



UNIVERSITY
of
GLASGOW

Safaei, F. and Fathy, M. and Khonsari, A. and Ould-Khaoua, M. and Shafiei, H. and Khosravipour, S. (2007) On quantifying fault patterns of the mesh interconnect networks. In, *21st Annual Conference on Advanced Networking and Applications, 2007. AINA '07, 21-23 May 2007*, pages pp. 956-961, Niagra Falls, Ontario, Canada.

<http://eprints.gla.ac.uk/3734/>

On Quantifying Fault Patterns of the Mesh Interconnect Networks

F. Safaei^{a,b}, M. Fathy^b, A. Khonsari^{c,a}, M. Ould-Khaoua^d, H. Shafiei^a, S. Khosravipour^a

^a *IPM School of Computer Science, Tehran, Iran*

^b *Dept. of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*

^c *Dept. of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*

^d *Dept. of Computing Science, University of Glasgow, UK*

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

Abstract

One of the key issues in the design of Multiprocessors System-on-Chip (MP-SoCs), multicomputers, and peer-to-peer networks is the development of an efficient communication network to provide high throughput and low latency and its ability to survive beyond the failure of individual components. Generally, the faulty components may be coalesced into fault regions, which are classified into convex and concave shapes. In this paper, we propose a mathematical solution for counting the number of common fault patterns in a 2-D mesh interconnect network including both convex (I-shape, II-shape, □-shape) and concave (L-shape, U-shape, T-shape, +-shape, H-shape) regions. The results presented in this paper which have been validated through simulation experiments can play a key role when studying, particularly, the performance analysis of fault-tolerant routing algorithms and measure of a network fault-tolerance expressed as the probability of a disconnection.

1. Introduction

Advances in VLSI technology will soon allow a single chip to contain more than one billion transistors, indicating that a large number of processing units (such as CPU, DSP, multimedia processor) shall be integrated into one packaged chip. In these systems, communication resources are competed by the vast volume computational resources. Over the recent years, a new trend in the design of communication architecture in multi-core Systems-on-Chip (SoCs) has appeared in the research literature. In particular, researchers in [1] have recommended that a complex SoC can be viewed as a micro-network of multiple blocks due to the fact that this trend converges to the Networks-on-Chip (NoCs) solution. With a communication-centric design style, NoC was proposed to mitigate the complex communication problems.

Interconnect structure of regular high performance systems such as multicomputers is a promising architecture to achieve high performance NoC design. The interconnect networks in such systems have become the center of focus, because the network structures and topologies, as well as the processing elements greatly influence the system *cost* (complexity), *performance* (throughput), and *reliability* (fault-tolerance) [2, 3]. The design of NoCs trades-off several important choices, such as topology, routing strategy, and application mapping to the network nodes. Nowadays there are some experimental NoCs being developed such as SPIN [4], aSOC [5], CLICHÉ [6], and OCTAGON [7].

The NoC system is composed of a large number of interconnected components (such as processors, embedded memories, and intermediate processor blocks) where communication is achieved by sending messages over a scalable interconnect network. While providing adequate Quality-of-Service (QoS) under performance constraints will continue to present a major design concern, achieving on-chip fault-tolerant communication will become increasingly more important. Therefore, fault-tolerance is becoming a key concern in designing parallel computer systems, MP-SoCs, multicomputers, and cluster computers. Fault-tolerant algorithms in such systems aim at providing continuous operations in the presence of one or more faults by allowing the graceful degradation of system.

In recent years, many researchers have addressed to several issues in the field of fault-tolerance and reliability analysis of large scale parallel and distributed systems [4-16]. These researches span a diverse range of systems and applications such as massively parallel processors [8], cluster-based systems [9], mobile systems [10], sensor networks [11], and more recently network on chip [1].

The simplicity and accuracy of analytical model as an alternative to simulation make it an attractive and effective tool for characterizing the performance trade-

off. A model attempts to explain behaviors of a system, using a set of simple and understandable rules. The research presented in this paper use theoretical results of combinatorics to enumerate the number of occurrences of different common fault patterns in a mesh network.

The paper is organized into five main sections. Section 2 reviews some definitions and background of the structure of mesh networks. Moreover, the concept of fault models and fault patterns used in the paper is introduced in this section. In Section 3, the general solution to compute the number of common fault patterns occur in the mesh interconnect topologies has been proposed. Section 4 confirms the proposed mathematical expressions using a simulation approach. Finally, Section 5 concludes our study and presents possible directions for future work.

2. Preliminaries

This section initiated by a discussion on 2-D mesh structure and then describes the necessary background information that is used in the paper.

2.1 The Mesh topology

A 2-D mesh with $N = R \times C$ nodes has an interior node degree of 4 and $E = 2 \times R \times C - R - C$ channels. Each node u has an address (u_x, u_y) , where $u_x \in \{0, 1, 2, \dots, R-1\}$, $u_y \in \{0, 1, 2, \dots, C-1\}$. 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. Thus, nodes at the corners and edges in the network have two and three neighbors, respectively.

2.2 Fault models

The fault-tolerant computing literature is extensive and thorough in the definition of fault models for the treatment of faulty digital systems. Faults in a network can take many forms, such as hardware faults when a node or a link stops to function, software bugs, or malicious sniffing or removal of packets. In this paper, we shall focus on hardware faults only. Hardware faults may be present at network startup time, in which case the fault is considered to be *static* [3, 12, 14]. Routing tables can be calculated based on knowledge of these faults, and loaded into each routing node before the network is made available for traffic. In contrast, a *dynamic fault* occurs while the network is running. To

avoid shutting down and restarting the network manually, there are approaches which allow automatic reconfiguration. These approaches might refuse new packets access the network until reconfiguration is completed. Other approaches such as adaptive routing algorithms can also be used.

2.3 Fault patterns

Adjacent faulty nodes are coalesced into *fault regions*, which may lead to different patterns of failed components. To analyze the performance of fault-tolerant systems, it is important to identify and quantify the fault regions, which may occur in the network. Faulty regions extended by faulty components, may form *convex* (also known as block faults) or *concave* shape [3, 12-16].

Definition [3, 12, 13]: A convex region is defined as a region F in which a line segment connecting any two points in F lies entirely within F . If we change the "line segment" in the standard convex region definition to "horizontal or vertical line segment", the resulted region is called *rectilinear convex segments*. Any region that is not convex is a concave region.

Examples of convex regions are I-shape, \square -shape and concave regions are L-shape, U-shape, T-shape, H-shape, +-shape. Figure 1 illustrates common types of convex and concave fault regions in a mesh network.

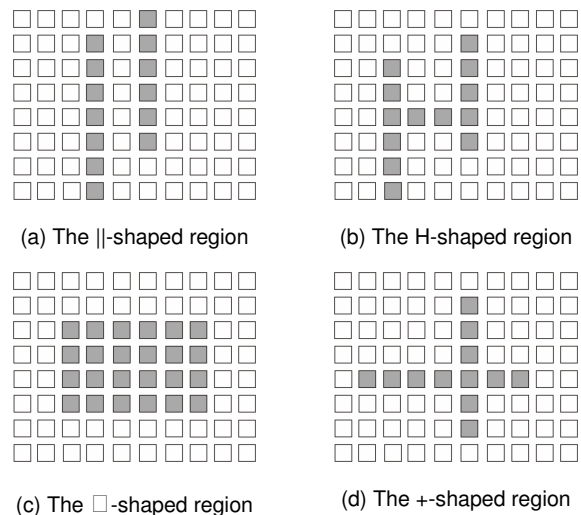


Figure 1: Examples of convex and concave fault patterns in a 2-D mesh interconnect network.

3. The mathematical model

In this section, we propose a general solution to enumerate the number of occurrences of fault patterns

in the mesh topology. We begin by establishing some conventions. Let F_n denote the fault region constructed by n faulty nodes ($n > 2$). Some of the common fault patterns occurring in the network are depicted in Figure 1. Our main purpose is to compute the number of occurrences of F_n in a given network with a total of N nodes. Let $\Phi(F_n)$ denote the total number of occurrences of pattern F_n conditioning on n faulty nodes in a mesh. We also assume that the probabilities of node failure in the network are equally alike and independent of each other. Due to the asymmetric property of the mesh topology, it is necessary to compute the $\Phi(F_n)$ for each shape separately. In what follows, we calculate $\Phi(F_n)$ for five common fault patterns. However, by using the same approach, enumeration of the other fault patterns can be realized for each shape.

Case 1 (the l-shape): The simplest fault pattern is constructed when all n faulty nodes lie in a straight line. It is noteworthy to mention that we have to distinguish between a horizontal and vertical line in a mesh with R rows and C columns. We would like to enumerate all cases that is possible for a line to be embedded in the mesh without being distorted. We assume that $n \leq C$ for the vertical line and $n \leq R$ for the horizontal line. It is clear that a line may rest in C places for vertical case and R places for horizontal line. Moreover, for each placement there are $R-n$ safe positions that a vertical line can be allocated and $C-n$ safe positions for horizontal case. It follows immediately from the above discussion that for the vertical line, $\Phi(F_n)$ can be calculated as

$$\Phi(F_n) = (R - n) \times C \quad (1)$$

Similarly, by exchanging the roles of R and C , we have the following equation for the horizontal line as

$$\Phi(F_n) = (C - n) \times R \quad (2)$$

Case 2 (the L-shape): Another common fault pattern is L-shape. In order to enumerate the number of L-shaped fault patterns with n faulty nodes we employ a form of L which is illustrated in Figure 2. The number of L-shaped patterns in which their related line segments contain i and x faulty nodes can be represented as

$$L(i, x) = (R - i) \times (C - x), \quad (3)$$

where $x = n - i$.

By the definition above and noting that the variable i is in $[1, n - 2]$ interval, the number of L-shapes in a mesh network can be derived as

$$L_{mesh} = \sum_{i=1}^{n-2} L(i, x) \quad (4)$$

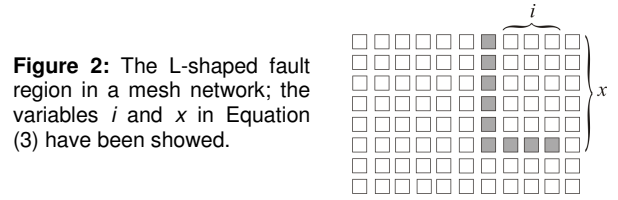


Figure 2: The L-shaped fault region in a mesh network; the variables i and x in Equation (3) have been showed.

On the other hand, depending on the location of each pattern in the network, there are various coefficients of rotation and reflection. Let \aleph_{φ_n} indicate the number of ways that a pattern F_n can be reflected in a mesh and let \aleph_{F_n} denote the number of ways that F_n can be rotated in a given network without affecting its shape and topology. These parameters are heavily dependent on the shape of F_n . It can be easily shown that the values of \aleph_{F_n} and \aleph_{φ_n} for the L-shaped pattern are the same and equal to 4. We can therefore obtain $\Phi(F_n)$ as

$$\Phi(F_n) = 4 \times \sum_{i=1}^{n-2} L(i, x) \quad (5)$$

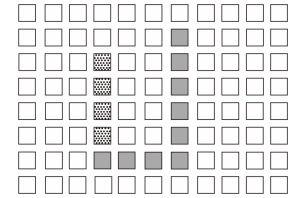


Figure 3: The U-shaped fault region is constructed from an L and a straight line.

Case 3 (the U-shape): Now, let us consider the U-shaped fault pattern. To enumerate all cases in which the U-shaped pattern can be embedded in a mesh, we use the following approach. As it can be seen in Figure 3, the U-shaped pattern consists of a single line which is joined to an L-shaped pattern. Thus, the number of different L-shaped patterns for each connected line yields the following summation

$$\sum_i L(i, x-1) \quad (6)$$

Moreover, the length of each connected line may be variable and any node that is added to one side of the L may decrease one node from the other side. So, for each new case we should enumerate the number of possible L-shaped patterns that is denoted by

$$\sum_{k=2}^{\infty} \sum_{i=0}^{n-2k} L(i-2, x-k-2) \quad (7)$$

where subscripts k and i are relating to add one node in the line segment and the L-shape, respectively. In the case that the vertical line segments of pattern U are equivalent (which is referred as the balanced U), the

values of \mathfrak{K}_{φ_n} and \mathfrak{K}_{F_n} are 1 and 4, respectively. In the unbalanced case, however, \mathfrak{K}_{φ_n} and \mathfrak{K}_{F_n} are identical and equal to 4. Therefore, in Equation (7), we have to distinguish these two cases. That is

$$(\mathfrak{K}_{\varphi_n} + \mathfrak{K}_{F_n}) \left[\sum_{k=2}^{\infty} \sum_{i=0}^{n-2(k-1)} L(i-2, x-k-2) \right] + \mathfrak{K}_{\varphi_n} \sum_{k=2}^{\infty} L(n-2k-3, k-1) \quad (8)$$

By using the above formula, $\Phi(F_n)$ can be described in a general form which is given by

$$\Phi(F_n) = 4 \times \sum_{k=2}^{\infty} \left[2 \times \sum_{i=0}^{n-2(k-1)} L(i-2, x-k-2) + L(n-2k-3, k-1) \right] \quad (9)$$

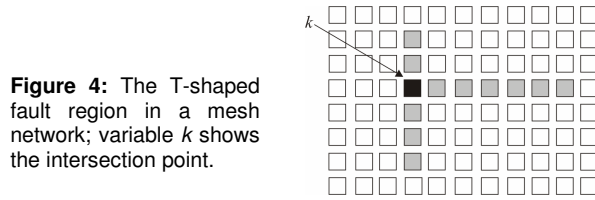


Figure 4: The T-shaped fault region in a mesh network; variable k shows the intersection point.

Case 4 (the T-shape): One of the most common fault patterns is T-shape. For counting the number of pattern T in a mesh, we first represent the following proposition.

Proposition: let n be the number of faulty nodes in a mesh network. Then, we can propose the following expression $T_{mesh}(n) = U_{mesh}(n-1)$, for $n \geq 5$.

Proof: Take U as an arbitrary pattern with n faulty nodes. To compute the number of U-shaped patterns we assume that one of its line segments is being fixed such that we can enumerate the number of existing L-shaped patterns. Therefore, the corresponding line segment may be altered using an appropriate manner. To enumerate the number of T-patterns, let the unchanged line segment be in the opposite direction so that we can compute the number of L-shaped patterns (see Figure 4). However, in this case we need to identify the number of L-patterns in which their intersection point (depicted by k in Figure 4) is being fixed. To this end, it is required to append this point to the other fixed vertices in the pattern. Consequently, the number of T-patterns would be equal to the number of U-patterns with a number of faulty nodes minus one. ♦

By using the above theorem, $\Phi(F_n)$ can be expressed as

$$\Phi(F_n) = 2 \times \sum_{k=2}^{\infty} \left[2 \times \sum_{i=0}^{n-2k-3} L(i-2, x-k-2) + L(n-2k-4, k-1) \right] \quad (10)$$

Case 5 (the H-shape): Finally, to enumerate the number of H-shaped fault patterns we observe that the pattern H may be formed of an U-shape added to two vertical line segments that are joined to two intersection vertices of U (see Figure 3). Therefore, the number of H in a mesh network can be expressed as

$$\sum \sum U_{mesh} \quad (11)$$

The lower and upper limits of summation terms in Equation (11) must be chosen due to the fact that in an H-shaped fault pattern the length of any vertical line segment may not be less than one and at least one fault-free column of nodes must be considered between every two line segments. As a result of the discussion above, we can determine the related mathematical formula as

$$\Phi(F_n) = \sum_{j=1}^{n-7} \sum_{l=1}^{n-6-j} U_{mesh} \quad (12)$$

Calculating the $\Phi(F_n)$ for other patterns, except \square -shape, is treated in a similar manner. For \square -shaped fault pattern, the number of ways to construct a rectangular region from n nodes should satisfy the following conditions

$$\sum_{x_i} L(x_i, n(x_i-1)/x_i) \quad (13)$$

where n divides x_i and is described in terms of its power-prime factorization [17]. That is

$$n = \prod_{i=1}^m x_i^{\alpha_i} \quad (14)$$

In Table 1, we list the mathematical equations that are used to calculate the number of occurrences of the most common fault patterns in an $R \times C$ mesh network. Note that, all equations hold when $R=C$. However, in the case that $R \neq C$, $L(i, x)$ will be replaced by the following equations

$$L(i, x) = (R-i) \times (C-x) \quad (15)$$

or

$$L(i, x) = (C-i) \times (R-x) \quad (16)$$

4. Simulation experiments

In the previous section, we have derived mathematical expressions to calculate the total number of occurrences of pattern F_n conditioning on n faulty nodes that can arise in the mesh networks. These analytical expressions form the core of other fault patterns enumeration in other topologies and can be extensively generalized.

Table 1: The mathematical expressions to calculate the number of the most common fault patterns in a mesh network[†].

Fault Pattern	Mathematical expression
l-shape	$(R-n) \times C$ or $(C-n) \times R$
=-shape	$\sum_{i=0}^n (\alpha \times \beta) \times \sum_{j=0}^{C-3} \gamma$ $\alpha = R - i, \beta = R - x + 2, \gamma = C - 2 - j, \beta \leq R; \alpha, \beta, \gamma > 0$
ll-shape	$\sum_{i=0}^n (\alpha \times \beta) \times \sum_{j=0}^{R-3} \gamma$ $\alpha = C - i, \beta = C - x + 2, \gamma = R - 2 - j, \beta \leq C; \alpha, \beta, \gamma > 0$
□-shape	$\sum_{x_i} L(x_i, n(x_i - 1)/x_i) ; n = \prod_{i=1}^m x_i^{\alpha_i}$ and n divides x_i
L-shape	$4 \times \sum_{i=1}^{n-2} L(i, x)$
T-shape	$4 \times \sum_{k=2}^{\infty} \left(\sum_{i=0}^{n-2k-3} L(i-2, x-k-2) + L(n-2k-4, k-1) \right)$
U-shape	$8 \times \sum_{k=2}^{\infty} \left(\sum_{i=0}^{n-2(k-1)} L(i-2, x-k-2) + L(n-2k-3, k-1) \right)$
+shape	$\sum_{k=1}^{n-3} \sum_{j=1}^{n-k-2} \sum_{i=1}^{n-2} L(i, x)$
H-shape	$\sum_{j=1}^{n-7} \sum_{l=1}^{n-6-j} U_{mesh}$

[†] For the case $R \neq C$, the function $L(i, x)$ is set to $(R - i) \times (C - x)$, or $(C - i) \times (R - x)$.

Table 2: Simulation results of the number of fault patterns in the mesh networks for several combinations of network sizes and different number of failed nodes.

Fault Pattern	n	2 × 3	4 × 4	5 × 6	8 × 8	10 × 10	13 × 4	14 × 18	17 × 22	20 × 20	32 × 32
l-shape	5	0	0	16	64	120	36	376	592	640	1792
	8	0	0	0	16	60	24	280	475	520	1600
	15	0	0	0	0	0	0	56	202	240	1152
U-shape	5	2	24	62	168	288	114	824	1270	1368	3720
	8	0	32	238	1028	2012	458	6724	10806	11732	33956
	15	0	0	0	704	3408	360	25024	46550	51888	180000
T-shape	5	0	40	136	424	760	256	2296	3592	3880	10792
	8	0	0	100	1040	2360	460	8960	14810	16160	48560
	15	0	0	0	24	840	96	19656	42432	48360	190632
□-shape	5	0	0	0	0	0	0	0	0	0	0
	8	0	6	22	70	126	42	382	598	646	1798
	15	0	0	10	48	96	18	328	530	576	1680
+shape	5	0	4	12	36	64	22	192	300	324	900
	8	0	0	54	384	824	118	3024	4974	5424	16224
	15	0	0	0	36	1204	62	20020	40898	46332	176748
H-shape	5	0	0	0	0	0	0	0	0	0	0
	8	0	20	85	300	560	155	1780	2825	3060	8700
	15	0	0	16	8744	39776	5936	235320	421854	467136	1553904
ll-shape (vertical)	5	0	60	456	3444	10224	1320	84552	201120	233244	1727940
	8	0	3	210	3087	11340	66	120666	299880	341145	2724435
	15	0	0	0	84	2016	0	126672	376320	397404	4127340
=-shape (horizontal)	5	0	60	440	3444	10224	1572	84320	200760	233244	1727940
	8	0	3	100	3087	11340	2016	111384	282240	341145	2724435
	15	0	0	0	84	2016	1092	76160	276360	397404	4127340
L-shape	5	0	40	136	424	760	256	2296	3592	3880	10792
	8	0	0	40	416	944	184	3584	5924	6464	19424
	15	0	0	0	4	140	16	3276	7072	8060	31772

```

00 Subroutine MakePattern (PatternType, FaultyNodes)
01 for all available shapes with PatternType and FaultyNodes
02   do build four main vertices for each fault pattern
03 Program PatternCount
04   input Row, Col, FaultyNodes, PatternType
05   Count ← 0
06 if construction of the pattern with given FaultyNodes is
07   impossible then return 0
08   MakePattern (PatternType, FaultyNodes)
09   for each constructed pattern do
10     Shape.Width ← Max (width of four main vertices)
11     – Min (width of four main vertices)
12     /* Shape.Width is the width of a specific pattern */
13     Shape.Height ← Max (height of four main vertices)
14     – Min (height of four main vertices)
15     /* Shape.Height is the height of a specific pattern */
16     for i ← Shape.Width to Row do
17       for j ← Shape.Height to Col do
18         Count ← Count +1
19   return Count

```

Figure 5: Pseudocode to compute the number of fault patterns in a mesh network.

An experimental approach is necessary to verify the mathematical expressions that have been developed in this paper. A program has been developed which simulates the analytical model. The objective of the simulation is to enumerate the number of fault pattern occurrences in the mesh networks for different number of faulty nodes. Figure 5 indicates the algorithm used in the simulation methodology. Table 2 reveals the results obtained from simulation experiments and mathematical expressions in the 2×3, 4×4, 5×6, 8×8, 10×10, 13×4, 14×18, 17×22, 20×20, and 32×32 mesh networks when the number of faulty nodes is set to $n = 5, 8, 15$.

5. Conclusions

A large number of fault-tolerant routing algorithms have been proposed in the literature for parallel computer systems. To study the performance behavior of these algorithms, one must investigate the topological shapes of fault patterns, which can be categorized to convex (I-shape, II-shape, and □-shape) and concave (L-shape, U-shape, +-shape, T-shape, and H-shape) regions. In this paper, we have derived mathematical expressions to calculate the number of common fault patterns in the mesh networks. The novelty of the proposed expressions lies in their simplicity and modularity. Since, successive failures might eventually lead to a situation of disconnected network topology, attempting to propose a probabilistic measure of network disconnection expressed as a probability of occurrences of a fault pattern may be a

more challenging extension of our approach in future directions.

References

- [1] P. P. Pande, et al., Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures, *IEEE Trans. Computers*, Vol. 54, No. 8, pp. 1025-1040, 2005.
- [2] W.J. Dally and B. Towles, *Principles and practices of interconnection networks*, Morgan Kaufman Publishers, 2004.
- [3] J. Duato, S. Yalamanchili, L.M. Ni, *Interconnection networks: An engineering approach*, Morgan Kaufmann Publishers, 2003.
- [4] P. Guerrier, A. Greiner, A Generic Architecture for On-Chip Packet-Switched Interconnections, *Proc. Design and Test in Europe (DATE)*, pp. 250-256, Mar. 2000.
- [5] J. Liang et al. aSOC: A Scalable, Single-Chip Communication Architecture, *PACT'2000*, IEEE Computer Society, Los Alamitos, 2000.
- [6] S. Kumar et al., A Network on Chip Architecture and Design Methodology, *Proc. Int'l Symp. VLSI (ISVLSI)*, pp. 117-124, 2002.
- [7] F. Karim et al., An Interconnect Architecture for Networking Systems on Chips, *IEEE Micro*, Vol. 22, No. 5, pp. 36-45, Sept./Oct. 2002.
- [8] S. Chakravorty, L. V. Kalé, A Fault Tolerant Protocol for Massively Parallel Systems, *IPDPS* 2004.
- [9] J. N. Al-Karaki, Performance Analysis of Repairable Cluster of Workstations, *IPDPS* 2004.
- [10] D. Karimou, J. Myoupo, A Fault-Tolerant Permutation Routing Algorithm in Mobile Ad-Hoc Networks, *Lecture Notes on Computer Science*, Vol. 3421, pp.107-115, 2005.
- [11] G. Gupta, M. Younis, Fault-tolerant clustering of wireless sensor networks, *IEEE Conf. on Wireless Communications and Networking*, pp. 1579-1584, March 2003.
- [12] 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.
- [13] J. Wu, Z. Jiang, On Constructing the Minimum Orthogonal Convex Polygon in 2-D Faulty Meshes, *IPDPS* 2004.
- [14] M. E. Gómez, et al., A New Adaptive Fault-Tolerant Routing Methodology for Direct Networks, pp. 462-473, *HiPC* 2004.
- [15] J. M. Montañana, et al., A Transition-Based Fault-Tolerant Routing Methodology for InfiniBand Networks, *IPDPS* 2004.
- [16] R. V. Boppana, S. Chalasani, Fault-Tolerant Wormhole Routing Algorithms for Mesh Networks, *IEEE Trans. Computers*, Vol. 44, No. 7, pp.848-864, 1995.
- [17] K. H. Rosen, *Discrete Mathematics and its Applications*, McGraw-Hill, 5th Ed., New York, 2002.