Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 10 (11): 903-908, 2010
ISSN 1812-5654

Fault-Tolerant Routing in Butterfly Networks

Mohammed H. Mahafzah
Department of Computer Science, Faculty of Information Technology,
Philadelphia University, Amman, 19392, Jordan

Abstract: This research shows that Butterfly networks can be fault-tolerant using Masked Interval Routing Scheme (MIRS). The MIRS was introduced with the aim of compressing the routing tables in a network. It was shown that MIRS could drastically reduce interval information stored in networks such as globe and hypercube graphs, compared to the classical Interval Routing Scheme (IRS). In Butterfly graphs of O(N) vertices the number of intervals per edge goes down from $\Omega\left(\sqrt{N/\log N}\right)$ in IRS to O(logN) in MIRS. This research shows that MIRS may be advantageously used in Butterfly networks, proving that optimal routing with one interval per edge is still possible with a harmless subset of faulty vertices. This research gives an optimal algorithm to reconfigure the intervals in the presence of faults.

Key words: Distributed networks, butterfly network, Interval Routing Scheme (IRS), Masked Interval Routing Scheme (MIRS)

INTRODUCTION

The routing of messages in a network is one of the most fundamental tasks in parallel and distributed systems. The number of edges between any two vertices in a network measures the cost of sending a message between them. Therefore, it is highly desirable to route along short paths. At each intermediate vertex, path information is needed to guide the message to the destination. The simplest approach to store this information is by maintaining a complete routing table at each of the n vertices, which gives for possible destination the name of the next vertex on a shortest path to the destination. This approach requires that n-1 items of routing information be stored at each vertex in the network, with each item being a vertex name. Therefore, a more compact way is needed for representing the routing table. One way to make the tables more compact is to establish some relation among the group of vertices associated with an edge (Tan and van Leeuwen, 1995). A well-known approach that has been used to compact such groups of vertices is Interval Routing Scheme (IRS).

Interval routing was introduced by Santoro and Khatib (1985). It has found industrial applications in the C104 Router Chip used in the (INMOS, 1991) T9000 Transputer design, (Welch *et al*., 1993) and is clearly reviewed by Gavoille (2000). This method assigns a distinct label from the set {0, ..., n-1} for each vertex. Each edge is labeled with a unique subinterval of the interval [0, n-1]. The set of intervals associated with the edges of a vertex must be disjoint and their union covers the interval [0, n-1]. In general, all subintervals can wraparound. This is denoted as IRS. As a result, a routing table of d entries is needed at a vertex, where d is the degree of the vertex. When a message with a destination x arrives at an intermediate vertex, a comparison between x and the vertex label is made. If they match, the message is consumed at the vertex. Otherwise, the local routing table is searched for the interval containing x and the corresponding edge is selected (Bakker *et al*., 1991).

The routing scheme is called optimal if the messages are routed along shortest paths. Hereafter, it refers to optimal IRS simply by IRS. The size of the routing tables could be reduced at the cost of increasing the lengths of the routing paths (Gavoille and Guevermont, 1998; Peleg and Upfal, 1989). If you allow multiple labels to be associated with every edge, then you have a multi-label interval routing scheme k-IRS, when there are at most k intervals per edge, k>0. This scheme Van Leeuwen and Tan (1987) deals with certain graphs that do not have optimal routing with one interval per edge (Kranakis *et al*., 1996), (1-IRS).

Another approach called Masked Interval Routing Scheme (MIRS), introduced by Luccio *et al*. (1998) makes interval routing more flexible and deals with networks that are difficult to deal with in IRS. MIRS introduced as a new interval routing scheme, where a mask is added to each interval to indicate particular subsets of consecutive labels.

In MIRS the labels can be consecutive in many different ways. According to MIRS an interval is expressed through three integers [Start, Stop, Mask],

Table 1: Three intervals in MIRS and their binary interpretation

Interval	Labels specified							
[1, 15, 1]	1	3	5	7	9	11	13	15
[000$\underline{1}$, 111$\underline{1}$]	0001	0011	0101	0111	1001	1010	1101	1111
[0, 10, 5]	0	2	8	10				
[00$\underline{0}$0, 101$\underline{0}$]	0000	0010	1000	1010				
[2, 14, 3]	2	6	10	14				
[00$\underline{11}$, 11$\underline{10}$]	0010	0110	1010	1110				

(S, T and M, for short). These three integers indicate the initial, final labels of the interval and its mask respectively. The three integers have to be represented in binary and the bits equal to 1 in the mask indicate that the bits in the corresponding positions of the start and stop labels remain unchanged for all the labels of the interval. For instance, the interval [0, 8, 2] specifies the labels 0, 1, 4, 5, 8, since S = 0000 (using four bits), T = 1000 and the mask M = 0010 means that the second less significant bit remains unchanged for all the labels of the interval; note that, by ignoring the second bit, all these numbers are consecutive. Significant results are shown in Table 1; several sets of intervals obtained with different masks (notice the masked bits are underlined).

By introducing the mask, a large variety of sets of consecutive labels can be obtained, thus, making interval routing more flexible. For example, using MIRS and labels of r bits and masks contain i ones, you can express $2^i\binom{r}{i}$ sets of 2^{r-i} consecutive elements, where the bits in the non masked positions are all 0's in S and all 1's in T. To represent the intervals you need 3 $\log_2 n$ bits, where n is the number of the vertices of the graph, hence n-1 is the value of the maximum label.

This research shows that MIRS can be advantageously used in fault-tolerant butterflies with a harmless subset of faulty vertices.

MIRS ON BUTTERFLY NETWORK

The family of butterfly graph is well known (Leighton, 1992), a butterfly BF(n) consists of $N = 2^n (n+1)$ vertices arranged in 2^n rows numbered from 0 to 2^n-1 and n+1 levels (columns) numbered from 0 to n. Each vertex v is denoted by a pair of integers (i, j) indicating the level and the row of v, respectively. An edge connects two vertices (i, j), (h, k) if h = i -1 and j = k (straight edge) or the binary representations of j and k differ in the hth bit (cross edge). Figure 1 shows BF(3) with $N = 2^3.4 = 32$ vertices arranged in four levels numbered 0..3 and eight rows numbered 0...7 in binary.

The vertices are numbered from 0 to N-1 level by level, from right to left. In fact, a vertex (i, j) is labeled with the integer formed by the concatenation i j, with i represented in decimal and j represented in binary as $j_n j_{n-1} \ldots j_1$. For example, vertex (2, 5) of BF(3) is labeled 2101.

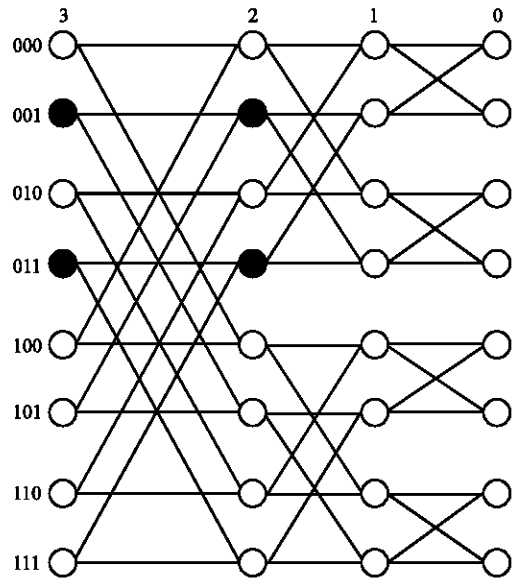

Fig. 1: Butterfly BF(3) with $N = 2^3$, 4 = 32 vertices. The four dark vertices form the interval [2001, 3011]

The reason for this mixed representation is to enhance readability, taking into account that only the bits of j will be masked and need to be explicitly indicated under the compact notation with underlining for mask. For example, the interval [2001, 3011] with two masked bits specifies the four vertices 2001, 2011, 3001, 3011 in BF(3). The four dark vertices in Fig. 1 form the interval [2001, 3011].

To apply MIRS to the butterfly you first need to determine a shortest path between any two vertices S, T, or better the initial edge of such a path which leaves S and whose associated interval contains T. Let S and T be labeled by $sa_n \ldots a_1$ and $tb_n \ldots b_1$ respectively. Let e_1, e_2, e_3, e_4 be the edges leaving S, where e_1, e_2 are the straight and cross edges from level s to level s-1 (for s>0) and e_3, e_4 are the straight and cross edges from s to level s+1 (for s<n), respectively. These edges are shown in Fig. 2 for vertex 1100. For $a_n \ldots a_1 \neq b_n \ldots b_1$, let r be the smallest value i for which $a_i \neq b_i$; and let l be the greatest value i for which $a_i \neq b_i$. The following lemma detects the first edge of a shortest path between S and T as a function of s, t, r, l. The proof easily follows from known properties of the butterfly.

Lemma 1: In BF(n) there exists a shortest path between two arbitrary vertices S and T with the following first edge f(S,T):

$$\text{For } t \geq s \geq r: \; f(S, T) = \begin{cases} e_1, \text{ if } a_s = b_s \\ e_2, \text{ if } a_s \neq b_s \end{cases}$$

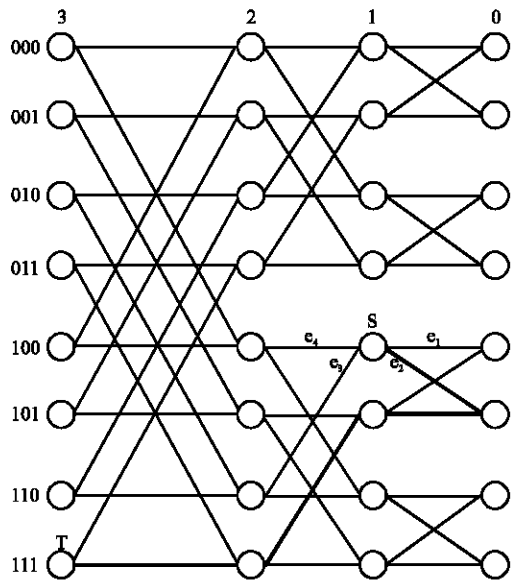

Fig. 2: The shortest path between S = 1100 and T = 3111 is shown with darkened lines in butterfly BF(3)

$$\text{For } s>t \text{ and } s \geq l \text{ (if } l \text{ exists): } f(S,T) = \begin{cases} e_1, \text{if } a_s = b_s \\ e_2, \text{if } a_s \neq b_s \end{cases}$$

$$\text{For } l > s > t\text{: } f(S,T) = \begin{cases} e_3, \text{if } a_{s+1} = b_{s+1} \\ e_4, \text{if } a_{s+1} \neq b_{s+1} \end{cases}$$

$$\text{For } t \geq s \text{ and } r>s \text{ (if r exists): } f(S,T) = \begin{cases} e_3, \text{if } a_{s+1} = b_{s+1} \\ e_4, \text{if } a_{s+1} \neq b_{s+1} \end{cases}$$

Lemma 1 covers all the possible situations for S and T. In particular, for s = 0, e_1 and e_2 are undefined and the comparison between a_s and b_s does not apply (cases-1 and 2). For s = n, e_3 and e_4 are undefined and the comparison between a_{s+1} and b_{s+1} does not apply (cases-3 and 4). For example let S = 1100 and T= 3111 in BF(3), as shown in Fig. 3. You have s = 1, t = 3, r = 1, l = 2 and case-1 applies with $a_s \neq b_s$. Then f(S,T) = e_2 and the shortest path between S and T proceeds. The shortest path between S and T is shown with darkened lines. Now, it can be shown how to construct efficient MIRS on butterflies. For a bit x, let $\bar{x}$ denote its complement.

Theorem 1: for any vertex S labeled $sa_n \ldots a_1$ in BF (n), MIRS can be built as follows:

1 Intervals on e_1:
   1.1 $[s0\ldots0a_s\,0\ldots0\,\underline{a_r^- a_{r-1}}, n1\ldots1a_s\,1\ldots1\,\underline{a_r^- a_{r-1}\ldots a_1}]$, for l = r = s-1
   1.2 $[0\,\underline{a_n\ldots a_s}\,0\ldots0, (s-1)\,\underline{a_n\ldots a_s}\,1\ldots1]$
2 Intervals on e_2:
   2.1 $[s0\ldots0\,\underline{a_s^-}\,0\ldots0, n1\ldots1\,\underline{a_s^-}\,1\ldots1]$
   2.2 $[0an,,,\,\underline{a_{s+1}\,a_s^-}\,0\ldots0, (s-1)\,\underline{a_n\ldots a_{s+1}\,a_s^-}\,1\ldots1]$
3 Intervals on e_3:
   3.1 $[0\,\underline{a_n\ldots a_{l+1}a_l^-}\,0\ldots0\,\underline{\bar{a}_{l+1}}\,0\ldots0, (s-1)\,a_n\ldots a_{l+1}a_l^-\,1\ldots1\,\underline{\bar{a}_{l+1}}\,1\ldots1]$, for s+2 = l = n
   3.2 $[s0\ldots0\,\underline{a_{s+1}\ldots a_1}, n1\ldots1\,\underline{a_{s+1}\ldots a_1}]$
4 Intervals on e_4:
   4.1 $[00\ldots0\,\underline{\bar{a}_{s+1}}\,0\ldots0, (s-1)1\ldots1\,\underline{\bar{a}_{s+1}}\,1\ldots1]$
   4.2 $[s0\ldots0\,\underline{\bar{a}_{s+1}\,a_s\ldots a_1}, n1\ldots1\,\underline{\bar{a}_{s+1}\,a_s\ldots a_1}]$

Proof: Intervals on e_1. The intervals grouped under 1.1 are built according to case 1 of lemma-1 with $a_s = b_s$. In fact, all destination vertices T labeled $tb_n\ldots b_1$, with $t \geq s \geq r$ and $b_s = a_s$, are reached through e_1. For any value of r such vertices have $b_r \neq a_r$ and $b_{r-1}\ldots b_1 = a_{r-1}\ldots a_1$, which implies that the values $a_r^-\,a_{r-1}\ldots a_1$ and a_s are fixed for all these vertices, while the other bits take all possible values. By the label structure, for any given value of r all the destinations T from an interval masked in positions s, r, r -1, ..., 1.

The interval marked 1.2 is built according to case 2 of lemma-1 with $a_s = b_s$. All the destination vertices T with s>t, s ≥ 1 and $b_s = a_s$, are reached through e_1. These vertices share with S all the bits in positions n, n-1, …, s, hence they form an interval masked in such positions.

The proof for the intervals on e_2, e_3 and e_4 proceeds by similar inspection. In particular the intervals marked 2.1 and 2.2 are respectively built according to cases-1 and 2 of lemma-1, with $a_s \neq b_s$; the intervals marked 3.1 and 3.2 are built according to cases 3 and 4 of lemma-1, with $a_{s+1} = b_{s+1}$; and the intervals marked 4.1 and 4.2 are built according to cases 3 and 4 of lemma-1, with $a_{s+1} \neq b_{s+1}$.

Consider BF(4) in Fig. 3, only the upper half of the graph (rows 0000 to 0111) is shown. According to Theorem-1, the intervals on the edges leaving from vertex S labeled 30110 are the following:

- Intervals on e_1: $\alpha = [30\underline{10}\underline{1}, 41\underline{11}\underline{1}]$, $\beta = [30\underline{100}, 41\underline{100}]$, $\gamma = [0\underline{01}00, 2\underline{01}11]$
- Intervals on e_2: $\delta = [30\underline{0}00, 41\underline{0}11]$, $\epsilon = [0\underline{00}00, 2\underline{00}11]$
- Intervals on e_3: $\phi = [30110, 40110]$
- Intervals on e_4: $\iota = [0\underline{1}000, 2\underline{1}111]$, $\omega = [3\underline{1110}, 4\underline{1110}]$

Note that there are three intervals on e_1, with the ones denoted by α and β corresponding to group 1.1 Theorem-1 (we have s-1 = 2 intervals) and γ corresponding to 1.2. There is only one interval ϕ on e_3, corresponding to 3.2 and including S (see observation-1). The intervals grouped under 3.1 in the theorem are defined for s + 2 ≤ n, hence do not exist here because you

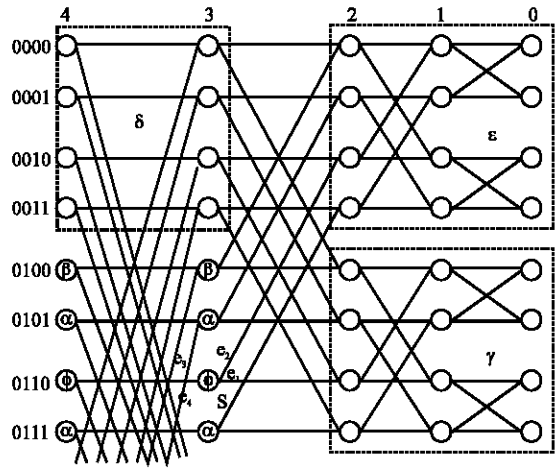

Fig. 3: MIRS for vertex 30110 in BF(4) according to Theorem-1

have s = 3 = n-1. All the vertices in intervals γ, ϵ and ϕ belong to the upper half of the butterfly. The vertices in the other intervals are splitted between the two halves of the graph, except for the ones in ι that are fully contained in the lower half.

As already mentioned, the butterfly requires $\Omega\left(\sqrt{N/\log N}\right)$ intervals per edge (Kralovic *et al.*, 2000) in any IRS. We obtain a remarkable decrement in the number of intervals using a MIRS constructed as indicated in Theorem-1.

Corollary 1: BF(n) admits a MIRS with n intervals.

Proof: The number of intervals on edge e_1 is s - 1 + 1 = s ≤ n, where the maximum occurs for the vertices at level n. Similarly, the number of intervals on edge e_3 is n-(s+2)+1+1= n-s ≤ n, where the maximum occurs for the vertices at level 0. Finally, only two intervals are present on edges e_2 and e_4. Recalling that BF(n) has $N = 2^n(n+1)$ vertices, from corollary-1, an upper bound of O(log N) intervals per edge can be immediately derived.

Observation 1: The interval marked 3.2 in Theorem-1 includes the label $sa_n \ldots a_1$, that is it includes S. In general this is not a drawback because S is never searched for from S itself. If this is not acceptable, it can be shown that interval 3.2 can be broken into O(log N) disjoint intervals not including S. The upper bound of O(log N) intervals per edge still holds.

The use of an increased set of labels can be applied to particular sub-graphs of a butterfly, to maintain a low number of intervals per edge. First note that BF(n) includes 2^{n-r} smaller butterflies BF(r), 0≤r≤n-1, which

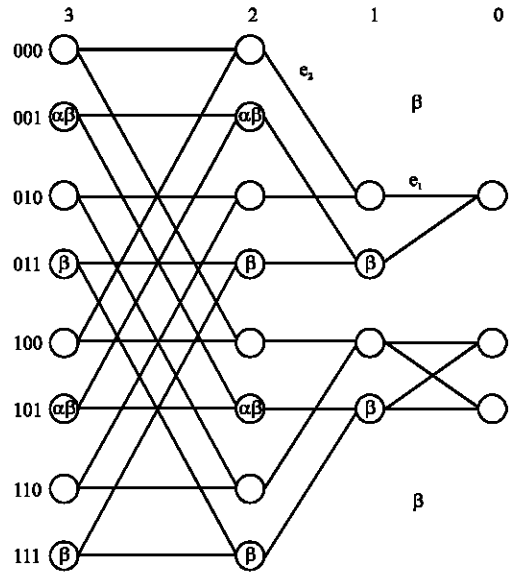

Fig. 4: Extended butterfly BF(3,1) with increased set of labels {0000, ..., 3111} pertaining to BF(3)

occupy the levels 0 to r of BF(n), (Fig. 3). The vertices of each sub-graph BF(r) are identified by constant values of the row-label bits $j_n \ldots j_{r+1}$ (the upper BF(1) in BF(3) is identified by the bit values j_3j_2=00). Two BF(r) are adjacent if their vertices have the same values of $j_n \ldots j_{r+2}$ (The upper two BF(1) in BF(3) are adjacent, since they are identified by j_3=0). An extended butterfly BF(n,k), 0≤k≤n-1, is obtained by removing from BF(n) some non-adjacent sub-graphs BF(r) with possibly different values r≤k. The extended butterfly BF(3,1) shown in Fig. 4 is obtained from BF(3) removing two sub-graph BF(1) and one sub-graph BF(0) consisting of the single vertex 0011. Note that each vertex of BF(n, k) maintains the leaving edges e_3, e_4, but edge e_1, or e_2 may be missing. This happens for the critical vertices, that is the ones for which an adjacent vertex of BF(n) has been removed (all the vertices of level 2 in Fig. 4).

For BF(n,k),an increased set of labels is adopted. The same set of labels of BF(n) with the vertices of BF(n,k) maintaining the labels of the corresponding vertices of BF(n) and the unused labels being dummy. Consider now two arbitrary vertices u, v of BF(n,k) and the minimal path π from u to v in BF(n) determined by the routing intervals of Theorem-1. If all the edges of π are present in BF(n,k), then this path will also be adopted in the routing of BF(n,k). If instead, going from u to v along π, we encounter a vertex z that is critical in BF(n,k) and π proceeds through the missing edge e_1 (respectively e_2) we must change the path. It is easy to verify that there is another minimal path π' in BF(n) that goes along e_2

(respectively e_1) and is totally contained in BF(n,k). The π' is adopted in the routing of BF(n,k). Now, it is not difficult to verify that the following construction generates valid MIRS intervals for BF(n,k):

- For each non-critical vertex, build the intervals on e_1, e_2, e_3 and e_4 as for the corresponding vertex of BF(n)
- For each critical vertex with missing edge e_1, build the intervals on e_2, e_3 and e_4 as for the corresponding vertex of BF(n) and add on e_2 the intervals 1.1 of Theorem-1. The interval 1.2 is ignored, because is entirely composed of dummy labels
- For each critical vertex with missing edge e_2, build the intervals on e_1, e_3 and e_4 as for the corresponding vertex of BF(n) and add on e_1 the interval 2.1 of Theorem-1. The interval 2.2 is ignored, because is entirely composed of dummy labels

In Fig. 4, edge e_1 from vertex 2000 is missing, then the interval α = [2001, 3101] is attached to e_2 (α is the only interval in the set 1.1, see Theorem-1 with s = 2). Edge e_2 from vertex 1010 is missing, then the interval β = [1001, 3111] is attached to e_1

Corollary 2: BF(n,k) admits a MIRS with n+1 intervals.

Proof: Immediately from the procedure above. The number of intervals on e_1, for missing e_2, or on e_2, for missing e_1, assumes maximum value n +1 for s = n in Theorem-1.

RECONFIGURATION ALGORITHM

The reconfiguration procedure for the 1-MIRS in the presence of faults can be derived as follows: Let X be any vertex with standard binary label $x_{k-1}...x_0$ and Y be another vertex with standard binary label $y_{k-1}...y_0$ adjacent to X. Assume that vertex Y belongs to the harmless subset of vertices F, which is now failed. Now we can rearrange the intervals of the edges leaving X to cope with this new situation. Let j be the bit position at which X and Y differ. Considering first the right-most bit of X, x_0, we assign the interval [x_{k-1}..... x'_0, x_{k-1} ... x'_j... x'_0, 1…101…1], where the 0 is at bit position j, to edge (X- x_{k-1}..... x'_0). Secondly, consider x_1 and assign the interval [x_{k-1}..... x'_1x_0, x_{k-1}... x'_j... $x_1$1, 1…101…10], where the 0's are at bit positions j and 0, to edge (X- x_{k-1}..... x'_1x_0). Now consider x_2 and assign the interval [x_{k-1}..... $x'_2x_1x_0$, x_{k-1}... x'_j ... $x'_2$11, 1…101…100], where the 0's are at bit positions j, 0 and 1, to edge (X-x_{k-1}..... $x'_2x_1x_0$). Similarly on all other bits I=3,…,k-1 except where i = j to obtain the remaining intervals. The pseudo-code for the reconfiguration procedure is given:

Algorithm reconfigure (Y):

```
/* This algorithm is executed by vertex X upon failure of its neighbor vertex Y∈F
/* Assume W, Z and M are the initial, final and mask for the constructed intervals
/* That is, the constructed intervals will be in the form [W, Z, M]
/* let j be the bit position at which X and Y differ
begin
    M = 11…11; let m_j = 0;
    Z = X; z_j = x'_j;
    for I = 0 to k -1 do
    begin
        if i=j then continue; (i.e., next i)
        W = X; w_j =x'_i ;
        z_i = x'_i ;
        assign the interval [W, Z, M] to edge (X-W) ;
        m_i = 0; z_i = 1;
    end
End Reconfigure
```

The algorithm is clearly very efficient and can be executed independently and in parallel, by each vertex X, provided that X knows the identity of a faulty neighbor Y (if any) and there is a global guarantee that the faults belong to a harmless set of vertices. Obviously, it can be seen that the algorithm takes O(k) since all the bits of X will be processed (except the bit at which X differs from Y) and the standard binary label for any vertex is k bits.

CONCLUSIONS AND FUTURE RESEARCH

This research reviews the new technique of Masked Interval Routing Scheme (MIRS), which was introduced with the aim of compressing the routing tables stored in network. It was shown that MIRS could drastically reduce interval information stored in networks such as globe and hypercube graphs compared to the classical Interval Routing Scheme (IRS). In Butterfly graphs of O(N) vertices the number of intervals per edge goes down from $\Omega\left(\sqrt{N/\log N}\right)$ in IRS to O(logN) in MIRS.

This study shows that MIRS may be advantageously used in fault-tolerant networks, proving that optimal routing with one interval per edge is still possible in Butterflies with a proper subset of faulty vertices. This research gives the algorithm needed to reconfigure the intervals in the presence of faults in Butterfly networks. Further research needed in the reconfiguration problem of faulty networks considering different topologies. The definition of a harmless subset of faulty vertices should be generalized.

REFERENCES

Bakker, E.M., R.B. Tan and J. van Leeuwen, 1991. Linear interval routing schemes. Comput. J., 30: 298-307.

Gavoille, C. and E. Guevermont, 1998. Worst case bounds for shortest path interval routing. J. Algorithms, 27: 1-25.

Gavoille, C., 2000. A survey on interval routing. Theor. Comput. Sci., 245: 217-253.

INMOS, 1991. The T9000 Transputer Overview Manual. INMOS Ltd., Bristol, UK.

Kralovic, R., P. Ruzica and D. Stefancovic, 2000. The complexity of shortest path and dilation bounded interval routing. Theor. Comput. Sci., 234: 85-107.

Kranakis, E., D. Krizanc and S.S. Ravi, 1996. On multi-label linear interval routing schemes. Comput. J., 39: 133-139.

Leighton, F.T., 1992. Introduction to Parallel Algorithms and Architecture: Arrays, Trees, Hypercubes. Morgan Kaufmann, SanMateo, CA.

Luccio, F., M. Mahafzah, M. Omari and L. Pagli, 1998. Routing with use of masks. Proceedings of the 5th International Colloquium on Structural Information and Communication Complexity, June 22-24, Amalfi, Italy, pp: 188-200.

Peleg, D. and E. Upfal, 1989. A trade-off between space and efficiency for routing tables. J. ACM, 36: 510-530.

Santoro, M. and R. Khatib, 1985. Labeling and implicit routing in networks. Comput. J., 28: 5-8.

Tan, R.B. and J. van Leeuwen, 1995. Compact routing methods: A survey. Proceedings of the Colloquium on Structural Information and Communication Complexity, (SICC95), School of Computer Science, Carleton University, Ottawa, pp: 99-109.

Van Leeuwen, J. and R. Tan, 1987. Interval routing. Comput. J., 30: 298-307.

Welch, P.H., M.D. May and P.W. Thompson, 1993. Networks, Routers and Transputers: Function, Performance and Application. IOS Press, Netherlands, ISBN 90-5199-129-0.