

Evaluating the Performance of Adaptive Fault-Tolerant Routing Algorithms for Wormhole-Switched Mesh Interconnect Networks*

F. Safaei^{1,3}, A. Khonsari^{2,1}, M. Fathy³, A. H. Shantia⁴, M. Ould-Khaoua⁵

¹IPM School of Computer Science, Tehran, Iran

²Dept. of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

³Dept. of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

⁴Islamic Azad University, North Branch, Tehran, Iran

⁵Dept. of Computing Science, University of Glasgow, UK

{safaei, ak}@ipm.ir, {f_safaei, mahfathy}@iust.ac.ir, shantia_amirhosein@computer.org, mohamed@dcs.gla.ac.uk

Abstract

One of the fundamental problems in parallel computing is how to efficiently perform routing in a faulty network each component of which fails with some probability. This paper presents a comparative performance study of ten prominent adaptive fault-tolerant routing algorithms in wormhole-switched 2-D mesh interconnect networks. These networks carry a routing scheme suggested by Boppana and Chalasani [1] as an instance of a fault-tolerant method. The suggested scheme is widely used in the literature to achieve high adaptivity and support inter-processor communications in parallel computer systems due to its ability to preserve both communication performance and fault-tolerant demands in these networks. The performance measures studied are the throughput, average message latency and average usage of virtual channels per node. Results obtained through simulation suggest two classes of presented routing schemes as high performance candidate in most faulty networks.

1. Introduction

The advent of semiconductor circuit and the development with great speed of the modern communication technology has made it possible to construct and design more complicated, more convenient economical high performance computers and very complex massive interconnect networks. These networks connect tens of thousands Processing Elements (PEs) for communication networks which are capable of executing parallel algorithms.

Mesh networks are among the most important and popular interconnect network topologies for large-scale parallel computer systems. The advantages of mesh networks include their simplicity, regularity, and good

scalability. A number of large research and commercial large-scale parallel systems have been built based on the mesh topologies [2].

Since processors in a large interconnect network need to communicate with each other, efficient communication is essential to enhance the performance of the system. To decrease the amount of time in transmitting data, these systems adopt a wormhole switching mechanism [2, 3]. With wormhole switching, a message is divided into a sequence of fixed-size units of data, called flits. The header flit of a message contains all of the information needed to decide about the selection of next channel on the route. As the header flit advances, the remaining flits follow it in a pipeline fashion. When the header flit reaches a node that has no suitable output channel available, all of the flits in the message are blocked until the channel is freed. This form of routing technique makes the message transmission time almost independent from the distance between two nodes if the network is contention-free.

Two important issues in designing routing algorithms are deadlock and livelock freedom. These properties are necessary for the guaranteed delivery of a message to its destination. Another issue in designing routing algorithm is designing a fault-tolerant mechanism that can route messages in the presence of faulty components. An extensive amount of work has been done on fault-tolerant routing in mesh networks [1, 3-7]. Boura and Das [7] proposed an adaptive deadlock-free fault-tolerant routing algorithm for meshes. A node labeling technique is used to identify nodes that may cause routing difficulty. Messages are routed adaptively in healthy regions. This method can tolerate any number of faults by using three virtual channels per physical channel. Boppana and Chalasani [1] have proposed an efficient routing scheme using

* This research was in part supported by a grant from I.P.M. (No. CS1385-2-02).

wormhole switching to enhance current routing algorithms to handle a rectangular (or block) fault model for mesh networks. The concepts of fault-ring (f-ring) and fault-chain (f-chain) are introduced and used for routing messages around the fault regions. In this paper, the performance vicissitudes of ten routing algorithms is investigated in wormhole-switched 2-D mesh networks based on a routing scheme suggested by Boppana and Chalasani as an instance of a fault-tolerant routing methodology widely used in the literature for supporting high adaptivity and inter-processor communications in such systems. Two of these algorithms are the basis of the other four algorithms. The routing algorithms used are the Positive-Hop (PHop), Negative-Hop (NHop), Duato's routing, Minimal-Adaptive, Fully-Adaptive, and Boura's routing algorithm [7]. The other four algorithms are resulted from some modifications on the two basic algorithms.

The rest of the paper is structured as follows. Section 2 describes the necessary information to understand the paper. Section 3 explains briefly the basic adaptive routing algorithms, fortified with the Boppana-Chalasani's scheme. Four sets of modified routing algorithms are introduced in Section 4. Section 5 gives simulation results on the performance of these routing algorithms in the presence and absence of faults. This section also characterizes the distribution of traffic load around fault-rings. Conclusions and possible directions for future work are presented in Section 6.

2. Preliminaries

This section briefly describes 2-D mesh and then explores the Boppana-Chalasani's routing algorithm.

2.1 The Mesh topology

The mesh networks frequently appear in various applications of networks. The topological structure of a mesh network is defined as the Cartesian product $P_l \times P_m$ of undirected paths P_l and P_m , denoted by $G(l, m)$ [2]. Sometimes, $G(l, m)$ is called an $l \times m$ mesh in the literature. The $G(k, k)$ is a 2-D $k \times k$ mesh with $N=k^2$ nodes has an interior node degree of 4 and a network diameter of $2(k-1)$. Each node u has an address (u_x, u_y) , where $u_x, u_y \in \{0, 1, 2, \dots, k-1\}$. Two nodes $u: (u_x, u_y)$ and $v: (v_x, v_y)$ are connected if their addresses differ in one and only one dimension, say dimension x ; moreover, $|u_x - v_x| = 1$. Similarly, if they differ in dimension y , then $|u_y - v_y| = 1$. The mesh topology is inherently asymmetric as a result of the absence of the wrap-around connections along each dimension.

2.2 Fault models

Many applications of interconnect networks require high reliability and availability. A large parallel computer requires that its interconnect network operates without packet loss for ten thousands of hours. Thus, these networks must employ error control mechanism to continue operation without interruption, and possibly without packet loss, despite the failure of a component. In a network, there exist two classes of faults: either the entire PE along with its associated router can fail or just a physical link may fail. The former is referred to as a *node failure*, and the latter as a *link failure* [2]. On a node failure occasion, all physical links incident on the failed node are also marked faulty at adjacent routers [1, 2, 3-6]. Adjacent faulty nodes are coalesced into *fault regions*, which may lead to different patterns of failed components. Faulty regions, extended by faulty components, may form convex (also known as block faults) or concave shape [2, 6, 8].

We, in this paper, will focus on the block (convex) fault model, which is suitable for modeling faults at the chip, multichip modules, and board level in networks with grid structures, particularly in the mesh topologies. Furthermore, our approach tolerates only node failures since the structure of nodes are more complex than links, and thus, possess higher failure rates [1, 2, 6, 8]. We assume that fault patterns are static and do not *disconnect* the network. A network with faulty nodes is disconnected (by faulty nodes) if there are non-faulty nodes u and v in the network such that no fault-free path from u to v can be found in the network [1, 3, 6].

2.3 The Boppana-Chalasani's routing scheme

Boppana and Chalasani [1] have presented an efficient method to enhance the current wormhole-switched routing algorithms developed for high radix, low dimensional mesh networks for fault-tolerant routing. They considered arbitrarily located faulty blocks and assumed only local knowledge of faults. Messages are routed minimally when not blocked by faults. This constraint is relaxed to facilitate routing in the face of faults. The key concept they used is an f-ring consisting of fault-free nodes and physical rings which can be formed around each fault region. Their fault-tolerant scheme uses these f-rings to route messages around fault regions. They showed that at most four additional virtual channels are sufficient to make any adaptive algorithm tolerate multiple faulty blocks in mesh networks. They have also shown that for a fully adaptive algorithm fortified with their method, good

performance may be obtained with as many as 10% failures.

3. Fault-tolerant routing algorithms in mesh networks

In this section, we address the issue of incorporating fault-tolerance into fully adaptive wormhole-switched routing algorithms which can tolerate multiple block faults in a mesh network. Our approach is to employ a number of prominent adaptive algorithms as much as possible to route messages. When a message is blocked by faults at node u and there is no fault-free link $\langle u, v \rangle$ such that the link from u to v is along the shortest path, the additional routing scheme is needed to route the message around the fault regions. To this end, we incorporate the routing scheme suggested by Boppana and Chalasani, as a well-known instance of a fault-tolerant method widely used in the literature to enhance adaptive routing algorithms for fault-tolerant routings. To the best of our knowledge, this work is the first effort in characterizing a variety of adaptive routings regarding their fault-tolerant performance and design trade-offs. To further illustration of our evaluation, we consider five well-known basic routing algorithms: two hop-based routings (PHop, NHop), Duato's methodology, Minimal-Adaptive, Fully-Adaptive, and three other sets of improved algorithms based on the basic routings in addition to showing how the system performance is affected by these routing algorithms. Boura and Das [7] also presented a fault-tolerant routing scheme in which a node labeling technique was used. We also compare the mentioned above algorithms with their routing scheme.

In the PHop algorithm [9], the number of buffer classes in each node is equal to the diameter of the network plus one. Thus, for n -D mesh networks the number of buffer classes is equal to $n(k-1)+1$. A message, as soon as generated, is placed in the buffer of class 0 in the source node. During the course of its crossing towards its destination, the message occupies a buffer of class i at an intermediate node if and only if the message has taken exactly i hops to reach that intermediate node. A disadvantage of this algorithm is that it requires a large number of buffer classes. The NHop routing algorithm is discussed in [9]. To employ this algorithm, the network is colored, and each node is given a label corresponding to its color. A hop by a message is a negative hop if it moves from a node with higher label to a node with lower label. Any other hop is a non-negative hop. Messages when injected to the network have 0 negative hops and are routed minimally when there are no faults. If a message has taken $i \geq 0$ negative hops, then it uses virtual channels of class i for

its next hop. The NHop provides minimal fully adaptive routing in fault-free mesh using $1 + \lfloor n(k-1)/2 \rfloor$ virtual channels. The advantage of the NHop scheme is that it requires fewer buffer classes than does the PHop. In our simulations, we have used the NHop augmented with four additional virtual channels which incorporates the fault-tolerant scheme proposed by Boppana and Chalasani [1].

4. Modifications of the basic routing algorithms

Two hop-based fully adaptive algorithms (i.e., PHop and NHop) described in the previous section do not utilize virtual channels because of starting their journey originating from virtual channel 0. However, very few packets take the maximum number of hops (network diameter) and use all the virtual channels. Virtual channels with lower number are utilized more than virtual channels with higher numbers. In this study, the channel usage is identical for all traffic generation rates and any virtual channel set size.

The PHop and NHop routings can be modified by giving each header flit a number of *bonus cards*. In the PHop routing algorithm with bonus cards (which referred as Pbc), the number of bonus cards equals the diameter of the network minus the number of hops it is going to take to reach the destination. It is equal to the maximum possible of negative hops minus the number of required negative hops to reach the destination for the NHop scheme with bonus cards (which referred as Nbc). A message with no bonus cards is routed exactly the same way in the cases of PHop and NHop algorithms. In routing a message with b bonus cards, ($b \geq 1$), any of virtual channels numbered 0, 1, ..., b can be used for the first hop of the message. Thus, a message with bonus cards has a wider choice of virtual channels and is likely to choose the least congested one for the first hop.

4.1 Increasing adaptivity of the routing algorithms

We have used Duato's methodology [2, 10] for designing fully adaptive routing algorithms to further improve the adaptivity of the basic algorithms (PHop and NHop) and those improved with bonus cards (Pbc and Nbc). Duato [10] has described an adaptive routing algorithm that allows efficient router implementation due to its low requirement for virtual channels. The algorithm divides the virtual channels into two classes: I and II. At each routing step, a message can adaptively visit any available virtual channel from class I. If all the virtual channels belonging to class I are busy, it crosses

a virtual channel from class II using a deadlock-free routing algorithm described in Section 2 (basic and modified ones with bonus cards). If we employ PHop and NHop routing schemes (with or without bonus cards), at least $n(k-1)+1$ and $1+\lfloor n(k-1)/2 \rfloor$ virtual channels are required for class II, respectively. For example, to implement the PHop and NHop algorithms on a 10×10 mesh, these values are equal to 19 and 10, respectively. Network performance is maximized when the extra virtual channels are added to adaptive virtual channels in class I [2, 10]. Therefore, the best performance is achieved when class II contains minimum required virtual channels and extra virtual channels are allocated to class I.

5. Simulation results

To study the performance issues, we have developed a flit-level simulator. This simulator can be used for wormhole switching in 2-D meshes with and without faults. The crossbar switch in the router allows multiple messages to traverse a node simultaneously. It takes one network cycle to transmit a flit between neighbors. Mesh size simulated was 10×10 since radix 10 has been used in many previous studies [1, 3, 7, 9]. Up to 10% of faults were simulated in the experiments. In the literature, fixed-length message with 32, 64, or 100 flits are commonly considered [1, 3-7, 9]. We have used 100-flit messages in this study. In the simulation, we used a uniform traffic pattern; that is, a processor sending a message to any other active nodes with equal probability. Messages were generated at time intervals chosen from an exponential distribution. The processors routed messages in an asynchronous manner. Conflicts of requests for an output channel by multiple messages were resolved in a random manner. We consider only node failure assuming that a faulty node is randomly generated subject to the fault model. We also assume that faults are non-malicious, fault patterns are static [1-7] and do not disconnect the network. Therefore only non-faulty nodes generate messages. Further, messages are destined only to fault-free nodes. These assumptions are commonly made in fault analysis in the literature [1, 3-6, 8-10]. For each given number of faults, 1000 fault patterns were randomly selected for simulation. For each fault pattern, a simulation was run for a total of 30,000 cycles. Performance data were not collected in the first 10,000 cycles to allow the system to stabilize.

Figure 1 shows the saturation throughput of all algorithms. Algorithms can be divided into two categories. The first category consists of algorithms that are completely free in choosing the virtual

channels and the second category consists of those that are not free in choosing the virtual channels and they should follow specific rules. Concerning the fact that all the discussed algorithms are adaptive, their routing functions to find an appropriate path are the same and they are different only in the way of using the virtual channels. In the first category, Duato's routing, Fully-Adaptive, Boura's routing, and Minimal-Adaptive algorithms are to be found. The second category consists of Duato-Nbc, Duato-Pbc, Nbc, Pbc, NHop, and PHop, the first four algorithms of which are placed between the first and the second categories. In the first category, regarding that the selecting rule of each virtual channel is the same as to other virtual channels, the amount of saturation throughput is affected by the number of virtual channels, not by the way of using them. But in the second category in which using the virtual channels is dependent from the state of messages in the network, it will have a direct impact on the throughput of each algorithm. Therefore, the algorithms that employ the virtual channels with high regularity (i.e., the attempt for the balanced use of virtual channels such as Duato-Nbc routing) get better throughput, while an algorithm like PHop has less throughput due to the unbalanced use of the virtual channels.

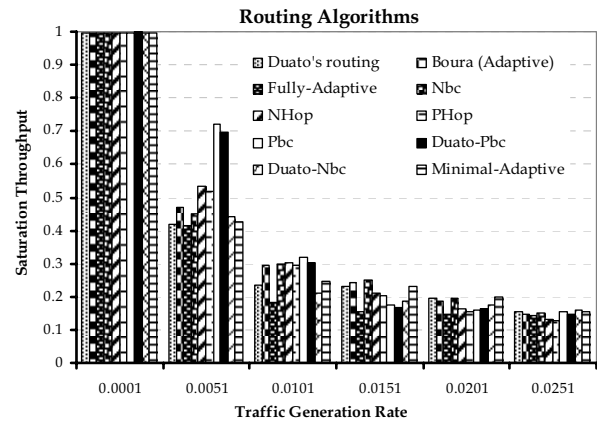


Figure 1: Comparison between the throughput of routing algorithms against the traffic load in a 10×10 mesh with 100-flit message length and 24 virtual channels per physical channel.

Another interesting point can be seen in Figure 1. Although the NHop algorithm uses quite similar supervision comparing to PHop in using virtual channels, it has better situation comparing with PHop. The reason of this is that, NHop requires at least 10 virtual channels in a 10×10 mesh network, so we have given it 12 classes of virtual channels in which each class contains 2 virtual channels to obtain approximately reasonable condition between all the

different algorithms. In each class, NHop is unrestricted for choosing any virtual channel, i.e., NHop algorithm has better throughput by using more buffer classes and it will get closer to the behavior of the Nbc algorithm. Hence, the algorithms that use the virtual channels in a regular manner will have a raise in their throughput. This case is not true about the first category.

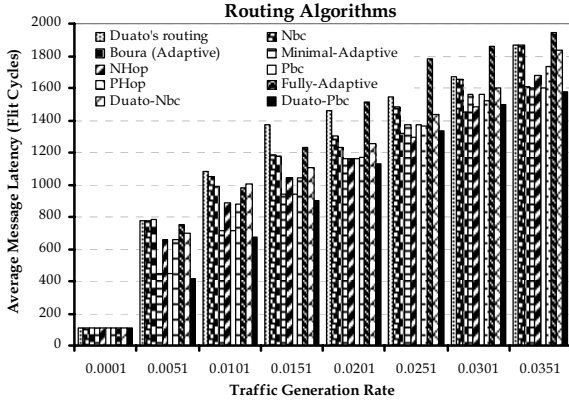
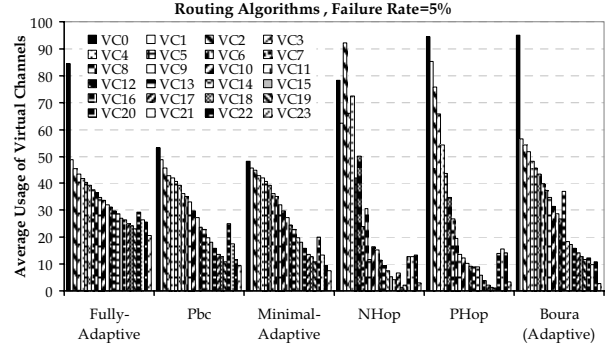
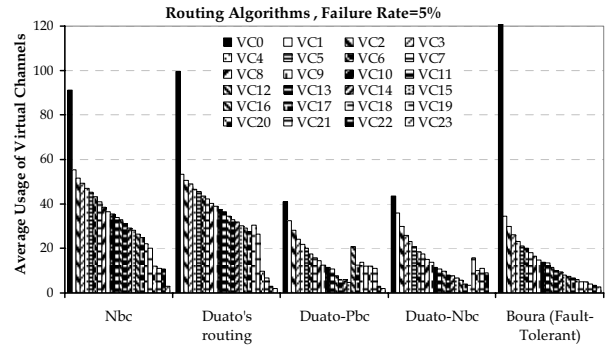


Figure 2: The average message latency of adaptive routing algorithms against the traffic load in a 10×10 mesh using 100-flit message length and 24 virtual channels per physical channel.

The average message latency of the routing algorithms for a 10×10 mesh is demonstrated against the offered traffic in Figure 2. In this configuration, we need 19 and 10 virtual channels classes for PHop and NHop algorithms, respectively. We have also used 1 and 2 virtual channels in each class, and four additional virtual channels for Boppana-Chalasani's routing scheme, respectively. Therefore, in this network each physical channel has 24 virtual channels which can ensure a fair comparison under almost equal hardware cost. Network configuration of Pbc and Nbc algorithms is the same as that of PHop and NHop routing algorithms. Figure 2 demonstrates that for low and medium traffic loads all six routing algorithms have the same latency, but they begin to behave differently around the saturation region. The basic algorithms with bonus cards (i.e., Pbc and Nbc) have better performance than the two basic algorithms, since we used the Pbc and Nbc algorithms for routing messages using class I virtual channels. This figure also reveals that under an equal number of virtual channels, Duato-based improved routings have higher throughputs and lower latencies compared to the basic algorithms with bonus cards. It can be seen that Duato's methodology with Nbc (Duato-Nbc) gets better performance due to the fact that in Nbc routing algorithm there are more virtual channels in the adaptive class (i.e., class I).



(a)



(b)

Figure 3: Virtual channel utilization under uniform traffic in a 10×10 mesh for adaptive routing algorithms with 100-flit message length and 24 virtual channels per physical channel; (a) Basic routing algorithms, (b) Nbc, Boura's fault-tolerant routing, and Duato's routing with Nbc and Pbc.

In Figure 3 the difference between the amounts of using virtual channels in the fault-free case and in the case in which there is 5% faults in the network has been illustrated. In this figure, virtual channels 0 (VC_0) and 1 (VC_1) belong to class I and other virtual channels belong to class II. As it can be seen, many variations have been made in the algorithms of the first category because of their flexibility to use the virtual channels (like for example Duato's routing). Among the second group of algorithms, those having more flexibility in choosing the virtual channels bear more variation which has caused the traffic of messages in the network more balanced in the amount of using virtual channels. The explanation of such phenomenon is that because of getting more congested and more usage of some virtual channels used through the Boppana-Chalasani's scheme, those algorithms have come towards other virtual channels resulting in the distribution of traffic load in the network. Therefore, these algorithms get controlled better than those such as PHop and NHop which do not have further flexibility in choosing the virtual channels. Albeit, the considered difference in NHop and PHop algorithms is only because of the

Boppana-Chalasani's scheme using these virtual channels. As the figure demonstrates, however, the state of NHop is better than PHop. This problem is because of the virtual channels classification in buffer classes (2 virtual channels per each class) and the flexibility in choosing virtual channels. Furthermore, the Boppana-Chalasani's algorithm uses 4 additional virtual channels. This adaptivity in using virtual channel classes has caused less effect on the NHop comparing to the PHop using 19 virtual channel classes. The Minimal-Adaptive routing is not different from other adaptive algorithms in the type of choosing the shortest path, but it surely uses the selected path and it does not apply a specific supervision in the way of using virtual channels. The message latency and the throughput of this algorithm are heavily dependent on the number of used virtual channels. The Fully-Adaptive algorithm is similar to the Minimal-Adaptive, but it may provide additional path diversity that can be used to avoid congestion. It can lead to livelock unless measures are taken to guarantee progress. Fully-Adaptive routing may misroute messages when it finds that all the virtual channels to be visited in the shortest path are busy. One approach is to allow misrouting only a fixed number of times. Consequently, the number of the misroutes is limited and is set to 10 in the simulation.

5.1 Performance study

We have simulated a 10×10 mesh with 5% and 10% of total faulty nodes in the network. In each case, we have randomly generated the required number of faulty nodes. To see the performance degradation with faults, we have also simulated the routing algorithms on a fault-free mesh. The simulation results reported in this section are for the mentioned above algorithms fortified with Boppana-Chalasani's routing scheme and the Boura's fault-tolerant routing algorithm. Comparative performance across different fault cases is in accordance with the fault sets used. Therefore, we simulated the routing algorithms for mentioned above fault cases. For each case, we have simulated 10 different fault sets for 100% traffic load. The values obtained from 10 different fault sets are averaged and shown in Figures 4 and 5. Two most important performance measures are message latency and network throughput. We use normalized throughput and normalized message latency as the performance metrics. The normalized throughput is equal to the number of messages received over the number of messages that can be transmitted at the maximum load [2]. The normalized throughput and the normalized message latency are plotted against percentage of faulty

nodes in Figures 4 and 5 for uniform traffic using 100-flit messages and 24 virtual channels per physical channel.

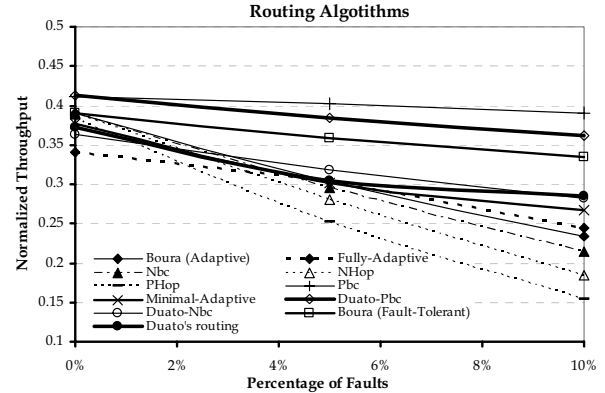


Figure 4: Comparison between the throughput of routing algorithms against the traffic load for a 10×10 mesh using 100-flit message length, 24 virtual channels per physical channel, and various fault cases 0%, 5%, and 10%.

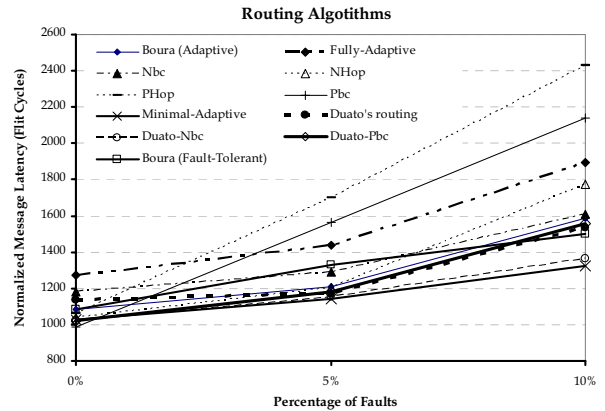


Figure 5: The normalized message latency of routing algorithms in a 10×10 mesh with 100-flit message length, 24 virtual channels per physical channel, and various fault cases 0%, 5%, and 10%.

For low traffic load, all ten algorithms have approximately the same latency. However, the six hop-based schemes and the other algorithms behave differently during and after saturation. The two algorithms achieved from store-and-forward switching, have similar throughputs with NHop being slightly better: NHop has better latency and throughput in saturation. In particular, NHop starts to saturate after 0.066 and PHop shows signs of saturation at about 0.045. The latencies of other algorithms rise abruptly at the point of saturation. Furthermore, the achieved throughputs of the three algorithms increase steadily. The NHop and Duato-Nbc algorithms achieve their peak throughputs of 0.389 and 0.363, respectively, at 100% traffic load, but the Nbc algorithm does not exhibit better performance than the Duato's algorithm.

The Pbc routing has higher peak throughput than the other routings, while the Fully-Adaptive has lower peak throughput and saturates more quickly. Another observation from figures is Duato's routing performs better than Minimal, Fully-Adaptive, and Boura's routing algorithms. Moreover, the fully adaptive hop-based schemes augmented with Duato's routing (i.e., Duato-Pbc, and Duato-Nbc) yield better throughput and message latency compared with the other algorithms for the conditions considered in this study. This could be due to the use of more virtual channels per physical channel in class II, balancing the traffic load on virtual channels. For instance, Duato-Pbc gives better throughput and also uses more virtual channels than any other algorithm.

5.2 Traffic analysis of routing algorithms around fault-rings

In regular networks with uniform traffic, network resources are used evenly among all the nodes. However, in the presence of faults, traffic is jammed in some regions of network and creates bottlenecks. These local bottlenecks may degrade the overall performance of network especially in case of wormhole-switched networks. These bottlenecks may cause uneven power dissipation in networks-on-chip (NoCs) and make hotspots in some regions of the chip. It is, therefore, essential for a fault-tolerant routing algorithm to be equipped with a comprehensive traffic analysis. Unfortunately, it is a time consuming process and requires advanced tools to examine the traffic around fault regions. Therefore, many of researchers did not perform a detailed analysis of traffic around fault regions [1, 3-6, 8-10]. In this section, with an attempt to gain a deep understanding of traffic analysis issue on faulty networks, we investigate the traffic load distribution around fault-rings.

In Figure 6, distribution of the traffic load on the nodes placed on the f-rings is evaluated comparing to other nodes in the network. Three fault regions overlapping in a row are considered as a block fault region with height 3 and width 2, and two block fault regions with height and width 1. As depicted in this figure, performance degradation in the routing algorithms is mainly related to some bottlenecks in small areas of the network especially at the corners of fault rings. This can be explained as follows: The bottleneck in a corner of fault region could propagate traffic to neighbor nodes and increase the total network latency. The reason behind this behavior has a root in the nature of wormhole switching, which has a great potential to propagate effects of regional bottlenecks into the entire network. In the fault-free case, the traffic

load distribution for the nodes on the f-rings is equivalent with that of other nodes in the network. For this case, Pbc, Duato-Pbc, and Boura's fault-tolerant routing are very attractive, with peak traffic load 15.7%, 16.8%, and 17%, respectively. In the face of failures, however, in the algorithms which do not have specific supervision in using virtual, there is heavy competition between channels around each f-ring, making f-rings act like a hotspot. For example, the performance of PHop routing with peak traffic 61.85% is worse than other adaptive routing algorithms. This is due to the fact that, the faulty nodes do not have any effect in propagating the traffic to other nodes and they only increase the latency of messages in the network. In PHop and NHop algorithms using virtual channels with a specific rule, messages routed via these routings are unable to avoid f-rings when crossing along their shortest paths. This can lead to reduce the throughput and increasing the latency even for the normal messages waiting for the channels reserved by misrouted messages prior to misrouting. It can be explained as follows: In PHop and NHop routings, according to the state and location of a message along its journey path, especial virtual channels are used. Furthermore, in the Boppana-Chalasani's scheme 4 additional virtual channels are needed when two f-rings are overlapped. Hence, PHop has used 24 virtual channels in a 10x10 mesh with overlapping f-rings.

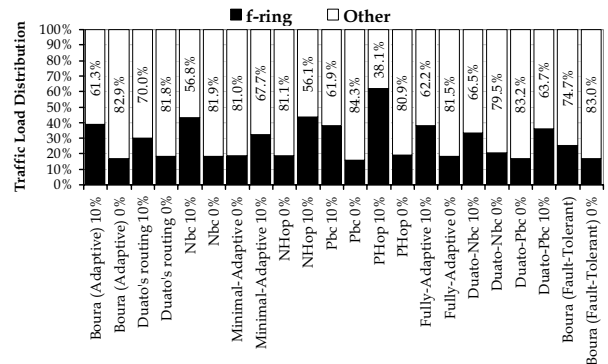


Figure 6: Traffic load distribution for routing algorithms around fault-rings in a 10x10 mesh using 100-flit message length, 24 virtual channels per physical channel, and various fault cases 0%, and 10%.

The Boppana-Chalasani's algorithm usually uses the first 4 virtual channels, while the PHop uses virtual channels according to the number of hops taken to reach the destination. Since the two algorithms cause congestion in using the virtual channels (in PHop the virtual channels of class 0 and in Boppana-Chalasani's routing the first 4 virtual channels), they will cause an overall congestion increasing traffic load on all the nodes in the network. Thus, the effects of hotspots

observed for PHop algorithm are more severe compared with the other routing algorithms. In summary, algorithms such as PHop and NHop do not have further flexibility to choose virtual channels. Moreover, because of the using type of virtual channels in Phop and Nhop with Boppana-Chalasani's scheme, the amount of traffic load on all the nodes would increase, resulting in more latency in the network. The more flexible routing algorithms in choosing virtual channels, the less latency when incorporated with the Boppana-Chalasani's scheme. This conclusion can be easily seen in comparison with Pbc, PHop, Nbc and NHop whose only difference is the flexibility to choose virtual channels. We, as a result, deduce that by more flexibility in choosing the virtual channels, the traffic load would decrease on those nodes not placed on the f-rings.

6. Conclusions

We have considered ten adaptive algorithms divided into two classes, named basic and modified. PHop, NHop, Minimal-Adaptive, Fully-Adaptive, Boura's algorithm, and Duato's methodology are six routings as the basic algorithms and Nbc, Pbc, Duato-Pbc and Duato-Nbc are modified ones. The basic algorithms PHop and NHop were improved with bonus cards (i.e., Pbc and Nbc). These four algorithms are designed based on Duato's routing. Among these routing algorithms, NHop and Nbc use the fewest number of virtual channels, nearly half of the network diameter. PHop and Pbc use the most virtual channels that equal the diameter of the network. Furthermore, modified algorithms with bonus cards try to balance the traffic load on virtual channels, a feature not given much attention previously. In order to incorporate the Boppana-Chalasani's scheme with another adaptive routing algorithm, those algorithms with high flexibility in choosing the virtual channels act better. Among them, the modified algorithms such as Duato-Pbc and Duato-Nbc that are a combination of the basic and modified cases, show quite better performance. In fact, an algorithm like Boppana-Chalasani follows a specific rule in choosing virtual channels. Now virtual channels are busy in the congested case of network; therefore, when incorporated in an algorithm using virtual channels in a random way without any regularity, it causes the virtual channels of higher classes to be quickly occupied. As a result, the Boppana-Chalasani's scheme does not succeed in acquiring virtual channels, the network rapidly gets congested, and the message latency increases. But algorithms such as Duato-Nbc or Duato-Pbc that use the virtual channels with a precise

regularity cause the congestion to be balanced through the network. In the first category, many messages are blocked at a given area in the network when all the virtual channels to be visited are busy. This may lead to rapid congestion in network. However, in the second category that algorithms with high regularity have been employed, each message waits to access the desired virtual channel depending on its location along its journey path. Consequently, the congestion will disperse more through the network helping in decrease of message latency. Future work includes driving an analytical modeling approach to investigate the performance behavior of these routing algorithms.

References

- [1] R. V. Boppana, S. Chalasani, Fault-Tolerant Wormhole Routing Algorithms for Mesh Networks, *IEEE Trans. Computers*, Vol. 44, No. 7, pp. 848-864, 1995.
- [2] J. Duato, S. Yalamanchili, L.M. Ni, *Interconnection networks: An engineering approach*, Morgan Kaufmann Publishers, 2003.
- [3] I. Theiss, *Modularity, Routing and Fault Tolerance in Interconnection Networks*, PhD thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, Feb. 2004.
- [4] M. E. Gómez, et al., A Routing Methodology for Achieving Fault Tolerance in Direct Networks, *IEEE Trans. on Comput.*, Vol. 55, No. 4, pp. 400-415, April 2006.
- [5] A. Mejia, et al., Segment-Based Routing: An Efficient Fault-Tolerant Routing Algorithm for Meshes and Tori, *IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, April 2006.
- [6] Y.J. Suh, et al., Software-based rerouting for fault-tolerant pipelined communication, *IEEE Trans. On Parallel and Distributed Systems*, Vol. 11, No. 3, March 2000.
- [7] Y. M. Boura, C. R. Das, Fault-tolerant routing in mesh networks, *International Conference on Parallel Processing*, 1995, pp. I-106-I-109.
- [8] J. Wu, Z. Jiang, On Constructing the Minimum Orthogonal Convex Polygon in 2-D Faulty Meshes, *IPDPS 2004*.
- [9] R.V. Boppana and S. Chalasani, A Framework for Designing Deadlock-Free Wormhole Routing Algorithms, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 2, pp. 169-183, Feb. 1996.
- [10] J. Duato, A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks, *IEEE Trans. Parallel Distrib. Syst.*, Vol. 4, No. 12, pp. 1320-1331, 1993.