Università di Pisa

Dipartimento di Informatica

Technical Report : TR-97-23

Masked Interval Routing: a New Routing Scheme

F. Luccio, M. Mahafzah, M. Omari, L. Pagli

December 05, 1997

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

Masked Interval Routing: a New Routing Scheme *

F. Luccio † M. Mahafzah ‡ M. Omari § L. Pagli ¶

Abstract

We introduce the new *Masked Interval Routing Scheme*, *MIRS* for short, where a mask is added to each interval to indicate particular subsets of "consecutive" labels. Interval routing becomes more flexible, with the classical *IRS* scheme being a special case of *MIRS*. We then take two directions. First we show that the interval information stored in the network may be drastically reduced in the hard cases, proving that in globe graphs of $O(n^2)$ vertices the number of intervals per edge goes down from $\Omega(n)$ to $O(log\ n)$. The technique is then extended to globe graphs of arbitrary dimensions. Second we show that *MIRS* may be advantageously used in fault-tolerant networks, proving that optimal routing with one interval per edge is still possible in hypercubes with a "harmless" subset of faulty vertices. This work is aimed to introducing a new technique. Further research is needed in both the directions taken here. Still, the examples provided show that MIRS may be useful in practical applications.

Keywords: Distributed Network, Interval Routing Scheme, Masked Interval, Globe Graph, Hypercube, Fault Tolerance.

1 Introduction

Message routing between processors is a basic activity that takes place in the network of parallel and distributed systems. In order to route a message from a source processor to a destination processor, a path must be identified. Since the cost of sending a message is measured by the number of links the message has to traverse, it is highly desirable to route along short paths. Representing the network in graph form, processors and links will be respectively called vertices and edges.

Path information must be available at each intermediate vertex to guide the message from its source to its destination. This information is usually stored in a routing table in each of the N vertices of the network, one entry for each possible destination. Although this approach guarantees an optimal routing (shortest path), it is very expensive for large networks since it requires a table of $N - 1$ entries at each vertex. Therefore, a more compact way is needed for representing the routing table.

*This work was partially supported by the Italian Ministery of Foreign Affairs under a Cooperation Project between Italy and Jordan.

†Dipartimento di Informatica, Università di Pisa, Italy

‡Department of Computer Science, Mu'tah University, Jordan

§Department of Computer Science, Mu'tah University, Jordan

¶Dipartimento di Informatica, Università di Pisa, Italy

The tables can be made more compact by establishing some relations among the vertices associated with each edge. An approach to attain such a reduction is *interval routing* introduced by Santoro and Khatib [8], that has found industrial applications in the C104 Router Chip used in the INMOS T9000 Transputer design [3, 7]. An excellent review is contained in [2]. This method assigns a distinct label from the set $\{0, ..., N-1\}$ to 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 wrap-around. This is denoted as *Interval Routing Scheme*, *IRS* for short. As a results, a routing table of d entries is needed at a vertex, where d is the degree of the vertex. This leads to a very simple routing algorithm. When a message with a destination x arrives at an intermediate vertex, a comparison between x and the vertex label is made. If the two agree 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.

The routing scheme is called *optimal* if the messages are routed along shortest paths. From now on we implicitly refer to optimal IRS. If we allow multiple intervals to be associated with every edge, then we have a *multi-interval* scheme *k-IRS* when there are at most k intervals per edge, $k > 0$. This scheme was introduced by van Leeuwen and Tan [6] to deal with certain graphs that do not have optimal routing with one interval per edge (now called 1-*IRS*).

In this paper we introduce the new *Masked Interval Routing Scheme*, *MIRS* for short, where a mask is added to each interval to indicate particular subsets of "consecutive" labels. *k-MIRS* is similarly defined. This idea allows to make the interval routing scheme more flexible, at the expences of a longer specification of each interval. In particular, we first define the new interval routing scheme in section 2, then we apply the new scheme to two important families of graphs, namely globe graphs and hypercubes. Globe graphs are important by a theoretical point of view, since they represent a hard case for *IRS*, as they require $k = \Omega(\sqrt{N})$ intervals per edge to attain shortest path routing [4]. We show that these graphs admit *MIRS* with $k = O(log\ N)$ intervals. The result is then extended to globe graphs of arbitrary shape, with the introduction of dummy labels that simulate a regular structure. This strategy does not introduce routing drawbacks because dummy vertices are never searched for.

Hypercubes are very usefull in practice, since they represent a standard way of connecting processors in a network. Althought hypercubes can be optimally treated with 1-*IRS*, the new scheme allows to make them fault-tolerant in presence of interruptions of vertices, thereby making *MIRS* important in practical applications.

2 The masked IRS

We now introduce the new *Masked Interval Routing Scheme (MIRS)*, where labels can be "consecutive" in many different ways. According to *MIRS* an interval is now expressed throught three integers *Start, Stop and Mask* indicating the initial and final labels of the interval, and its mask, respectivley. In symbols the interval is denoted by $[S,T,M]$, and its three elements, together with all the labels specified by the interval, have to be interpreted as binary numbers. In fact, the 1's in the binary representation of M indicate that the bits in the corresponding positions remain unchanged for all the labels of the interval (therefore those bits must be the same in S and T). For instance, using four bits, the interval [0,9,2] specifies the labels 0, 1, 4, 5, 8, 9. In fact $S = 0000$, $T = 1001$, and the mask $M = 0010$ indicates that the second bit from the

interval	labels specified							
$[1, 15, 1]$	1	3	5	7	9	11	13	15
$[0001, 1111, 0001]$	0001	0011	0101	0111	1001	1011	1101	1111
$[0, 10, 5]$	0	2	8	10				
$[0000, 1010, 0101]$	0000	0010	1000	1010				
$[2, 14, 3]$	2	6	10	14				
$[0010, 1110, 0011]$	0010	0110	1010	1110				

Figure 1: Three intervals in *MIRS* and their binary interpretation. The first one contains all the odd labels. The third one has mask 0011, hence its labels are spaced by 4.

right, which is equal to 0 in S and T, mantains this value, as occurs in the binary representation of the labels reported above. Note that all these labels are consecutive if we ignore the second bit. Other examples are shown in figure 1.

Using *MIRS* we obtain a very flexible interval routing scheme because we can express a large variety of sets of consecutive labels. For example, using *MIRS* with labels of r bits and a mask of i bits, we 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. These sets correspond to all the ways of allocating the 1's in the mask, which indicate the positions of the fixed bits in the labels, that can be in turn equal to 1 or to 0. To represent the intervals we need $3log_2N$ bits, where N is the number of the vertices of the graph, hence $N-1$ is the value of the maximum label.

In *IRS* the search for the interval where the destination label is contained is realized, on each edge, trought a simple variation of binary search on the sequence of interval extremes. In *MIRS* this strategy is no more possible because the pairs S, T related to different intervals may overlap. For instance, we can have the following intervals on the same edge: [2,8,5], [7,15,1], [0,5,2]. The search for a destination can then be organized by comparing for each interval the bits of x with the ones of S (or T) in the masked positions, and, in case of matching, by checking if $S \leq x \leq T$. The search takes linear time in the number of intervals, in the worst case.

To summarize, $MIRS$ is more flexible than IRS but requires a longer specification for the intervals (the length is increased by one third), and a search algorithm linear in the number of intervals on each edge.

3 Application to globe graphs

Globe graphs are known in the literature as graphs paticularly difficult to handle in *IRS* [4]. A globe graph G^r_s, r odd, is constituted by s chains of r vertices, whose starting and ending points are connected to two additional vertices. The vertices of the chains form an $r \times s$ array (see figure 2). In [4] it is proved that any *IRS* for G^n_{n+1} needs $\Omega(n)$ intervals per edge, that is, there are edges in this graph requiring a number of intervals linear in the array dimensions. Globe graph G^7_8 of figure 2, with the labeling shown, has an *IRS* with $n = 7$ intervals on the edges leaving from vertex 57. If consecutive labels are instead placed on the chains, the edges in the

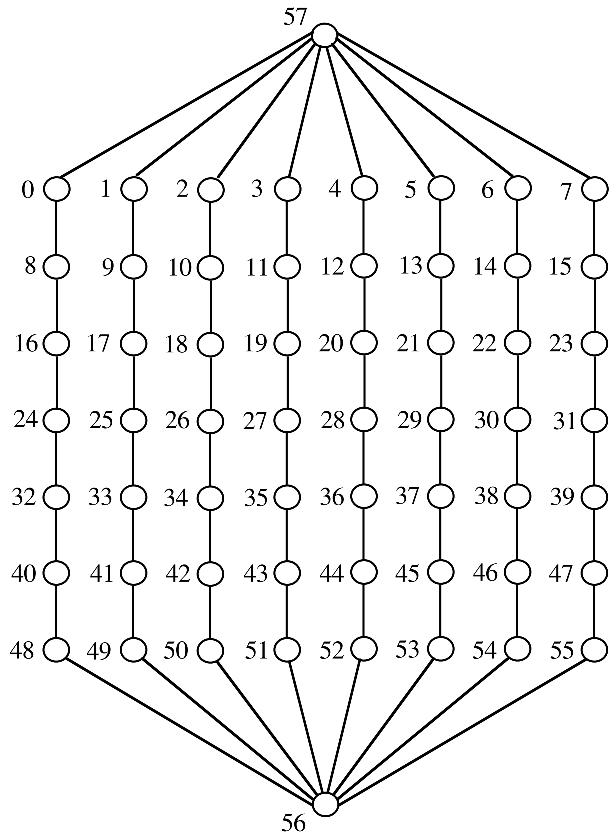

Figure 2: A globe graph G_8^7 with a possible labeling. *IRS* requires $n+1=8$ intervals [57,57], [0,0], [8,8], ..., [48,48] on the edge (56-48). *MIRS* requires $log_2(n+1)+2=5$ intervals [4,39,4], [0,33,6], [3,35,7], [2,10,7], [57,57,0] on the edge (18-10).

middle of the chains require $n+1=8$ intervals.

We now show that globe graphs can be efficiently routed under *MIRS*. First consider the example of figure 2. The edges leaving vertices 56 and 57 have one interval each. In fact the vertices in the columns have labels spaced by 8, hence all interval have mask 000111. For example the interval on edge (57-3) is [3,51,7]; the interval on edge (57-0) is [0,56,7]; the interval on edge (56-49) is [1,57,7]. All the other edges have at most five intervals. Take, for example, the edge (18-10) and proceed as follows:

1. Since (18-10) is on a column in the "left half" of G_8^7, consider first the vertices in the "right half". To reach these vertices from 18, the intervals on (18-10) must include the labels 4, 5, 6, 7, 12, 13, 14, 15, 20, ..., 39. These labels form the unique interval [4,39,4].

2. Consider the columns in the left half. Divide these columns into a left and a right portion, namely the one with columns starting with 0,1, and the one with colums starting with 2, 3. Since (18-10) is in the second portion, build the new interval [0,33,6] that includes the labels 0, 1, 8, 9, 16, ..., 33 of the first portion.

3. Proceed recursively. In this example the only column starting at 3 is left, whose labels

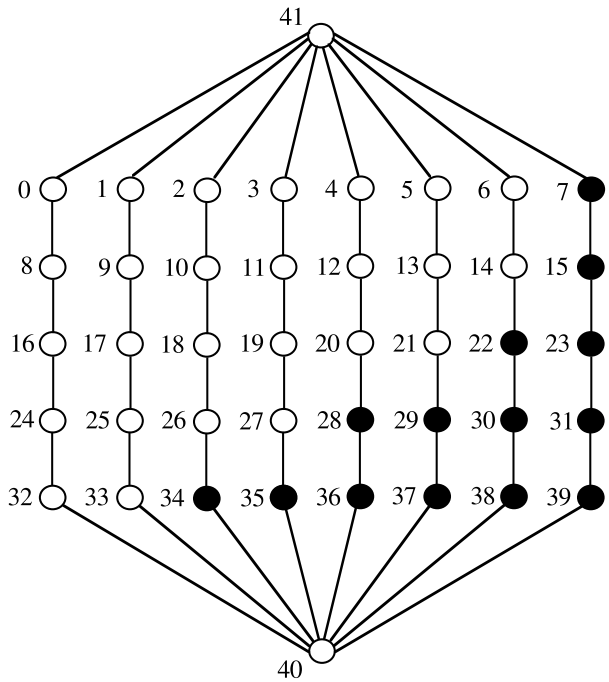

Figure 3: A globe graph G_7^{*5} with $N = 27$ vertices (white dots). Black dots are dummy vertices supporting the increased set of labels $\{0, \ldots, 41\}$.

3,11,19, ..., 35 form the interval [3,35,7].

4. Add the interval [2,10,7] for the column of 18, and [57,57,0] to complete the interval set.

In general we can pose:

Theorem 1 *G_{n+1}^n, with $n + 1 = 2^k$, admits an* MIRS *with $k + 2$ intervals.*
Proof The proof is costructive. The intervals are built with an immediate generalization of the procedure above. □

To handle globe graphs with arbitrary dimensions we can use a new strategy, that may prove useful for interval routing in general. For a graph of N vertices use *increased set of labels* $\{0, 1, \ldots, Q\}$, $Q > N - 1$, so that some labels X will appear in the intervals but are not assigned to any vertex. This does not cause routing drowbacks, since X will never be searched for. The size (number of bits) of the labels increases, however, we will see that this does not pose any problem if Q is "not too large". We have:

Corollary 1 *G_n^m, with $2^{k-1} < n < 2^k$, admits an* MIRS *with $k + 2$ intervals and increased set of labels $\{0, \ldots, 2^k m + 1\}$.*
Proof Border the array to the right with $2^k - n$ additional columns of dummy vertices. Then label the graph and build the intervals as in theorem 1. □

The labelling introduced in corollary 1 at most doubles the number of labels in the set. That is, at most one extra bit is used to code S, T and M in the intervals. Clearly this does not affect the search time tangibly.

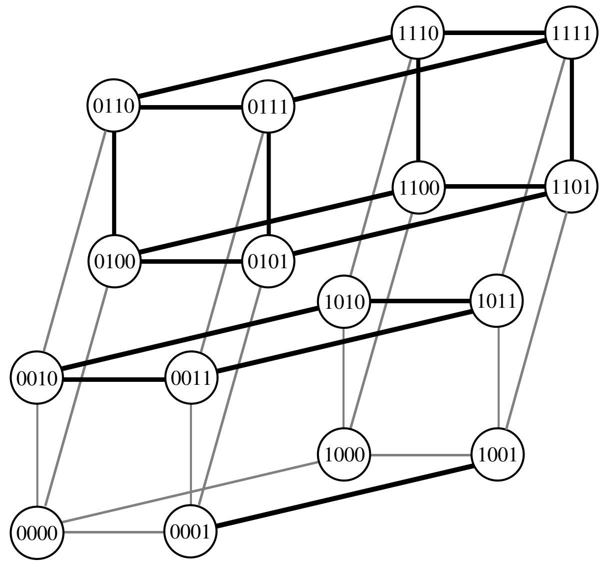

Figure 4: A hypercube of 2^4 vertices with standard binary labelling. Under 1-*IRS*, the intervals assigned to the edges (0000-1000), (0000-0100), (0000-0010) and (0000-0001) are [1000,1111], [0100,0111] [0010,0011] and [0001,0001] respectively (see property 1). The engrossed edges mark a reconfigured partition for vertex 0000 in 1-*MIRS*, under failure of vertex 1000.

To show the possible general advantage in using an increased set of labels, consider an *extended globe graph* G_s^{*r}, where the array has s columns and a variable number t of elements in each column, $1 \leq t \leq r$. For convenience draw G_s^{*r} with the columns ordered from left to right with decreasing number of vertices, as shown in figure 3. We have:

Corollary 2 *G_s^{*r}, with $2^{k-1} < s < 2^k$, admits an* MIRS *with $k+2$ intervals increased set of labels $\{0, \ldots, 2^k r + 1\}$.*
Proof Insert dummy vertices so to have an array of 2^k columns of r vertices each (figure 3). Then label the graph and build the intervals as in theorem 1. $\square$

Clearly, for r and s of the same order, *MIRS* with an extended set of labels is much more efficient than *IRS*.

4 Application to Hypercubes

Hypercube graphs admit an optimal 1-*IRS*, that is one interval per edge is sufficient to find the shortest path for each vertex, even if no wrap-around of the intervals is allowed. See for example [1, 5]. Hypercubes are then easy to handle. This is partly due to the fact that the standard binary addressing implicitly specifies an optimal path to be followed from source to destination. In fact, the addresses of two adjacent vertices differ in exactly one bit. A shortest path between two vertices is determined by changing exactly once (in any order) the bits that differ in the two addresses.

Part of a 1-*IRS* for a hypercube of 2^4 vertices is discussed in figure 4. From each vertex, the four outgoing edges lead to subcubes of 2^3, 2^2, 2^1 and 2^0 vertices, respectively, each one corresponding to an interval of consecutive labels. For example, starting from vertex 0000 and following edge (0000-1000), the 2^3 vertices from 1000 to 1111 can be reached with a shortest path of at most 4 steps.

Let H be the set of 2^k vertices of a hypercube with standard labelling. We have:

Property 1 *For any vertex $X \in H$ there exists exactly one 1-*IRS*. This is based on a partition of $H - \{X\}$ in k disjoint subsets of $2^{k-1}, 2^{k-2}, \ldots, 2^0$ elements, each one corresponding to an interval associated to an edge leaving from X.*

Proof Build a 1-*IRS* in the following way. Starting from vertex X, with binary label $x_{k-1} \ldots x_0$, choose $[\overline{x}_{k-1}0 \ldots 0, \overline{x}_{k-1}1 \ldots 1]$ as first interval, and assign it to the edge leading to vertex $\overline{x}_{k-1}x_{k-2} \ldots x_0$. Choose $[x_{k-1}\overline{x}_{k-2}0 \ldots 0, x_{k-1}\overline{x}_{k-2}1 \ldots 1]$ as second interval, and assign it to edge leading to vertex $x_{k-1}\overline{x}_{k-2} \ldots x_0$. Continue until the interval $[x_{k-1}x_{k-2} \ldots \overline{x}_0, x_{k-1}x_{k-2} \ldots \overline{x}_0]$ containing a single vertex is formed.

We now show that the above is the only 1-*IRS* possible. Consider an arbitrary vertex Y with label $y_{k-1} \ldots y_0$ to be reached from X, and assume that $Y > X$, meaning that $y_{k-1} \ldots y_0 > x_{k-1} \ldots x_0$. (The case $Y < X$ can be symmetrically treated, and is omitted). Let $y_i = 0, x_i = 1$ for a given i. In any 1-*IRS*, Y cannot be included in the interval of the edge $(X - Z)$, with Z labelled $x_{k-1} \ldots x_{i+1}0x_{i-1} \ldots x_0$, because $Z < X < Y$, and that interval contains Z and does not contain X. Then Y must be in the interval of an edge $(X - W)$ such that $y_j = 1, x_j = 0$ and W is labelled $x_{k-1} \ldots x_{j+1}1x_{j-1} \ldots x_0$, for some j. We now prove that j is the most significant position for which $y_j = 1, x_j = 0$. By contradiction, let $r > j$ be such a position, hence $y_r = 1, y_j = 1, x_r = 0, x_j = 0$. We have $X < W < Y$, then the interval I_{XW} of edge $(X - W)$ must contain all the labels from W to Y, that is from $x_{k-1} \ldots x_{r+1}0x_{r-1} \ldots x_{j+1}1x_{j-1} \ldots x_0$ to $y_{k-1} \ldots y_{r+1}1y_{r-1} \ldots y_{j+1}1y_{j-1} \ldots y_0$ (note that $X < Y$ implies $x_{k-1} \ldots x_{r+1} = y_{k-1} \ldots y_{r+1}$). Consider now the vertex V labelled $y_{k-1} \ldots y_{r+1}100 \ldots 0$. We have $W < V < Y$, then the label of V is in I_{XW}. Therefore V must be reached from X through edge $(X - W)$ with a shortest path, that is impossible because the j-th bit has value 0 in the label of X, 1 in the label of W, and 0 in the label of V, thereby changing twice along the path. In conclusion we have proved that all vertices $Y > X$ such that $x_j = 0$, $y_j = 1$, and j is the most significant position where this occurs, must be reached from X along the edge correspoding to changing x_j from 0 to 1.

Similarly we can prove that all vertices $Y < X$ such that $x_j = 1$, $y_j = 0$, and j is the most significant position where this occurs, must be reached from X along the edge correspoding to changing x_j from 1 to 0. This is exactly the way the intervals were constructed at the beginning of the proof. □

Referring to the masked interval routing scheme we have:

Property 2 *For any vertex $X \in H$, there exist $k!$ different 1-*MIRS*'s based on partitions of $H - \{X\}$ in k disjoint subsets of $2^{k-1}, 2^{k-2}, \ldots, 2^0$ elements, each one corresponding to an interval associated to an edge leaving from X.*

Proof Consider the label $x_{k-1} \ldots x_0$ of X, and choose an arbitrary bit x_{i_1}, $0 \leq i_1 \leq k-1$. The vertices labelled from $0 \ldots 0\overline{x}_{i_1}0 \ldots 0$ to $1 \ldots 1\overline{x}_{i_1}1 \ldots 1$ form an interval of size 2^{k-1} with mask $0 \ldots 010 \ldots 0$, where the 1 is in position i_1. This interval contains the label $x_{k-1} \ldots x_{i_1+1}\overline{x}_{i_1}x_{i_1-1} \ldots x_0$ of a vertex W adjacent to X, and can be assigned to the edge $(X - W)$. That is, there are k ways of selecting an interval of 2^{k-1} labels, each one to be associated to a different edge leaving from X.

For each choice of i_1, choose a second bit x_{i_2} (in one of the remaining $k-1$ positions), to build a masked interval of size 2^{k-2} from $0\ldots 0\overline{x}_{i_2}0\ldots 0x_{i_1}0\ldots 0$ to $1\ldots 1\overline{x}_{i_2}1\ldots 1x_{i_1}1\ldots 1$, with mask $0\ldots 010\ldots 010\ldots 0$, with the 1's in positions i_2 and i_1. This interval contains the label $x_{k-1}\ldots x_{i_2+1}\overline{x}_{i_2}x_{i_2-1}\ldots x_{i_1}\ldots x_0$ of a vertex Z adjacent to X, and can be assigned to the edge $(X-Z)$.

Proceed similarly on all other bit positions $i_3,\ldots,i_k$ to obtain the stated result. □

Any partition of $H-\{X\}$ built as in the proof of property 2 will be called *canonical*. For X, each selected position i_j, $1\le j\le k$, is the *pivot* for the subset of 2^{k-j} elements (or for the corresponding interval). Note that each subset of a canonical partition consists of the 2^{k-j} vertices of a subcube whose bit labels have constant values in positions i_1 to i_j. The partition of $H-\{0000\}$ marked with engrossed edges in figure 4, and augmented with a subset containing the only vertex 1000, is a canonical partition. Here position 3 is the pivot for the subcube $\{0010,0011,1010,1011\}$.

The variety of ways with which we can express masked intervals in hypercubes is interesting for fault tolerance reasons. Consider a network organized as an hypercube and suppose that one of the vertices falls. Referring to figure 4, suppose that a message has to be routed from vertex 0000 to vertex 1100, and vertex 1000 falls. According to the 1-*IRS* built as in property 1, the shortest path to be followed between the two vertices is 0000-1000-1100, that is now interrupted. Note that if a vertex X falls, its label can be kept in an interval for the sake of maintaining the integrity of such an interval (simply X will not be searched for, as for the dummy vertices introduced in the previous section). However, X cannot be inserted in a route. By property 1 we cannot rearrange the intervals of the edges leaving vertex 0000 to cope with this new situation. That is, no 1-*IRS* can be built in case of one vertex failure. However, an optimal rearrangement can be done in 1-*MIRS*, as a consequence of property 2. We consider an example first.

In the hypercube of figure 4, upon failure of vertex 1000, the intervals on the edges leaving from vertex 0000 can be reconfigured by assigning [0100,1111,0100] to edge (0000-0100); [0010,1011,0110] to edge (0000-0010); and [0001,1001,0111] to edge (0000-0001). This new partition is marked with engrossed edges in the figure. The interval on edge (0000-1000) is discarded because such an edge is unusable. The additional failure of another vertex, say 0011, would affect the intervals of the vertices adjacent to it, but not the ones of 0000. The general reconfiguration strategy is presented in the proof of the next theorem.

In the set of vertices H of the hypercube we shall say that two vertices X,Y are at distance $d(X,Y)=h$ if their standard binary labels differ exactly in h positions. Clearly $d(X,Y)$ is the length of a shortest path between X and Y. We then pose:

Definition 1 *A subset of vertices $F\subset H$ is* failure harmless *(shortly* harmless*) if each vertex $X\in H-F$ is at distance one from at most one vertex $Y\in F$. A canonical partition of $H-\{X\}$ is* consistent *with F if $\{Y\}$ is one of the elements of the partition (in fact, the one containing a single element). If Y does not exist, any canonical partition of $H-\{X\}$ is consistent with F.*

In figure 4, the subset $F=\{0011,1000,1100\}$ is harmless. For $X=0000$ the canonical partition of $H-\{X\}$ marked with engrossed edges and augmented with the subset $\{1000\}$, is consistent with F (we have $Y=1000$).

Before procceding we must clarify the definition of $MIRS$ in the case of faulty vertices, because shortest paths may become longer. For example the concurrent failure of 1000 and 0100 cuts the two paths of length 2 between 0000 and 1100, and the shortest paths between these two vertices have now length 4. We say that a 1-*MIRS* is *strong* if, for any two non faulty vertices X, Z the routing specifies a shortest path of length $d(X,Z)$ (composed of non faulty vertices). We now show that failure of the vertices of a harmless subset does not prevent the construction of a strong 1-*MIRS*.

Theorem 2 *A hypercube admits a strong 1-*MIRS *under concurrent failure of vertices, if all these vertices belong to a harmless subset $F \subset H$.*
Proof For each vertex $X \in H - F$ build the intervals of the leaving edges according to a partition of $H - \{X\}$ consistent with F. We prove that this is a strong 1-*MIRS* by induction on k, $|H| = 2^k$.

The basis $k = 1$, $k = 2$ is trivial. Inductively assume that the theorem holds for all the subcubes of $\leq 2^{k-1}$ vertices. Consider the whole hypercube of 2^k vertices. For any $X \in H - F$, let P_X be a partition of $H - \{X\}$ consistent with F. We first prove that the restriction of P_X to any subcube C of 2^{k-1} vertices containing X is consistent with the restriction F_C of F to C.

All the vertices of C have a constant bit in a position h (this bit is equal to x_h, the h-th bit of X). Among all the pivots of P_X, let i_c be the one equal to h. All the subsets of P_X with pivot i_j, $j < c$, have half of the vertices in C, namely those with the h-th bit equal to x_h. The subset of P_X with pivot h is not contained in C. All the subsets with pivot i_j, $j > c$, are totally contained in C. Then, P_X is clearly canonical in C and consistent with F_C.

Divide now the hypercube in two subcubes C_1, C_2 of 2^{k-1} vertices having constant bit in position i_1, namely, this bit is equal to x_{i_1} in C_1, and to $\overline{x}_{i_1}$ in C_2 (then C_1 contains X). The restriction of P_X to C_1 corresponds to a strong 1-*MIRS* for X in C_1, and is obviously part of a strong 1-*MIRS* in the whole hypercube. The interval $[0\ldots0\overline{x}_{i_1}0\ldots0, 1\ldots1\overline{x}_{i_1}1\ldots1, 0\ldots010\ldots0]$, associated to the edge $(X - Z)$, $Z \in C_2$, encompasses all the vertices of C_2. We have to prove that this is consistent with a strong 1-*MIRS* for X in the whole hypercube. In fact, the restriction to C_2 of the consistent partition chosen for Z corresponds to a strong 1-*MIRS* in C_2 by induction. Therefore, the routing from X to any vertex W of C_2 can be realized with a path of length $d(X,W)$ passing through Z and lying in C_2. □

A reconfiguration procedure for the 1-*MIRS* in the presence of faults can be immediately derived from the proof of theorem 2. 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 neighbour Y (if any), and there is a global guarantee that the faults belong to a harmless set of vertices.

5 Concluding Remarks

The new technique of masked interval routing has been introduced here, with the aim of compressing the routing tables stored in a network. The investigation has taken two directions. First we have shown that interval information may be drastically reduced over the standard scheme, in networks difficult to handle. This is the case of globe graphs of $O(n^2)$ vertices, where the number of intervals per edge goes down from $\Omega(n)$ to $O(log\ n)$. We have then extended the technique to globe graphs of arbitrary dimensions. Second we have shown that masked intervals may be advantageously used in fault-tolerant networks, proving in particular that 1-interval routing is still possible in hypercubes with a proper subset of faulty vertices.

This is an introductory work. Further research is needed in both the directions taken here. The class of graphs that may benefit of the new routing scheme must be fully investigated, with the addition of other significant families to the one of globe graphs. In this context, the straightforward algorithm proposed here for the search of a destination label in a set of masked intervals should be possibly improved.

The reconfiguration problem of faulty networks must be also reconsidered on different topologies. The definition of a "harmless" subset of faulty vertices should be generalized, together with an evaluation of the maximal cardinality of such a subset. An interesting problem here is the one of designing a proper agreement algorithm run by the non faulty processors, to let them decide that the set of faulty vertices is harmless without need of an external control. Efficient reconfiguration in the presence of faulty edges should also be studied.

References

[1] E. M. Bakker, R.B. Tan, J. van Leeuwen. Linear Interval Routing Schemes. *Algorithms Review* 2 (1991) 45-61.

[2] P. Fraigniaud, C. Gavoille. Interval Routing Schemes. *École Normale Supérieure, Lyon*, Res. Rep. 94-04, 1994.

[3] INMOS *The T9000 Transputer Overview Manual*, 1991.

[4] R. Kralovic, P. Ruzica, D. Stefancovic. The Complexity of Shortest Path and Dilation Bounded interval Routing. *Resource Material in Compact Routing: Research Meeting on Structural Information and Communication Complexity*, Siena, June 1997.

[5] E. Kranakis, D. Krizanc, S.S.Ravi. On Multi-Label Linear Interval Routing Schemes . In: *Distributed Algorithms.* LNCS 790, Springer Verlag 1993, 338-349.

[6] J. van Leeuwen and R. Tan. Interval Routing *The Computer Journal*, 30 (1987) 298-307.

[7] M.D. May, P.W. Thompson, P.H. Welch. *Networks, Routers and Transputers.* INMOS, Tech. Rep. 1993.

[8] M. Santoro, R. Khatib. Labeling and Implicit Routing in Networks. *The Computer Journal*, 28 (1985) 5-8.