A study of Fault-Tolerant Routing Protocols for Wormhole Networks

Vaishali Kulkarni <vaishali.kulkarni@sun.com>
Gaurav Garg <gaurav.garg@stanford.edu>

Introduction

Large scale parallel computers are potential candidates for providing very large computational power. Due to critical nature of the applications for which such systems are deployed, reliability & fault tolerance is a key metric on which they are evaluated. As the number elements in a large scale multicomputer increase, so does the probability of one or more elements failing. As a result, system reliability & fault tolerance has become a key issue in design of such systems. Multicomputers rely on interconnection networks to implement message passing mechanism between the processing nodes. Therefore, reliability of the interconnection network is very important to the reliability of the whole system. Interconnection networks are characterized by their topology, switching, routing & flow control mechanism. Wormhole switching/routing has emerged as the most widely used switching/routing technique in massively parallel computers due to it's performance & implementation advantage over other techniques such as circuit switching, store & forward(SAF) or virtual cut-through. In this report we investigate various techniques which have been reported in literature for enhancing the reliability & fault-tolerance properties of a wormhole routed interconnection network.

Background

Current multicomputers & multiprocessors use wormhole switching[6] in which messages are serialized into a series of data units or flits which follow the header flit through the network in a pipelined fashion. The channels/buffers that the header flit passes through are reserved for the trailing payload flits until freed by the tail flit. If the header flit gets blocked at some node due to non-availability of network resources (channels/buffers) then all the trailing flits are also blocked in the previous buffers. The pipelined nature of the data transmission in a wormhole switched network leads to lower latency & lesser hardware requirements at each node. It also makes the network more susceptible to deadlock & network failure since the message flits are distributed across several channels & in the event of header flit getting blocked will block a larger number of other messages traversing the network. This problem is exacerbated in the presence of faults.

Several techniques for enhancing the fault tolerant properties of a wormhole switched network have been proposed in the past few years. Most of these techniques focus on supporting static faults which are present when the network is powered on as opposed the dynamic faults which can happen any time during the operation of the network. Several techniques such

'as hardware support for removal of interrupted message from the network & notification of the source of transmission failure have been proposed to deal with dynamic faults. The focus of this report is techniques used to ensure network operation in presence of static faults.

To be fault tolerant a mechanism must be able to establish a path between any given pair of node & guarantee deadlock freedom in presence of faults. These techniques differ in various aspects & can be classified among the following semi-independent axes:

- network topology they are suited for & have been evaluated for (hypercubes vs. low dimensional meshes/tori vs. multistage networks)
- routing algorithm & dead-lock avoidance mechanism used (dimension order vs. turn model vs. planar adaptive vs. scouting vs. dimension reversal etc.)
- degree of adaptivity provided
- maximum number of faults that can be tolerated while providing fault tolerance
- whether certain non-faulty nodes are marked as faulty to form regions of certain topological properties
- amount of hardware support required eg. internode communication, number of virtual channels etc
- whether only minimal paths are allowed or mis-routing/back-tracking is allowed
- performance penalty for providing fault tolerance in terms of latency/throughput
- whether the performance of the network degrades gracefully/non-gracefully in presence of faults.

The following sections begins with a theoretical framework developed by Duato which gives numerical bounds of maximum number of simultaneous faults a given network with certain topology & routing function can tolerate. This section also describes a methodology for designing fault tolerant routing algorithms proposed in the same paper. It is followed by origin based origin based fault tolerant routing scheme. In the subsequent section, a routing protocol based on formation of fault-ring and fault-chain is described followed by a class of algorithm based on misrouting & back-tracking are discussed.

.

I. Theory of Fault tolerant routing for worm-hole networks

This section contains a review of following paper by Duato:

Jose Duato, “A Theory of Fault-Tolerant Routing in Wormhole Networks”, *IEEE Transaction on Parallel & Distributed Systems,* vol.8, no.8, pp.790-802, August 1997

1. Overview

A theoretical formulation of the problem which quantifies the fault tolerant properties of an interconnection network has been reported in [] by Duato. It defines the degree of fault tolerance as the maximum number of faults that a network can sustain while still being able to remain fully connected & deadlock free and derives necessary & sufficient conditions for a network to be able to sustain a given number of faults. It is based on the theory of deadlock free adaptive routing for wormhole routed networks developed by the same author.

2. Definitions & Theorems

Following are the key definitions & theorems described in the paper.

DEFINITION 1: *A routing function R for a given interconnection network I is connected iff, for any pair of nodes* $x, y \in N$, *it is possible to establish a path* $P(x, y) \in P(C)$ *between them using channels belonging to the sets supplied by R.*

DEFINITION 2: *A routing function R for a given interconnection network I is coherent iff for every path P that can be established by R, every prefix of P is also a path of R*

DEFINITION 3: *Given an interconnection network I and a routing function R, a channel* $c_i \in C$ *is redundant iff after removing that channel, the resulting routing subfunction R* is connected & deadlock free.*

DEFINITION 4: *A given routing function R for an interconnection network I has a redundancy level equal to r iff* $\forall c_1, c_2 \ldots c_r \in C, C^* = C - \{c_1, c_2, \ldots, c_r\}$ *defines a routing subfunction* R^* *that is connected & deadlock free and* $\exists c_1, c_2, \ldots, c_{r+1} \in C, C^\dagger = C - \{c_1, c_2, \ldots, c_{r+1}\}$ *defines a routing subfunction* $R^\dagger$ *that is not connected or is not deadlock-free.*

DEFINITION 5: *A given routing function R for an interconnection network I is minimally connected iff it is connected and after removing any channel supplied by R for some destinations, R is no longer connected. Definitions for connectivity, coherence & at least one minimally connected routing subfunction.*

Based on the above definitions, the necessary & sufficient conditions for the network to be fault tolerant are formulated as follows:

THEOREM 1:*A coherent routing function R for an interconnection network has a redundancy level equal to r iff* $\exists k > r$ *minimally connected routing subfunctions* $R_1, R_2, \ldots, R_k$ *and these are the only such routing subfunctions that have no cycles in their corresponding extended channel dependency graphs, and there exists a subset of r+1 channels* $C_m \subset C$ *such that it is the smallest subset of C satisfying that* $\forall i \in \{1, 2, \ldots, k\}, C_m \cap C \neq \varnothing$ *where* $C_i = \bigcup_{\forall x, y \in N} R_i(x, y)$ *is a set of channels supplied by the routing subfunction* R_i.

PROOF SKETCH: (if part) Suppose there exist only k minimally connected routing subfunctions, $R_1, R_2, \ldots, R_k$ and that C_i is the set of channels supplied by R_i. Further let’s suppose C_m is the smallest subset of C containing at least one channel c_i from each C_i & that C_m contains r+1 elements. Removing each element of C_m from C will prevent us from using any of R_i as all of them were minimally connected. => R is not longer connected & deadlock free as no connected, deadlock-free routing subfunction exists. Thus, the network can’t tolerate r+1 faults. If r channels, $c \in C_m$ are removed then there would be atleast one C_i with all it’s elements unremoved from C. Thus, R is connected & deadlock free.

(only if part) Let R be a coherent routing function with a redundancy level equal r. Since, R is connected & deadlock free there exists at least one routing subfunction that is connected & deadlock free. Let $R_1, R_2, \ldots, R_k$ be all minimally connected routing subfunctions & C_m be the smallest set containing atleast one channel supplied by all the routing subfunctions. Obviously, $|C_m| \leq k$ and $k > r$. If $|C_m| \leq r$, then after removing all the elements of C_m from C, R would no longer be connected & deadlock free. If $|C_m| > r + 1$, then removing any set of r+1 channels would still leave one R_i which has all it’s channels unremoved, thus R would still be connected & deadlock-free. Thus, $|C_m| = r$ to satisfy the assumption that R has a redundancy level equal to r.

The above mentioned theory has been developed for physical channels. This theory is applicable if virtual channels are used & a failure can affect a subset of virtual channels belonging to a physical channel. However, faults on physical channels are more common in which case all the virtual channels are considered faulty & the theory can be modified to define the redundancy in terms of physical channels for a network using virtual channels.

3. Design Methodology

Finally, a methodology to for the design of fault tolerant routing algorithms with redundancy level r has been proposed. It begins by a non-minimal fully-adaptive connected & dead-lock free routing function R. It proceeds by identifying for each channel at each node or the network, at least r channels supplied by R such that when all of them are removed, the resulting routing functions is still connected & deadlock free (worst case for connectivity usually occurs when all the faulty nodes are incident on the same node). If adequate number of redundant channels are not found, R is extended by adding physical/virtual channels in such a way that no cyclic dependencies are introduced. The symmetry inherent to a topology need to be exploited in applying these steps to all the nodes in the network. This methodology is illustrated by a design example using routing algorithm based on turn model for a 2D mesh.

4. Criticism

Various fault tolerance techniques have been reported in literature for wormhole routed networks. However, due to different methodology adopted for measuring & reporting the degree of fault tolerance & performance, it is not straightforward to compare the algorithms against each other. This paper provides a theoretical framework which can be used to compare different algorithms. Further it can be used to prove or disprove fault-tolerant properties of new algorithms. Also, it can be easily extended for other classes of switching algorithms. In addition, a heuristic has been proposed which can be useful in design of new or extension of existing routing algorithms to be fault-tolerant.

Although, it is a very elegant theory which gives conditions that a network must satisfy in order to be fault tolerant, the application of this theory is not straightforward. Analyzing the redundancy level of a routing algorithm requires finding all the routing subfunctions which satisfy a particular property. This problem is not trivial & has been conjectured to be NP-complete in the paper. A heuristic has been proposed to design fault tolerant routing algorithms but it's not very simple to implement or accurate. Origin based Fault-tolerant routing algorithm

II. Origin based Fault Tolerant Routing Algorithm

This section contains a review of class of algorithms named origin based fault tolerant algorithms proposed by Libeskind-Hadas & Brandt in the following paper:

R. Libeskind-Hadas & E. Brandt, "Origin based fault tolerant routing in the mesh," *Proc. of the Symp. on High Performance Computer Architecture(Jan.), 102-111*

1. Overview

The key issues this technique tries to address are:

- the degree of adaptivity provided by a fault-tolerant routing algorithm. ie what are the restrictions that a routing algorithm imposes on the selection of output channels.
- marking of non-faulty nodes as faulty to form fault regions of certain topological shape.
- maximum number of faults that can be tolerated

This algorithms tries to maximize the number of routes a packet might take & the number of faults that can be tolerated while minimizing the number of non-faulty nodes marked as faulty. Additionally, this paper introduces a technique which doesn't depend on virtual channels. The key concepts & techniques are presented next followed by a informal proof & criticism of this technique.

2. Description

In this technique, one of the nodes is selected as the origin. All the other nodes are labelled with respect to the origin. Let (x_s, y_s) & (x_t, y_t) be the coordinates of source & destination nodes respectively. The channels of the network are divided among the IN subnetwork & the OUT subnetwork depending on whether the channel is directed towards or away from the origin. Further, following entities are defines for the network:

DEFINITION 1:*The outbox for the destination node t with coordinates (x_t, y_t) is the set of all nodes n with coordinates (x_n, y_n) such that $0 \le x_n \le x_t$ and $0 \le y_n \le y_t$.*

DEFINITION 2:*The diagonal band for the destination node t is the set of all nodes v satisfying the properties:*

- *node v is in the outbox for t; and*
- *if (x_t, y_t) & (x_v, y_v) are the coordinates of t & v respectively, then $x_t - x_v = y_t - y_v + e, e \in \{-1, 0, 1\}$*

DEFINITION 3:*A node v is safe w.r.t. destination t if*

- *node v is in the outbox for t; and*
- *for any pattern of faults in which v and t are nonfaulty, there exists a fault-free path in the OUT subnetwork from v to t.*

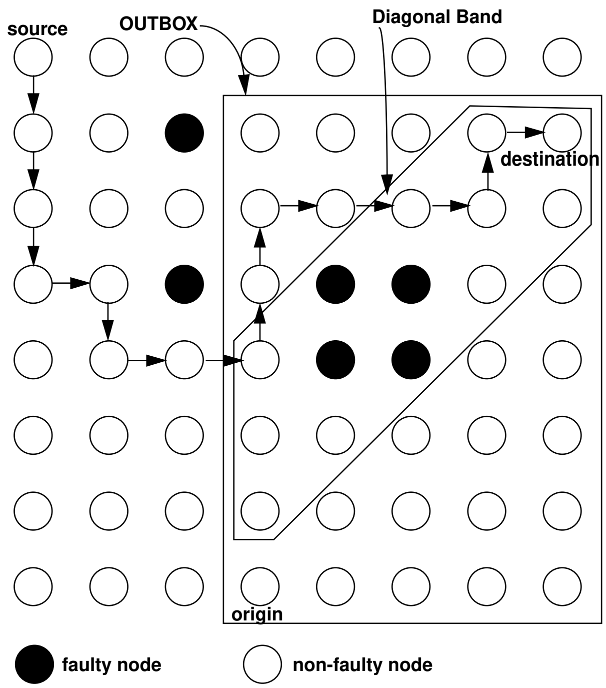


Figure 1: Origin based fault-tolerant routing

The operation of origin based routing proceeds in three phases as shown in Figure 1.

PHASE I: The packet is routed adaptively towards the origin using the IN subnetwork while the header flit doesn't not reach the origin or a node in the OUTBOX. The origin by definition is in the OUTBOX.

PHASE II:When the header flit enters the OUTBOX, the second phase begins. In this phase, at each node, the distance to the nearest safe node is computed & compared with the distance to nearest faulty node. If the distance to the faulty node is greater then the distance to the safe node then the packet is routed to the safe node. Otherwise the routing continues in the IN subnetwork.

PHASE III:Third phase begins when the packet reaches a safe node. Since this is a safe node, there exists a path from the current node to the destination t. Continue routing to destination t.

Operation of the algorithm depends on marking nonfaulty neighbors of faulty nodes to be faulty to form rectangular fault regions. Each node needs to store the distance to it's nearest faulty & safe network in each direction. It is claimed that this technique requires lesser number of non-faulty nodes to be marked faulty & lesser state information to be stored at each node in comparison to other algorithms which require the fault region to be convex. Further the choice of origin is constrained by the condition that the corresponding row & column can't have any faults. This constrains the maximum number of faults that can be tolerated to max(m-1,n-1) for a $m \times n$ mesh. These two conditions guarantee that there is atleast one path from any node in the network to the origin using only the channels in the IN subnetwork. Origin by definition is in the OUTBOX. Thus, the header is guaranteed to reach either the origin or some other node in the OUTBOX.

Further, the nodes in the diagonal band are proved to be safe w.r.t to the destination. Thus, once the header reaches a node in the diagonal band there exists a path to the destination. Origin, by definition, is adjacent to the diagonal band. So, in the worst case the header reaches the origin & is routed to the destination. The constraints that dictate the choice of origin guarantee that there exists a path from the origin to any node in the network.

Since the header is routed towards the origin the Phase I, this algorithm may result in non-minimal paths. If that is not acceptable then the algorithm may be constrained to choose productive channels over non-productive channels during Phase I. A channel is productive if it reduces the distance between the header & the destination node.

The following theorems guarantee deadlock & livelock freedom.

THEOREM 1:*Fault-tolerant origin based routing is livelock free.*

PROOF SKETCH:The algorithm begins in Phase I. By the choice of origin & formation of rectangular fault regions, there exists at least one channel at each node which would reduce the distance to the origin by 1. Thus, the header reaches the origin in a finite number of steps at which time the routing enters Phase II. Now, the header either reaches a safe node or reaches the origin in finite number of steps using the IN subnetwork. If it reaches a safe node then it can be routed to the destination. Otherwise, it reaches the origin which by definition is in the diagonal band hence safe.

THEOREM 2:*Fault-tolerant origin based routing is deadlock free.*

PROOF SKETCH:It is sufficient to prove that any packet follows a path in which the channels can be labeled with strictly increasing or decreasing numbers. Let δ denote the maximum rectilinear distance in the mesh. Further, label each channel (u, v) in the in subnetwork with $\delta + d(u, o)$ and each channel in the out subnetwork with $\delta - d(u, o)$. Notice that a channel can't be in both IN & the OUT subnetwork. Now the operation of the algorithm can be analyzed. While routing in the IN subnetwork, since the distance to the origin is always decreasing, the labels on the channels are strictly decreasing. Once, the packet reaches a safe node or the origin, it is routed in the OUT subnetwork. Since, in this phase, the distance from the node is increasing, the labels are in strictly decreasing order. Thus, algorithm is deadlock free.

3. Results

The results of implementing the proposed algorithm for a 10×10 mesh using productive routing have been reported by the authors. Simulations are performed to measure the following network parameters:

- Number of usable nodes as a function of actual faulty nodes. The authors claim that for same number of network faults the number of non-faulty pseudo-faults is smaller with this class of algorithm as compared to other classes which require convex fault regions or f-cubes/f-rings. Only 12% of non-faulty nodes are marked as faulty in presence of 10% faulty nodes. Further, the performance is shown to degrade gracefully as the number of faults increase.
- Degree of adaptivity in presence of faults. At each step of the algorithm, a header flit can have at most 2 outgoing candidate channels in absence of any faults. The authors report that in presence of faults, an average of 1.34 output channels were available at each router. Also, the output channel vs. number of faults curve is relatively flat.
- Average path length. An average increase in path length of 9-10% over the shortest path is reported in presence of faults.

4. Criticism

The algorithm is shown to require lesser number of non-faulty nodes as faulty as compared to other algorithms based in convex fault regions, f-cubes & f-rings while tolerating upto max(m-1,n-1) faults & providing a high degree of adaptivity. Also, the algorithm degrades gracefully as the number of faults in the network increase. Also, the hardware cost of such a network will be less as compared to networks which require virtual channels.

However, the implementation cost due to the requirement that each node store the distance to nearest safe & faulty neighbor are not considered nor is any mechanism to automatically configure this information described.

It still requires that non-faulty nodes/channels be marked as faulty. Hence, it's might be outperformed by other techniques such as PCS or SR which do not depend on fault regions.

The algorithm can't be extended to other topologies such as tori or hypercubes.

The performance of the algorithm depends on choice of the origin. If minimal routing is used then source/destination pairs with the same row or column must route along that row/column resulting in loss of adaptivity. Number of such pairs achieves a maxima when the origin is at the corner & a minima when the origin is at the center. However, if the origin is at the center then all traffic which crosses quadrants must go through the origin resulting in a hotspot effect at the origin. Heuristics can be used to aid appropriate choice of the origin for different traffic patterns & failure probabilities.

Also, the origin is assumed to be non-faulty. If a dynamic fault occurs on any channel constrained by choice of origin then this algorithm will fail. Again, PCS & SR techniques will outperform this algorithm in this respect.

As an interesting observation, the authors have mentioned that negative first routing proposed by Glass & Ni is a special case of origin based routing when the origin is at the bottom left hand corner of the mesh.

III. Fault tolerant routing using fault-ring and f-chains in mesh and tori networks

This section contains a review of these 2 papers

Fault-tolerant wormhole routing algorithms for mesh networks R. V. Boppana and S. Chalasani IEEE Trans. on Computers, July 1995 and *Fault-tolerant wormhole routing in tori, S. Chalasani and R. Boppana. Fault Proceedings of 8th ACM Conference on Supercomputing, July, 1994*

1 Description of the Problem:

In these two papers the problem of incorporating fault tolerance in non adaptive (e-cube) as well as fully adaptive wormhole routing is addressed. The first paper demonstrates the fault tolerance scheme used for the e-cube algorithm and then it is extended to the fully adaptive routing algorithm. With e-cube routing even a single fault in the network blocks the communication between several other pairs.

Apart from establishing a route between pair of nodes in the presence of fault, the fault tolerant algorithms should exhibit additional features over their fault free counterparts. The issues for fault free routing includes low latency, high throughput, avoidance of deadlock,livelock and starvation and ability to work well under various traffic conditions. The fault tolerant routing algorithm's additional requirements are graceful performance degradation in the presence of fault, ability to handle faults with small increase in routing complexity and ability to work only with local fault knowledge- fault status of the neighboring nodes. For each fault region there exist one or more paths that pass through fault free nodes and links and encircles the fault which are called fault rings. This paper shows that fault rings can be used to route message around the fault regions using only local knowledge of faults.

However, the constraint is that the faulty nodes or links are confined to rectangular blocks.

2 Proposed solution to the problem:

2.1 concepts of fault rings and fault chains

For each fault region in the network with faults, it is feasible to connect the fault free components around the region to form a ring. This is the fault ring f-ring, for that region and consists of fault free nodes and links that are adjacent to one or more components of the fault-region. When the f-region touches one or more boundaries of the network, a fault chain, f-chain is formed. When a fault occurs, the corresponding f-ring or f-chain can be formed in a distributed manner using two step process.

1. Each node that detected a faulty link sends this message to its neighbors in other dimension.

2. Based on the messages received, each node determines its position in the f-ring There are totally eight possible positions to be in f-ring: NW corner, N, NE corner, E, SE corner, S, SW corner and W.

2.2 The fault tolerant e-cube (f-cube) algorithm

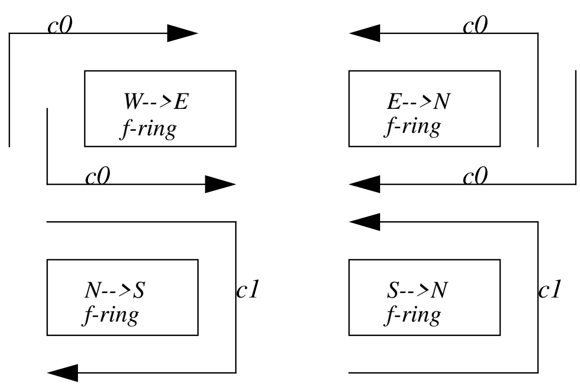
The fault tolerant variant of the e-cube called f-cube uses 2 virtual channels is described first to handle nonoverlapping fault ring in 2D meshes. The f-cube algorithm begins with sending the message to the next node using the normal e-cube algorithm that is takes row hops until it is in the same column as the destination. The message has a status parameter which can be normal or misrouted.

(1) start sending the message with a status of normal to the next hop using e-cube

(2) If the messages' status changes from normal to misrouted (because of fault), it's direction on the fault ring is set as follows.

- If the misrouted message is NS, set the direction to clockwise regardless of is current node and is routed on the f-ring in clockwise direction only.
- For message SN, the direction chosen is counter clockwise. For EW message, direction is set to counter clockwise if the destination is in a row above the current node or clockwise if the current node is in a row below the destination row. If the current node and destination node are in the same row then the direction is set randomly.
- .Similar rule is applied to WE message.

The diagram below shows the virtual channel c0,c1 assignments for each type of misrouted message.



2.3 proof of deadlock free routing of f-cube

Row messages may turn into column messages after few hops but column messages never turn into row messages. Since row messages use only virtual channel c0 and column messages use only virtual channel c1, there cannot be deadlock cycle involving both row and column message. Even when misrouting, c0 channels are used by WE messages on west->east link and on west column of f-rings. The EW messages use c0 channels only on east->west links and on east columns of f-rings hence the set of channels used by WE and EW are disjoint hence there can not be a deadlock among row messages. Also the NS and SN uses two disjoint sets of channels since they are routed in opposite directions on f-rings.

2.4 f-cube applied to f-chains

Since faults are known only locally, f-chains are treated as f-rings by all but the end nodes. Therefore, f-cube may route a message on a f-chain in a direction that actually leads to dead end. The following modification to f-cube solves this problem: If misrouted message M reaches an end node of an f-chain and if its e-cube hop takes M to its previous node or is on a faulty link, then the direction of M is reversed (from clockwise to counter clockwise and vice-versa) and then routed as f-cube routing rules described above. The u-turn on f-chains causes additional channel dependencies. For row and column misrouted messages on f-chains corresponding to faulty blocks touching leftmost column of mesh, the sets of virtual channels used are disjoint. But for misrouted column messages with end nodes on rightmost columns of the mesh, the virtual channel used but SN and NS messages are not disjoint causing deadlock. To avoid this problem, three virtual channels are needed.

2.5 f-cube applied to overlapping f-rings and f-chains

When two f-rings overlap along a column (row, respectively), some nodes in that column (row) belong to west (north) boundary of one f-ring and to east (south) of another f-ring. That means when f-rings overlap, the sets of physical channels used by WE and EW messages are no longer disjoint. The f-cube algorithm can be used in such cases by providing additional channels for each four of the message types. For 2D mesh, four virtual channels are needed

.

2.6 extension to multidimension meshes

The results developed for 2D mesh can be extended to n-dimensional meshes using planar-adaptive techniques. An nD box has base node x = (xn-1,...,xo) and apex node y = (yn-1,...,yo) and set of nodes t = (tn-1,...,to) such that $x_i \le t_i \le y_i$ for $0 \le i \le n$. If a fault set is contained in an nD box such that the interior of the box has only the faulty components and none on its exterior, then the fault set represents a block-fault. Let Ai,j where $0 \le i \le j < n$ denote the set of all 2D planes formed using dimension i and j. Also Ai,j = Aj,i. For f-cube only planes Ai,j, where $0 \le i < n$ and j=i+1(mod n) are used. A normal message travel in dimension $0 \leq i < n$ as per the e-cube algorithm. A blocked message uses the f-ring in its current 2D plane to get around faults. It still requires two virtual channels for non overlapping f-rings. In general the plane Ai,j where i is even uses virtual channels c0 and if i is odd, it uses virtual channel c1.

2.7 fault algorithm for fully adaptive tori

The f-cube algorithm described above can be extended to fully adaptive routing. For this case the routing algorithm can adaptively select the routes to avoid faults until it gets blocked finally in the last dimension because of a fault region. There is no loss of adaptivity because the message has only one dimension that remains to be traversed. This adds four virtual channel to support oblivious routing around the fault region.

The routing algorithm discussed in 2D meshes can also be extended to 2D torus network. However because of the wraparound links in tori, it creates cyclic dependencies causing deadlock. This can be eliminated by using four virtual channels and using routing restrictions. The messages are further classified by weather or not they will traverse the wraparound channel. There are eight types of message types classified based on relative position of the node where it is blocked with respect to the destination node and whether it will use the wraparound link in tori. The table below shows the message types and virtual channel assignment for each message type.

0+,0-,1+ and 1- are message types based on relative position of current node x= (x1,x0) with respect to destination node d = (d1,d0) and M indicates the message will not use wraparound links while W indicates message will use the wraparound link.

Table 1:

Message type	virtual channel	f-ring orientation
0+M	c0	clockwise
0+w	c0	counter-clockwise
0-M	c1	clockwise
0-W	c1	counter-clockwise
1+M	c2	clockwise
1+W	c2	counter-clockwise
1-M	c3	clockwise
1+W	c3	counter-clockwise

fault tolerant fully adaptive algorithm can be described as

1. If M still has a status of normal, continue routing using the original adaptive routing algorithm.

2. If M is misrouted because of fault, based on the message type route it using virtual channels shown the Table 1 above.

case 1. If the current node and the destination node match in dimension and if a hop which takes the message closer to its destination is available, that hope is taken.

case 2: If M is on a f-ring, it is routed using the virtual channel orientation as shown in Table 1 until it reaches the other parallel side of the f-ring such that the current node and destination are in same dimension.

This algorithm is deadlock free. However when the f-rings overlap, additional 8 virtual channels are needed.

3. Results

The simulation was carried out using 16x16 mesh and 16x16 torus networks using 1%,5% and 10% of total network links faulty. This is achieved by introducing a combination of node and link faults which were selected randomly. The results for f-cube showed that there is sharp drop in performance when small number of faults were introduced but then f-cube shows a graceful degradation of performance with further increase in faults.Similarly the fully adaptive case for 16x16 torus showed similar results in terms of graceful degradation with increase in number of faults. It shows about 20% drop in throughput from the fault free case with 10% faults.

4. Criticism

Although its quite important to handle faults in interconnection networks, it adds lot of burden to the routing algorithm and router architecture. A good measure of the fault tolerant routing is how well it performs in fault free network. It should be comparable in terms of performance with its fault-free counterpart when there are no faulty components present. The second measure is how much complexity it adds to router. And lastly how much traffic it generates to inform other nodes whenever fault occurs. The approach based on the f-rings and f-chains has following strong points.

- complexity which is added to the switch is not that much. Besides it involves adding just extra status bit to the message.
- deals with the faults on the boundary without labelling extra fault-free nodes as faulty.
- deadlock/livelock free for both e-cube and other fully adaptive routing
- it can be easily extended to multidimensional topologies.
- it requires only local knowledge of faults. maintaining global knowledge of fault across the entire network adds

complexity and increases traffic since it requires updating all nodes with faulty conditions.

- It can tolerate any number and combination of rectangular faulty blocks.
- It has been extended to tori network and to fully adaptive routing with addition of virtual channels.

below are the weak points

- It can handle only fault blocks which are rectangular in shape. The fact that it can't handle any arbitrary shape fault blocks is serious limitation. In general any arbitrary shaped fault blocks can be converted to rectangular block by labelling other non-faulty components as faulty. But this will cause a significant loss in computational power.
- It has no mechanism to take into account recovered faults at least on a regular basis. Once a component fails, it is considered faulty all the time even when the component recovers from fault.
- Although this algorithm has been extended to tori, fully adaptive and to overlapping f-rings, it requires additional virtual channels. In general if all these cases needed to be handled, then the number of virtual channels needed will be the rather large.

IV. fault tolerant misrouting backtracking with m misroute (MB-m) protocol

In this section the paper reviewed is

A family of fault-tolerant routing protocol for direct multiprocessor network P. T. Gaughan and S. Yalamanchili.IEEE Trans. on Parallel and Distributed Systems, 6(5):482--497, May 1995.

1. Description of the Problem

As the size of the multicomputer network increases, the probability of component failure becomes significant. However it is desirable that the network be functional in the presence of faults with graceful performance degradation. While the techniques based on Wormhole (WR) routing offer reduced latencies, they are not deadlock free in the presence of faults. These techniques are also constrained by pipelined flow of data immediately following the header. Hence if the header is blocked at a faulty component, the data flits will tie up the network resources until it is successfully routed to the destination. This can cause network stalls in the presence of faults. A new model is proposed called pipelined circuit switching which relaxes most of the constraints.

In this paper a PCS based routing protocol that use a depth first approach to routing is proposed. The algorithm realizes fault tolerance by providing sufficient adaptivity to tolerate the maximal number of physical link failures. Thus a good balance is achieved between adaptivity, performance and fault tolerance.

2. Proposed Solution

2.1 Preliminaries

<u>Pipelined Circuit Switching</u>

Unlike in Wormhole routing, data flits don't follow the header in pipelined circuit switching (PCS). So rather than blocking on busy channels, the header may backtrack and release previously reserved channels on the path and attempt an alternative path to the destination. When the header reaches the destination node, an ack flit is sent to the source node and then the data flits can be pipelined over the path just like in WR.

<u>Backtracking Protocol</u>

This protocol acquires a link during its search and later retraces its steps releasing some acquired links to search other regions. These algorithms use history information to ensure that no physical path is searched twice. This defines a search tree containing all candidate paths from the source to destination. The search is realized by depth first backtracking protocol using the following rules.

1. routing header starts at source s with history bits on all outgoing channels cleared. When it leaves a node, a history bit for the channel on which it leaves is set.

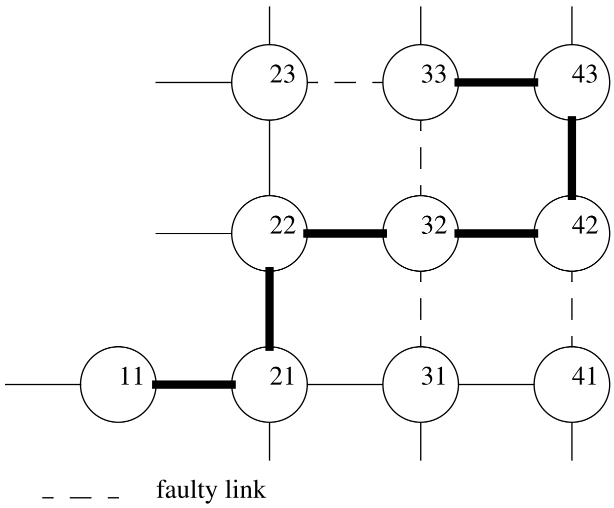
2. When a header enters a node, all descendent history bits are cleared and ancestor history bit is set.

3. routing header may leave on a channel with cleared history bit. If all channels have history bit set, then the header will backtrack to its immediate ancestor node.

4. when a backtracking header enters a node, history information is not changed. Search terminates when the header reaches destination node. If the header backtracks to source node, that means it failed to find the destination.

<u>Misrouting Backtracking (MB-m)</u>

MB-m is a family backtracking protocol with routing restriction than less than or equal to m unprofitable links (which takes away from the destination) may be allowed in a path at any time. So it contains all full or partial paths starting from source node to destination node containing less than or equal to m misroute. An example of MB-1 routing in 5-ary 2-cube is as follows. The source node is 11 and destination 33. The header routes from 11 to 31. At this point it cannot route to 32 so it misroutes to 41. Since MB-1 can only allow 1 misroute, it can only investigate links which take it closer to the destination. Since all those links are faulty, it is necessary to backtrack. When it backtracks to node 31, the options available at 31 are exhausted so it backtracks to 21. From here the header routes to 22 then to 32. Then it misroutes one more time to 42. From there it finds a profitable path to node 33. Note that a misroute that is backtracked over is not counted in the number of misroutes in subsequent routes



2.2 Performance and Fault Tolerant properties of MB-m

The search tree of the MB-m represents a set of paths between the source and destination that MB-m protocol can search. Below are some properties of the MB-m protocols.

1. Non redundancy properties: once a header backtracks across a link, it will not traverse that link again.

2. Exhaustive property: The header will not backtrack to a node n (visit the ancestor node of node n) until all the immediate descendent nodes of n are traversed. Furthermore, all the subtree of the descendant nodes will be traversed before backtracking to n so the search is exhaustive.

3.Finiteness property: By the exhaustive property, every node of the subtree is visited and hence search will terminate in finite time.

4. Path length property: let (s,d) be a pair of nodes in the network. The search tree for this pair of nodes contains path p if and only if p is less than equal to $\|s-d\|1 + 2m$

Fault tolerant properties:

PCS messages are by nature abortive. If a fault stops its progress, it will backtrack and find another path. It will release the network resources it reserved in an attempt to set up the path and hence it doesn't create network stall which is what happens in WR routing. The following property gives the fault handling ability of this protocol.

If 2n is the degree of the node in the network, MB-3 protocol can handle 2n-1 faults. So irrespective of the degree of node (n), MB-3 protocols can tolerate 2n-1 faults.

3 Results

Simulations were carried on 16-ary 2-cube network with and without faults. Without any fault, PCS suffers with large setup time when message size is smaller. Only when the message gets larger, the overhead in setting up the path gets smaller and then WR and PCS performance is comparable. In the presence of faults, a comparative study was performed with other fault tolerant routing protocols like Duato's Protocol, dimension Reversal, negative first and MB-3. Another controllable parameter in this scenario was number of virtual channels per physical channels since performance of some of the protocols which have freedom in misrouting is directly related to number of virtual channels. With virtual channels of 4, most of these protocols handle the network load injected. As the number of virtual channels are increased, Duato's protocol and Negative suffers from not being able to misroute. However dimension routing and MB-m seem to have comparable performance.

4 Criticism

These protocols since they use PCS perform well when the message size is longer because the path setup latency is a smaller fraction of total latency. Besides they exhibit deadlock free property since the data flits are not following the header and hence don't ask for resources while holding other resources. Any time the header is blocked, it can be aborted and the network resources in the entire path are not tied up.

Overall the paper has listed advantages over the conventional wormhole routing protocols (which is not really fault tolerant). The simulations result show better performance in faulty network with respect to other fault tolerant protocols. But these simulations were not performed using varying network load conditions. In MB-m protocol the cost of adding support for the maintaining history and building and maintaining search tree in the hardware is quite high. This hasn't been attributed while comparing this technique with other existing technique. It also increases the size of the header by a factor of logm.

The number of faults handled by MB-m protocols depend largely on how many backtracking (m) are allowed. MB-3 protocols can handle faults up to the network node degree. As the dimension n increases more number of faults can be tolerated but in smaller dimension, say 2-cube, it handle only 3 faults. Also these protocol are said to have finite property because of exhaustive search but that means depending on how many searches it made, the latencies can be rather large. For example, when a complete block of nodes are faulty, before backtracking, it will search the entire subtree of all descendants of that node. Also the number of backtracking allowed goes up as m is increased to handle more faults in the network.

Another potential drawback is PCS suffers when there are no faults and the message size is small. The larger path setup time can be considerable factor of the total latency. So in the fault free environment, its may show degraded performance compared to WR protocol when message sizes are small. Lastly it tolerates only static faults and cannot account for newly introduced faults in the network.

Some of these shortcomings can be overcome by a technique called Scouting Routing(SR) proposed by the same authors[7]. Instead of waiting for an acknowledgement from the header flit, the data flits follow the header flit at a distance d which varies from 0 in fault free regions to a programmable value when the header flit is traversing a region with faulty node. This is achieved by the headerflit sending ack/nack flits as it traverses the network & encounters non-faulty/faulty nodes. Thus, in a fault free network the performance of this network approaches that a wormhole routed network. The only penalty is the additional hardware requirement at each node to support dynamically adjustable distance between header & data flits. Also, there is overhead network traffic due to ack/nack flits.

V. Conclusion

We have studied a few techniques which deal with faults in a wormhole routed interconenction network. These techniques take different approaches to solve the problem of fault tolerance & are suitable for different applications & constraints.

The first proposal investigates the fundamental conditions that must be satisfyed by routing alsorithm in order for it to be fault tolerant. A methodlogy is also described for design of fault tolerant routing protocols. This theory & methodology can be applied to any network topology & routing protocol to evaluate & enhance it's fault tolrance.

The second describes a new technique called origin based fault tolerant routing which depends on fault regions being rectangular & restricts the set of channels that a header might use at each stage of the routing. These are applicable to 2D meshes. This technique is similar in it's approach to techniques based on the turn-model & planar-adaptive routing in that all require fault regions of certain shapes & restrict the set channels that might be used at each step. In fact, the negative first algorithm has been shown to be a special case of this technique. This technique handles fault not on the edge of the network quite well. Further, the algorithm might pay penalty of non-minimal routes in absence of faults if non-productive routes are allowed.

The fault tolerant routing technique based on the concept of fault rings works for mesh and tori for both e-cube type routing as well as fully adaptive routing algorithms. It can handle boundary faults and add less complexity to the hardware. But it can deal with block faults which are rectangular in shape.

The last protocol uses a class of misrouting backtracking (MB-m) protocols which are based on PCS techniques. They offer good fault tolerance for less to moderate number of faults but suffers when there are large numbre of faults and can have rather long routing latencies. Finally, an extensionof PCS called SR proposed by the same authors to overcome these shortcomings has been briefly discussed.

VI. References

1. P. T. Gaughan and S. Yalamanchili. A family of fault-tolerant routing protocols for direct multiprocessor networks. *IEEE Trans. on Parallel and Distributed Systems*, 6(5):482--497, May 1995.

2. R. V. Boppana and S. Chalasani. Fault-tolerant wormhole routing algorithms for mesh networks. *IEEE Trans. on Computers*, 44(7):848--864, July 1995.

3. S. Chalasani and R. Boppana. Fault-tolerant wormhole routing in tori. *Proceedings of 8th ACM Conference on Supercomputing*, July, 1994

4.Jose Duato, “A Theory of Fault-Tolerant Routing in Wormhole Networks”, *IEEE Transaction on Parallel & Distributed Systems*, vol.8, no.8, pp.790-802, August 1997

5. R. Libeskind-Hadas & E. Brandt, “Origin based fault tolerant routing in the mesh,” *Proc. of the Symp. on High Performance Computer Architecture*(Jan.), 102-111

6. P. Mohapatra, “Wormhole Routing Techniques for Directly Connected Multicomputer Systems,” *ACM Computing Surveys*, vol.30, no.3, Sept. 1998

7. Binh Vien Dao, Jose Duato & Sudhaker Yalamanchili, “Dynamically Configurable Message Flow Control for Fault-Tolerant Routing,” *IEEE Trans. on Parallel and Distributed Systems*, vol.10, no.1, Jan. 1999