ELSIF (NN=’1’ or NE=’1’) THEN
          sel<= north;
     ELSIF (EN=’1’ or EE=’1’) THEN
          sel <= east;
     END IF;
END IF;
```

Fig. 3. Pseudo code of delivering a packet in northeast direction

```
--northward packet
IF (N=’1’) THEN
     sel <= north;
ELSIF ENW=’1’ THEN
     sel <= east;
ELSIF WNE=’1’ THEN
     sel <= west; vc <= vc2;
ELSIF (EE=’1’ or EN=’1’) THEN
     sel <= east;
ELSIF (WW=’1’ or WN=’1’) THEN
     sel <= west; vc <= vc2;
END IF;
--eastward packet
IF (E=’1’) THEN
     sel <= east;
ELSIF NES=’1’ THEN
     sel <= north;
ELSIF SEN=’1’ THEN
     sel <= south;
ELSIF (NN=’1’ or NE=’1’) THEN
     sel <= north;
ELSIF (SS=’1’ or SE=’1’) THEN
     sel <= south;
END IF;
```

Fig. 4. Pseudo codes of delivering a packet in north or east direction

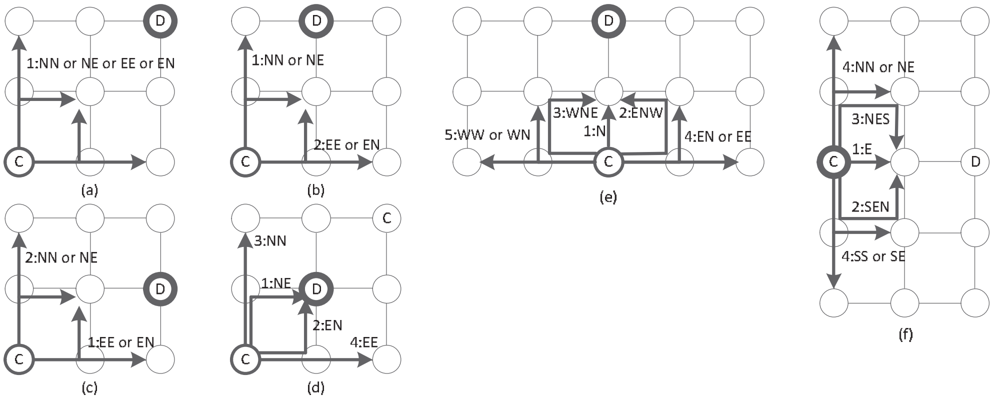

Fig. 5. (a),(b),(c), and (d) priority of minimal paths when destination is in northeast position, (e) north position, and (f) east position

Fig. 6 shows two examples of comparing MAFA with traditional methods which are based on contour strategy. In Fig. 6(a), a packet is sent from source S to destination D when the link (9,D) is faulty. Both methods deliver their packets from router S to router 4 according to the congestion condition. However, when packets arrive at router 4, the methods behave differently. MAFA avoids sending the packet to router 8 as the distance in one direction reaches zero. The other method makes the decision based on the congestion condition and may select router 8 as the next router. As it is obvious, the packet delivered to router 8 faces a faulty link at router 9 and has to be routed around it while the MAFA uses a shortest path to the destination.

In Fig. 6(b), two links are faulty in the network. Similar to the previous example, packets of both methods reach the router 5. MAFA is aware of the faulty links at NN and NE paths, so it sends the packet in the east direction instead of the north direction. As the other method does not use the fault information efficiently, it delivers the packet in the north direction where the packet is faced to faulty links and has to be returned to router 5.

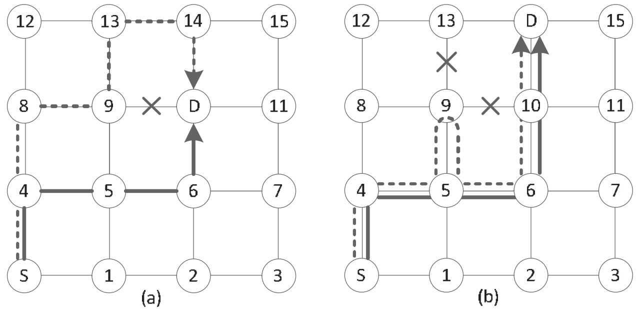

Fig. 6. Comparison of MAFA with detour-based methods

If all minimal paths are faulty, then the links on non-minimal paths should be examined, this has been implemented in Enhanced-MAFA. As MAFA is based on using minimal paths, it limits the possibility of routing packets through non-minimal paths. The aim of Enhanced-MAFA is to utilize the capability of the existing virtual channels for rerouting packets around faulty areas. When a packet enters a router through one of the input channels (i.e. L,N1,N2,S1,S2,E1,E2,W1, and W2), the routing unit determines one or several potential output channels to deliver the packet. The routing decision is based on the relative position of the current node and the destination node that is within one of the following eight cases: N, S, E, W, NE, NW, SE, and SW. A node receiving a packet needs to check for the faulty links and eligible turns prior connecting the input channel to the output channel. Although by all permitted turns shown in Fig. 1, the network is guaranteed to be deadlock free, however, they are not necessarily suitable choices for a packet at a node. The reason is that the packet may not be able to continue the path to the destination and is blocked. In Enhanced-MAFA, the potential output channels are selected in a way that not only the turn connecting the input channel to the output channel is permissible but also it is guaranteed that there is at least one non-faulty path from the next router to the destination. Fig. 7 presents the choices of output channels allowed by MAFA. According to MAFA, when a packet is generated at a router destined the northeast direction, at first EE, EN, NE, and NN links are checked to know whether they are faulty or not. If all minimal directions are faulty, the links on non-minimal directions, NW, SS, SE, SW, ES, WW, WN, and WS are examined. Note that the conditions on the borderline routers are slightly different and availability of the links should be examined in a special order.

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 [24]. For all the routers, the data width is set to 32 bits. The congestion threshold value is set to 3, meaning that a buffer is considered as a congested one when 3 out of 5 buffer slots are occupied. 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 PE. The request rate is defined as the ratio of the successful message injections into the network over the total number of injection attempts. The simulator is warmed up for 12,000 cycles and then the average performance is measured over another 200,000 cycles.

To have a fair comparison, we defined our baseline as a detour strategy similar to [17]. Like MAFA, the baseline method has two virtual channels and uses the allowable turns as in Fig. 1. Moreover, it is an adaptive method and the decision on the next hop is made based on the congestion condition of the corresponding input buffers at the neighboring nodes. Unlike MAFA, the baseline method may take unnecessary longer paths as discussed in the example of Fig. 6.

```
--Inport: input port;  dLoc: destination location

If Inport=L and dLoc={N,S,E,W,NE,NW,SW,SE} then
   Select <= NN,NE,NW,SS,SE,SW,EE,EN,ES,WW,WN,WS;

If Inport=N1 and dLoc={N,S,E,W,NE,NW,SW,SE} then
   Select <= NN,NW,SS,SE,SW,EE,EN,ES,WW,WN,WS;

If Inport=N2 and dLoc={N,S,W,NE,NW,SE,SW} then
   Select <= WW,WN,WS,SS,SW;
If Inport=N2 and dLoc={E} then
   Select <= EE;

If Inport=S1 and dLoc={N,S,W,NW,SW} then
   Select <= NN,NE,NW,SS,SW,EE,EN,ES,WW,WN,WS;
If Inport=S1 and dLoc={E,NE,SE} then
   Select <= NN,NE,EE,EN,EN,ES;

If Inport=E1 and dLoc={N,S,W,NE,NW,SE,SW} then
   Select <= NN,NW,SS,SW,WW,WN,WS;
If Inport=E1 and dLoc={E} then
   Select <= EE;

If Inport=E2 and dLoc={N,S,W,NE,NW,SE,SW} then
   Select <= NN,NW,SS,SW,WW,WN,WS;

If Inport=W1 and dLoc={N,S,W,NE,NW,SE,SW} then
   Select <= NN,NE,NW,SS,SE,SW,EE,EN,ES,WW,WN,WS;
If Inport=W1 and dLoc={E} then
   Select <= NN,NE,SS,SE,EE,EN,ES;

If Inport=W2 and dLoc={E} then
   Select <= EE;
```

Fig. 7. Non-minimal choices offered by Enhanced-MAFA

A. *Performance Analysis under 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 [21][22]. The mesh size is considered 4×4. In Fig. 8(a), the average communication latencies of the Enhanced-MAFA and baseline methods are measured for fault-free, one-faulty and two-faulty link cases. As observed from the results, in one-faulty and two-faulty cases, Enhanced-MAFA can reduce the latency compared with the baseline method. This is due to the fact that Enhanced-MAFA can route packets through minimal paths while in the baseline method, packets may take longer paths when facing a faulty link. However, in the fault-free case, the baseline methods performs better, since in Enhanced-MAFA packet adaptively is limited when packets get close to the destination node. In other situations, both methods have the same degree of adaptiveness.

B. *Performance Analysis under Hotspot Traffic Profile*

Under the hotspot traffic pattern, one or more 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 a single hotspot node at (2, 2) in 4×4 2D-mesh. The performance of the Enhanced-MAFA and the baseline method is also measured for fault-free, one-faulty and two-faulty link cases. The performance of each network with *H = 10%* is illustrated in Fig. 8(b). As observed from the figure, in the hotspot traffic and in all faulty cases, the performance improvement of MAFA is better than the detour-based scheme.

C. *Reliability Evaluation under uniform traffic profile*

To evaluate the reliability of Enhanced-MAFA, the number of faulty links increases from 1 to 6. All faulty links are selected using a random function. The results are obtained using 10000 iterations in a 6×6 mesh network when the traffic is uniform random. A network is reliable if all the injected packets reach their destinations. As shown in Fig. 9, Enhanced-MAFA can tolerate up to 6 faulty links by more than 91% reliability.

D. *Hardware Analysis*

To assess the area overhead and power consumption, the whole platform of each method is synthesized by Synopsys Design Compiler. We compared the area overhead and power consumption of Enhanced-MAFA with the baseline and DyXY methods [25]. DyXY also uses two virtual channels per dimension. The power consumption of DyXY is measured only in the fault-free case. Each scheme includes network interfaces, routers, and communication channels. For synthesizing we use the UMC 90nm technology at the operating frequency of 1GHz and supply voltage of 1V. We perform place-and-route, using Cadence Encounter, to have precise power and area estimations. The power dissipation is calculated using Synopsys PrimePower in a 6×6 2D mesh. The layout area and power consumption of each platform are shown in Table 1. As indicated in the table, Enhanced-MAFA has a larger area overhead than DyXY and a lower one than the baseline method. It is because of using a simpler routing unit at DyXY method and a more complex one in the baseline method. As indicated in the table even if the Enhanced-MAFA has to support a one-faulty link (while DyXY is fault-free), the power consumption of Enhanced-MAFA remains relatively small. This is due to the fact that Enhanced-MAFA could route packets through shortest paths and thus consuming less power.

V. CONCLUSION

In this paper, a fault-tolerant routing algorithm was presented using two virtual channels. The prohibited and permitted turns on each virtual channel are determined in such a way that prohibited turns in one virtual channel are permitted in the other one. Unlike traditional methods, packets are not limited to use only one virtual channel and can switch from the first to the second virtual channel. This improvement offers a large degree of adaptiveness for both minimal and non-minimal paths in order to tolerate multiple faulty links in the network. Another contribution of this paper is to propose a new fault's information propagation mechanism and utilize the information to deliver packets through shortest paths. To increase the reliability, the method takes advantage of non-minimal paths when all minimal paths are congested.

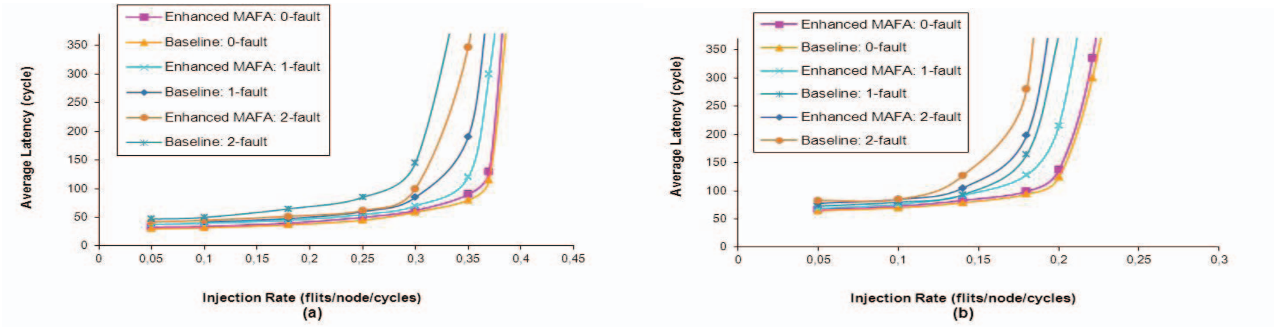

Fig. 8. Performance analysis of Enhanced-MAFA and the baseline method in 4×4 mesh network (a) under uniform traffic profile (b) hotspot traffic profile in fault-free, 1-faulty and 2-faulty cases.

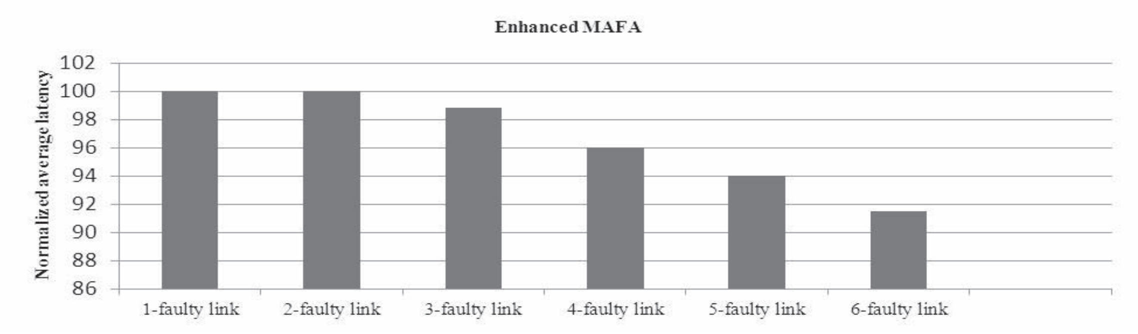

Fig. 9. Reliability evaluation of Enhanced-MAFA in 6×6 mesh network under uniform traffic profile

Table 1. Hardware implementation details

Network platforms	Area (mm^2)	Power (W) dynamic & static
DyXY	6.710	2.32
Enhanced_MAFA	6.794	2.41
baseline	6.913	2.78

REFERENCES

[1] W. Tsai, D. Zheng, S. Chen, and Y.H. Hu, "A fault-tolerant NoC scheme using bidirectional channel", in Proc. 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. DATE'03, pp. 350-355, 2003.

[3] T. C. Xu, P. Liljeberg, H. Tenhunen, "An Optimized Network-on-Chip Design for Data Parallel FFT," in Procedia Engineering, Vol. 30, pp. 313-318, 2012.

[4] M. Daneshtalab et al., "Adaptive Input-output Selection Based On-Chip Router Architecture," Journal of Low Power Electronics (JOLPE), Vol. 8, No. 1, pp. 11-29, 2012.

[5] M. Cuviello, S. Dey, X. Bai, and Y. Zhao, "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects", in Proc. of the IEEE/ACM International Conference on Computer-Aided Design, pp. 297-303, 1999.

[6] T. C. Xu et al., "A Greedy Heuristic Approximation Scheduling Algorithm for 3D Multicore Processors," in Proceedings of the 17th International Euro-Par Conference, Parallel Processing Workshops (Euro-Par), LNCS 7155/2012, pp.281-291, 2011.

[7] J. Duato, S. Yalamanchili, L. Ni, "Interconnection networks: an engineering approach", Morgan Kaufmann Publishers, 2003.

[8] M. Ebrahimi et al., "CATRA-Congestion Aware Trapezoid-based Routing Algorithm for On-Chip Networks," in Proceedings of 15th ACM/IEEE Design, Automation, and Test in Europe (DATE), pp. 320-325, Mar. 2012, Germany.

[9] M. Dehyadegari et al., "An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip," in Proceedings of 13th IEEE/NASA-ESA International Conference on Adaptive Hardware and Systems (AHS), pp. 208-214, June 2011, USA.

[10] M. Ebrahimi et al., "An Efficent Dynamic Multicast Routing Protocol for Distributing Traffic in NOCs," in Proceedings of 12th ACM/IEEE Design, Automation, and Test in Europe Conference (DATE), pp. 1064 - 1069 , April 2009, France.

[11] L.M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in direct networks", in Proc. IEEE Computer, v.26, I.2, pp.62-76, 1993.

[12] D. Fick et al. "Vicis: a reliable network for unreliable silicon", in Proc. of Design Automation Conference, pp. 812-816, 2009.

[13] S. Chalasani, R.V. Boppana, "Fault-tolerant wormhole routing algorithms for mesh networks", IEEE Trans on Computers, 44(7):848–64, 1995.

[14] PH. Sui, SD. Wang," An improved algorithm for fault-tolerant wormhole routing in meshes", IEEE Trans on Computers x;46(9):1040–2, 2011.

[15] S. Park, JH. Youn, B. Bose, "Fault-tolerant wormhole routing algorithms in meshes in the presence of concave faults", in Proc. of International

Parallel and Distributed Processing Symposium (IPDPS), p. 633–8, 2000.

[16] Z. Zhang, A. Greiner and S. Taktak, “A reconfigurable routing algorithm for a fault-tolerant 2D-mesh Network-on-Chip”, in Proc. DAC, pp. 441-446, 2008.

[17] M. Valinataja, S. Mohammadi, J. Plosila, P. Liljeberg, H. Tenhunen,”A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip”, in Proc. International Journal of Electronics and Communications (AEU), v. 65, I.7, pp. 630-640, 2011.

[18] M. Koibuchi, H. Matsutani, H. Amano, and T.M. Pinkston, “A Lightweight Fault-Tolerant Mechanism for Network-on-Chip”, in Proc. NOCS, pp.13-22, 2008.

[19] J. Wu, “A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model”, in Proc. IEEE transaction on computers, v. 52, pp.1154-1169 ,2003.

[20] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester and D. Blaauw, “A highly resilient routing algorithm for fault-tolerant NoCs”, in Proc. DATE, pp. 21-26, 2009.

[21] C.J. Glass et al., “The Turn Model for Adaptive Routing”, in Proc. 19th Int'l Symp. Computer Architecture, pp. 278-287, 1992.

[22] T. C. Xu, P. Liljeberg, H. Tenhunen, “Optimal Memory Controller Placement for Chip Multiprocessor,” In Proceedings of the 9th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS), pp. 217-226, 2011, Taiwan.

[23] V. Soteriou and L.-S. Peh. “Dynamic power management for power optimization of interconnection networks using on/off links”, in Proc. of the 11th Symposium on High Performance Interconnects, 2003.

[24] M. Ebrahimi et al., “Agent-based On-Chip Network Using Efficient Selection Method,” in Proceedings of 19th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 284-289, Oct 2011, Hongkong.

[25] M. Li, Q. A. Zeng and W. B. Jone, “DyXY - A proximity congestionaware deadlock-free dynamic routing method for Network on Chip”, in Proc. DAC, pp. 849-852, 2006.