Restoration mechanism for the N2R Topological Routing Algorithm

Gutierrez Lopez, Jose Manuel; Pedersen, Jens Myrup; Cuevas, Ruben; Madsen, Ole Brun

Published in:
Seventh International Conference on Networking, 2008. ICN 2008

DOI (link to publication from Publisher):
10.1109/ICN.2008.43

Publication date:
2008

Document Version
Early version, also known as pre-print

Restoration Mechanism for the N2R Topological Routing Algorithm

Jose M. Gutierrez
Department of Electronic Systems
Aalborg University
jgl@es.aau.dk

Jens M. Pedersen
Department of Electronic Systems
Aalborg University
jens@es.aau.dk

Ruben Cuevas
Departamento de Telemática
Universidad Carlos III de Madrid
rcuevas@it.uc3m.es

Ole B. Madsen
Department of Electronic Systems
Aalborg University
obm@es.aau.dk

Abstract

The topological routing over N2R structures has been studied and implemented using different techniques. An implemented algorithm achieves good performance and a restoration mechanism has been added to the algorithm to obtain higher reliability. This paper introduces the concept of restoration, when failures occurs, being able to reroute the packets to the destination when the main path is not available. The goal is prove that there is an easy and efficient method for restoration in case of a failure of any element.

1 Introduction

Topological routing is an alternative to traditional routing methods, based on tables. It allows for very fast restoration, and is particularly well suited for large-scale communication where table updates can be time consuming and introduces significant overheads. Topological routing is understood as follows:

At a given address scheme, from any node any packet can be routed given only knowledge of the addresses of the current node and the destination node, with no routing tables involved [1].

Several studies has demonstrated the potential of N2R networks for a degree three structure [2] and the feasibility of topological routing on many different regular topologies [1], [3] and [4] (Grid and Honeycomb), including N2R [5], [6] and [7]. The N2R topology and the previous studies are explained and summarized in the following Section 2. The N2R (Network with 2 Rings) structure is a type of generalized Double Ring structure, where inner ring links do not physically interconnect neighbor nodes.

Future networks will demand better characteristics for the different services supported. Those characteristics can be related to reliability and a number of other performance metrics. Wired networks (such as fibre optic networks) have the handicap of the possible physical cable cuts which implies the loss of connectivity between nodes [8]. Therefore, for topological routing being a feasible option for real networks, a mechanism against failures is required to achieve high performance networks capable of supporting future demands. The failure problem has been treated in [7] from the Path Protection perspective.

This paper treats the failure issue on the topological routing environment of the N2R networks from a new point of view, *"Path Restoration"* [9] and [10]. The goal is to analyze the potential, feasibility and performance of restoration on N2R networks. This method has been successfully implemented on other topologies with interesting results (Grid [3], Honeycomb [4]). Restoration basically consist on, in case of failure, the possibility of automatically rerouting the packets surrounding the failure element to reach the destination. Due to the more complex structure of the N2R than the Grid or Honeycomb an algorithm for applying a restoration mechanism is challenging. The solution must be simple, so the delay at each node is minimized due to the routing tasks, and the path obtained should be relatively short for capacity and performance optimization. *A priori*, the main problem to be solved is the avoidance of infinite loops (the packet is routed forming a loop and is not possible to reach the destination).

The previous algorithm *"Balanced Algorithm"* [5] (introduced in Section 2.2) has been modified to apply a restoration mechanism. The implemented modification is simulated in order to obtain values of traditional parameters as delay, diameter and average distance to analyze the potential Structural Quality of Service (SQoS) offered by this kind of topologies, a number of metrics and properties related with the logical structure of the network [9]. The results are compared with other solutions to document the feasibility, advantages, disadvantages and the potential for applying this new method.

The structure of the rest of the document is as follows. Section 2 treats the definitions, the proper notation and previous studies explanations concerning this topological routing issue. Section 3 introduces the modification of the previous algorithm to be able to implement a restoration mechanism. In Section 4 the proposed algorithm is simulated and the results are exposed. Finally, Section 5 presents the conclusions extracted from this paper.

2 Background

The development of the topological routing on N2R structures already has some results to start with. Therefore, this Section exposes the important ideas required to understand the whole concept of this paper. Subsections 2.1 - 2.4 introduce the basic properties of N2R structures, Balanced Algorithm, failure solutions and Restoration mechanisms:

2.1 N2R Structure

The number of nodes in the N2R structure is any positive even integer larger or equal to 6. These rings each contain the same number of nodes (*p*). Links in the outer ring and the links interconnecting the two rings can be described in the same way as the DR structure, but links in the inner ring are interconnecting node I_i and node $I_{(i+q)modp}$, where q is a positive integer. To avoid forming two separated networks in the inner ring, *q* must fulfil *gcd(p, q)* = 1 (Greatest Common Divisor), also *q* is evaluated from 1 to p/2 [2].

2.2 Balanced Algorithm (BA)

The first approach concerning the issue of topological routing was to implement an algorithm which could route a packet with no path information as routing tables or headers containing the complete path from any source to any destination nodes [6]. A second study proposed several algorithms to improve the results in terms of path distances and path completion time. The best solution found was named *"Balanced Algorithm"* which did not obtain the best results in path distances nor path completion time but those values had not a significant difference with the optimal [5]. The trade off between these two parameters was the best among the studied and therefore the one used to continue the work.

As a brief explanation, what the algorithm basically does is to calculate three distances to reach the destination. These distances are based on the following three types of transmissions: Using the outer ring, the inner ring clockwise or the inner ring counterclockwise.

The only required information to be able to find these values is the destination address. The current node address, *p* and *q* is assumed as implicit information at every node.

The shortest of these three possibilities is selected and the packet is forwarded using the link related to that option. At the next node the procedure starts all over again until the destination node is reached. For further information and deep explanations it is recommended to see [5]. The solution proposed in this paper for a reliable topological routing is based on this algorithm.

2.3 Failure Problem

The failure problem concerning N2R topological routing schemes already has two proposed solutions:

-Pre-calculated paths, this solution is brute force based and it requires long computational executions. In order to use the pre-calculated paths, routing tables are required at every node which implies tables look-up delays. The larger the network, the longer is that delay. This solution is not applicable on topological routing.

-Path protection mechanism [7], this solution was implemented for topological routing as the first approach on the failure protection. When a failure occurs, the algorithm is capable of rerouting the packets, from the source node, using an independent path of the original one. For this method it is required to notify to the source node the unavailability of the original path to execute the algorithm for the alternative path. Therefore, there is a transition time (when a failure occurs but the source does not know about it yet) when the network might not be protected.

These two options will be compared with the new proposal of restoration to be able to find the best solution for the failure problem on these environments.

2.4 Path Restoration

Path restoration methods have been implemented on different regular topologies topological routing methods [3], [4], [9] and [10]. This method requires small tables at the nodes to introduce the failure elements. When a failure occurs, the neighbors of the failure element introduce on their tables the failure element. When any packet is tried to be forwarded using that failure element, the path restoration mechanism is able to reroute the packet surrounding the failure and the destination can be reached. The failure tables are very simple and the maximum number of entries is two. The nodes are degree three, hence, if there are three neighbor elements failing at the same time, the node is isolated, and there is no need for three entries in the table.

3 Algorithm

The implemented algorithm, capable of supporting failures by path restoration, is based on the previous *BA* algorithm. The main goal is to implement a simple mechanism to minimize the routing tasks executed by every node to minimize the delay. At the same time, the paths obtained, when failure occurs, should be as short as possible to optimize the capacity of the global network.

The procedure, in principle, is quite simple. The *BA* algorithm calculates the best link to forward the packets (the shortest path). The modification consists, when there is a failure, on calculating the second best link to forward the packet. This calculation does not imply difficult changes on the *BA* and it is based on the same distances described in Subsection 2.2. The algorithm can be implemented in two ways:

Separated Algorithms at the failure neighbor, *Option 1*: The original path algorithm is executed always, even when there is an entry on the table. If the failure element in the table is the same as the one calculated for the original path to forward the packet, the algorithm is executed again calculating the second best link.

Unique algorithm at the failure neighbor, *Option 2*: The restoration algorithm is executed at any node with some entry on its table. Hence, the two best links are always calculated at this type of nodes, even though there will be occasions when it is not necessary. If the failure element is not the best link to forward the packet, the best path is not blocked, and the calculation of the second link is not necessary.

The two options will be analyzed and discussed in Section 4. The result of the two options will be the same path, but one will be faster than the other, reducing the delay. At the neighbor nodes of the failure it is possible to reroute the packet using the incoming link since it might be the best option. After this point, the packet is not allowed to travel backwards. At these nodes, also the same issue about how to apply the restoration method as at the neighbors of the failure can be discussed. For the moment, without considering which of the two options is used, the next example, illustrated in Fig. 1, explains the procedure. The protection path obtained with the Protection mechanism [7] is presented as well to discuss some of the differences between the two techniques.

The restoration path is exactly the same as the original one until a failure blocks the shortest path (in fact, the same algorithm is executed). When the original path is blocked, the second best link is used to forward the packet, to able to reach its destination. The protection path is quite different. The protection path is calculated from the source node, and it is significantly shorter due to the knowledge from the source node that the original path is blocked. This protection path calculation has two main differences:

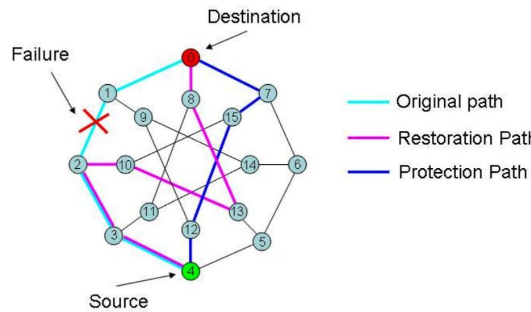

Figure 1. N2R(8,3) example

1) This method is applicable when the source node knows about the failure, in the transition time (since the failure occurs until the source receives the information about it) it is not possible. The failure notification requires some managing tasks besides the routing method itself, but the result gives shorter distances since the path can be modified from the source and not from the failure point.

2) The algorithm is significantly more complex, the execution time is double the original BA. The restoration mechanism is simpler and faster, in Section 4 the execution times values obtained for the simulation are presented.

At the time of simulating the algorithm, some problems were found at specific configurations. The problem is the commented infinite loops, the packet gets stuck on infinite loops and is not able to reach the destination. The problem can be solved by adding some conditions to avoid this loop. The loops are likely to take place when any of Conditions (1) or (2) is fulfilled, being F_e the failure element, N_c the current node and O_r the outer ring. Both conditions include the locations of the failure, therefore, in these cases, information about where the failure is must be included in the packet, it can be just one flag (1 bit).

$$\left\{\frac{p+1}{q}\right\} = 0 \;\&\; F_e \in O_r \;\&\; N_c > p \qquad (1)$$

$$F_e \in O_r \;\&\; N_c > p \;\&\; p > 30 \qquad (2)$$

4 Simulation

The simulation of the algorithm was performed varying the value of p from 5 to 100 (200 nodes in total) and testing at each of these values all the possible configurations (possible q values). It is important to mention that the algorithm works for any of these configurations, but for the graphical representations, only one q is chosen for each p. These q are the optimal (best result on diameter and average distance) for the original path obtained at [5]. The procedure is to simulate failures on the nodes and the algorithm simulates transmissions from all the nodes to all the nodes. The simulation covers the failure of all the nodes in the network, one at a time, to obtain a deterministic result. When a node fails, three of the links in the network are unavailable instead of one when there is a link failure. Therefore, the nodes failures are studied to give the worse case when an element fails, important to define the limitations of the network. In real wired networks, the links are more likely to fail than the nodes, hence, the distance values can be shorter than the presented ones, but never longer.

The graphs in Figs. 2-5 illustrate the comparison between the average distances and diameters of the precalculated paths, protection and restoration option and the execution time[1] at every node and path completion time of the protection and restoration mechanisms. All the values represented correspond to the paths obtained when the shortest one is not available and the Protection mechanism results are extracted from [7]. The execution time and path completion time correspond to the delay caused by the routing tasks, there are other factors that might have influence on the delay. These other factors are not included since they are independent of the routing method, therefore they do not affect the comparisons treated in this paper.

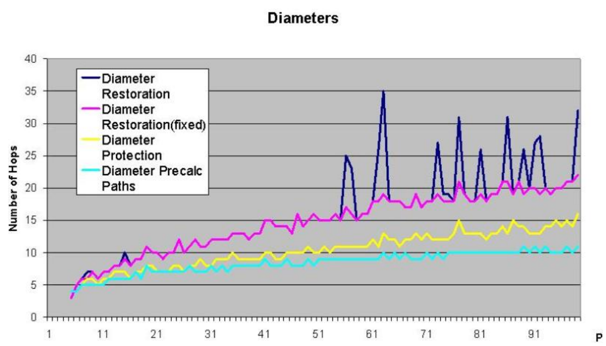

Figure 2. Diameters

The first of the parameters discussed is the diameter of the paths (maximum distance between any pair of nodes). The diameter values are always interesting to define the limitations of the network. Fig. 2 illustrates the comparison between the diameters of the precalculated, protection and restoration paths. As predicted, the diameter values obtained for the proposed restoration mechanism are longer, mainly due to the commented fact that there is no path modification until the failure is reached. Therefore, there is part of that path that could be found as a waste of resources. Below, at the completion time analysis, the benefits of this

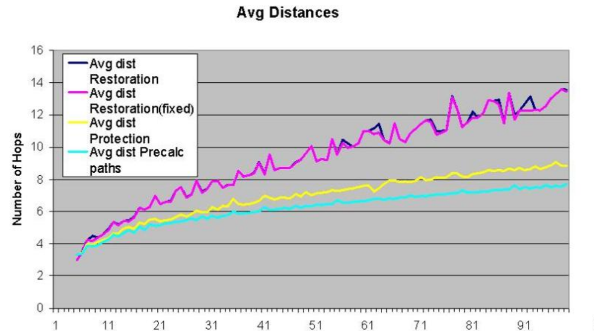

Figure 3. Average path distances

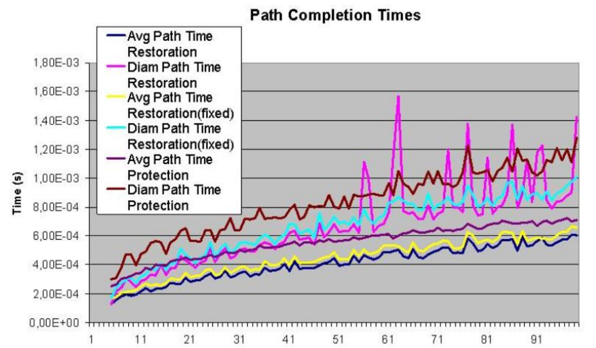

Figure 4. Path completion time

mechanism will be discussed. When $p > 50$ (more than 100 nodes) some peaks can be identified at certain values of p. These out of proportion peaks are mainly due to a problematic link decision at the time of routing the packet. The decision is taken as the second shortest path, the problem is that this option also includes the failure element. This decision is corrected along the path but with the consequence of very long distances. This problem can be predicted using the information about the location of the failure (due to Condition (2) this information is added to packet so it does not represent an addition of information) but the algorithm becomes more complex, this modification will be called *"Restoration (fixed)"*, included in Fig. 2.

Fig. 3 presents the average distance of the different options, as the diameter in Fig. 2, and the results are similar and the same conclusions can be extracted. In this case, the difference between the Restoration and Restoration (fixed) is not very significant. To see the effect of the path distances over the delay, Fig. 4 presents the average path and diameter completion times of the Protection mechanism and the two Restoration mechanisms. The results are very interesting, the Restoration mechanisms are faster than the Protection one. This fast execution allows to have lower delays in transmissions even though the distances are longer. The only exception is the diameter completion time for those

[1]The execution time was obtained under the same conditions as the *BA*, since depending on the machine where these algorithms are executed the result will vary. It is assumed though that the proportion will be maintained. The machine used is a Genuine Intel(R) CPU T2050 @1.60 GHz (2 CPU) and 1GB of RAM. The Software used is PHP and MySQL

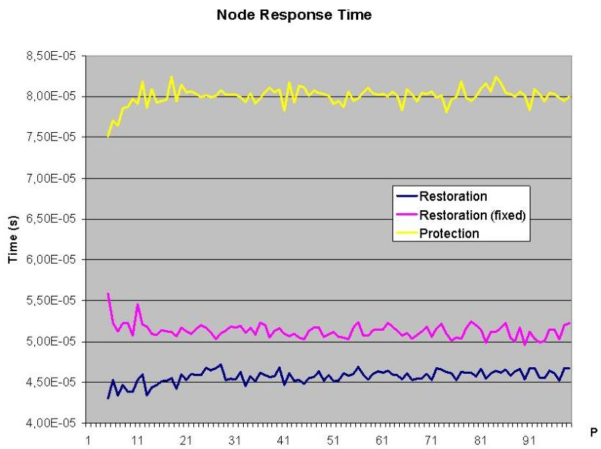

Figure 5. Node response time

problematic values of p, which in that case the Restoration (fixed) can be applied to solve this maximum delay.

Fig. 5 illustrates the average time each node spends on executing the algorithm. The same pattern, as at the previous N2R studies [5] and [7], is followed. The execution time of the algorithms is independent of the number of nodes in the network, while the table look-up time increases with the increment of entries. Based on these results the issue introduced in Section 3, about how to apply the algorithm when the packets reach a failure, can be discussed.

The average execution times considered for this explanation are: Original path algorithm time (BA) [5], $T_{BA} = 0.035(ms)$ and restoration algorithm time $T_R = 0.045(ms)$. The question is which method is better when there is an entry on the table of the current node. Due to the regularity of the topology, the links are equally used for the different transmissions between pairs of nodes. Then, an incoming packet is assumed that 50% will be forwarded using one link and the other 50% using the other, (the third one is the incoming link). If the current node, where the packet arrives, has an entry on its table, there is a 50% chance of having the shortest path blocked. If the algorithm executed is the original path one and if the outgoing link is the same as the failure, then, the Restoration mechanism is executed. The result for the average execution time at a failure neighbor is described by Formula (3).

$$T_{option1} = T_{BA}*0.5+(T_{BA}+T_R)*0.5 = 0.057(ms) \quad (3)$$

Instead, if the restoration mechanism is always applied when there is an entry on the table, the average execution time is T_R, see Formula (4).

$$T_{option2} = T_R = 0.045(ms) \quad (4)$$

Applying the same principle to the Restoration (fixed) mechanism the result is the same ($T_{Rf} = 0.051(ms)$). It is worthy to always apply the Restoration mechanism, *Option 2*, when there is an entry on the table of the current node, obtaining a faster average response to a failure. After this point in the path, it is also worthy to apply *Option 2* for the same reasons.

This simulation leads to the conclusion that there is no general solution for the failure problem, in case of an alternative to the traditional table routing techniques. There are two main factors that affect the decision:

Capacity optimization: The shorter the paths, the more optimized is the capacity of a network. Also, directly related with path length, the less elements involved in transmissions, the lower is the probability of failure of the path. In this case the Protection mechanism is the best option, but always considering that it is required to implement some managing task for the failure notice to the source and there is a transition time when the packets will be lost, and they must be retransmitted when the network is stabilized.

Delay: The faster the transmissions, the better performance of the network achieved. In this case, the Restoration mechanism performs better, and in the case of the problematic p values the fixed algorithm can be used. Unfortunately, there was no mathematical relation found between p and q to identify this problematic values, but if $p > 50$ the fixed algorithm can be applied with no significant difference on the delay on the non problematic ones.

4.1 Two Simultaneous Failures Extension

The previous solutions for the failure issue (precalculated second path or Protection mechanism) cannot guarantee at any situation the availability of a path between any pair of nodes when two failures occur simultaneously. The question is if the Restoration mechanism is capable of supporting two failures at the same time.

This extra feature of the algorithm is out of the scope of this study, but as an introduction, the algorithm was tested for a specific configuration to establish the bases of the problem. The configuration chosen is N2R(25,7), optimal configuration for 50 nodes, and all the possible combinations of two failures were simulated. The result is that the algorithm will reach the destination at all the cases but when the two failures are the two neighbors of the destination and belonging to the same ring . In this case the packet will be stuck on an infinite loop.

The problem has an easy solution with just the addition of two flags (two bits) to the header of the packet. A packet is transmitted and the previous hop to the destination is a failure located at the same ring. Then, the first flag is set to "1", and the packet is normally rerouted. The packet keeps travelling and at the previous hop before reaching the destination there is another failure and also located at the same ring. The other flag is set to "1" and when both flags are equal to “1”, the packet is automatically rerouted to the opposite ring (the one that the destination does not belong to) and it will be transmitted using that ring until the available neighbor is reached.

If the two failures are neighbors of the source there is no problem since the knowledge of the two best links to forward the packet implies the knowledge of the third one.

The first impression is that it is possible to support two failures. Therefore, it might be an interesting topic to study in future. If the number of simultaneous failures is larger than 2, the transmission cannot be guaranteed between any pair of nodes since if the failures are neighbors of the source or destination they are isolated. But in the case that those failures do not isolate any node, it would be interesting for further research the performance and, mainly, the infinite loop avoidance of the algorithm.

5 Conclusion and Discussion

This study concludes the deep work about the failure treatment on N2R networks applying topological routing schemes. There are different solutions which, depending on the requirements of the services and applications, can be better than others. The options are based on two different ideas: **Path Protection** and **Path Restoration**.

The Protection option optimizes, in a more efficient way, the capacity of the network and lowers the probability of failure of the path (protection path) since the logical distances between pairs of nodes are shorter. On the other hand, the path completion time is higher and there is the possibility of not having protection for a short period (transition time). The packets lost during this transition time must be retransmitted when the network is stabilized (the source received information about the failure). This mechanism requires some extra managing tasks to let the source nodes know about problems on the shortest paths.

The Restoration option performs faster achieving lower delays on the transmissions between nodes, but with the consequence of reducing the global capacity of the network. There is no management tasks and the mechanisms can automatically handle failure situations with no control packet traveling around the network. When a failure is blocking the original path, the rerouting is instantaneous, there is no notification of the failure to the source node. This feature implies lower probability of packet loss than the Protection mechanism. The potential of being able to handle two simultaneous failures in the network has been tested, obtaining promising results to keep studying this option. Restoration is the simplest method to deal with failures.

The key property of both options is the execution time of the algorithm, it is constant and independent on the number of nodes. The delay caused by the routing tasks at each node is approximately the same if the number of nodes in the network is 10, 1000 or 10000, which makes this idea very attractive for large scale networks. The table look-up time increases with the number of entries, thus, for a large number of nodes this look-up time can be unacceptably long.

Both approaches have interesting characteristics and benefits, hence, an option could be the combination of both solutions to handle failure problems. A priori, in case of combining the two options, the main challenge would be to define and implement a fast mechanism to make the decision. This decision mechanism should not have significant effect on the delay or else the benefits in terms of fast execution of the algorithm could be affected. Also, the potential criteria for deciding which option should be executed can be defined. These issues might be another interesting topic for further research.

References

[1] Pedersen, Jens Myrup ; Knudsen, Thomas Phillip ; Madsen, Ole Brun. “Topological Routing in Large-Scale Networks”. Proceedings of IEEE/CACT 2004, The 6th International Conference on Advanced Communication Technology. 2004.

[2] T. Jørgensen, L.Pedersen and J.M. Pedersen“Reliability in single, double and N2R ring network structures” The International Conference on Communications in Computing (CIC’05) Las Vegas, Nevada, United States, June 2005 .

[3] J.M. Pedersen, A. Patel, T.P. Knudsen , O.B. Madsen “Applying 4-regular Grid Structures in Large-Scale Access Networks” Computer communications 29 (2006) 1350-1362.

[4] Ivan Stojmenavie, “Honeycomb Networks: Topological Properties and Communication Algorithm”, IEEE Transaction on Parallel and Distributed System, Vol8, NO. IO, October 1997.

[5] Jose M. Gutierrez Lopez, Ruben Cuevas Rumin, Jens M. Pedersen and Ole B. Madsen “Improving Topological Routing in N2R Networks”. CAAN´07, August 2007, Canada.

[6] Jens M. Pedersen, Muhammad Tahir Riaz, Ole B. Madsen “A Simple, Efficient Routing Scheme for N2R Network Structures”, IT&T Annual Conference 2005 - Cork, Ireland .

[7] Jose M. Gutierrez, Jens M. Pedersen , Ruben Cuevas and Ole B. Madsen “Protection mechanism for the N2R Topological Routing Algorithm” Submitted to HPSR 08, China

[8] Wayne D. Grover, “Mesh-Based Survivable Networks, Options and Strategies for Optical, MPLS, SONET and ATM Network” Prentice Hall PTR, 1st edition (August 14, 2003).

[9] Jie Wu, ”A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model”, IEEE Transactions on Computers. V01.52. No.9, September 2003

[10] Ming-Jer Tsai, Sheng.De Wang, “Adaptive and Deadlock-Free Routing for Irregular Faulty Patterns in Mesh Multicomputer”. IEEE Transactions in Parallel and Distributed Systems, Vol. I I, No.1, January 2000.