Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp232-238)

# Efficient Routing Algorithms on Optoelectronic Networks

**J. Al-SADI**
Computer Science Department
Arab Open University
AMMAN - Jordan

**AHMAD M. AWWAD**
Computer Science Department
Zarka Private University
ZARKA - Jordan

*Abstract:* This paper proposes an efficient fault tolerant routing algorithm for the class of optoelectronic networks. Extended Optoelectronic Cube interconnection network; EOC for short; is introduced as a new member of this family. EOC is constructed from multiplying the hypercube as a factor network by it self. In this paper we utilize the features of optoelectronic networks which use both of electronic and optical networks. The EOC utilizes the good features of the Hypercube network due to its attractive features in terms of connectivity, semantic, low diameter, and multiple alternative paths in addition to the extended Optoelectronic capabilities of using both electronic and optical technologies. In this new proposed algorithm, each node A starts by computing the first level unsafety set, $S_1^A$, composed of the set of unreachable direct neighbours. It then performs m-1 exchanges with its neighbours to determine the k-level unsafety sets $S_k^A$ for all 1 ≤ k ≤ m, where m is an adjustable parameter between 1 and 2n. The k-level unsafety set at node A represents the set of all faulty nodes at Hamming distance k from A which either faulty or unreachable from A due to faulty nodes or links.

*Keywords:* Parallel and distributed systems; Hypercube network; Optoelectronic Networks; Fault-Tolerant Routing.

## 1. Introduction

Recently, there has been an increasing interest in the Optoelectronic network. It is also called Optical Transpose Interconnection Networks "OTIS-networks" [4, 15, 19]. Marsden *et al* were the first to propose the OTIS-networks [9]. Extensive modeling results for the OTIS have been reported in [7]. The achievable terabit throughput at a reasonable cost makes the Optoelectronic a strong competitor to the electronic alternatives [8, 9]. In the optoelectronic network, shorter (intra-chip) communication is realized by electronic interconnects while longer (inter-chip) communication is realized by free space interconnects. Using cube as a factor network will yield the Optoelectronic-Cube in denoting this network.

Optoelectronic-Cube is basically constructed by "multiplying" a cube topology by itself. The set of vertices is equal to the Cartesian product on the set of vertices in the factor cube network. The set of edges E in the Optoelectronic-Cube consists of two subsets, one is from the factor cube, called cube-type edges, and the other subset contains the transpose edges. The optoelectronic approach suggests implementing cube-type edges by electronic links since they involve intra-chip short links and implementing transpose edges by free space optics. Throughout this paper the terms "electronic move" and the "Optoelectronic move" (or "optical move") will be used to refer to data transmission based on electronic and optical technologies, respectively.

Although the Optoelectronic-Cube network has many attractive topological properties but it suffers from having limited optical links between the different groups. When source and destination nodes are in two different groups, the fact that only one optical link connects two distinguished groups directly create a congestion problem to most of the shortest paths that have to pass through this particular optical link. Furthermore, alternative paths are too long compared to the only one short path because they have to be routed via a third group which require passing via two optical links in addition to the electronic moves in each group to reach the destination.

To overcome the weakness of the topological properties of the Optoelectronic-Cube, the paper [1] proposed a new interconnection network topology called an Extended Optoelectronic-Cube; EOC for short; based on the "Optoelectronic-Cube" network. The EOC network is semantic, regular, and has a small diameter. The reader may refer to [1] for further details on the attractive topological properties of the EOC interconnection networks.

Our motivation in this paper is to support this new attractive network with a new fault tolerant routing algorithm capable of routing messages from any source node to any destination node in presence of reasonable number of faults in the network.

## 2. Related Work

The efficient inter-processor communication is the key to good system performance. The routing algorithm has great impact on network performance, as it is responsible for selecting a network path between two nodes involved in a one-to-one communication. Routing in fault-tolerant, fault-free cube networks and its variants have been extensively studied in the past (e. g. see [3, 5, 6, 11, 13, 17, 18]). Hardly may you find any fault-free or even fault-tolerant routing algorithm for the Optoelectronic-cube [2]. Since the EOC network is new, no fault tolerant routing algorithm has been written for this network. As the network size scales up the probability of processor and link failure also increases. It is therefore essential to design fault-tolerant routing algorithms that allow to route messages between non-faulty nodes in the presence of faulty components (links and nodes).

The new proposed fault-tolerant routing algorithm for the EOC based on the set of unsafety vectors utilizing the attractive topological properties of EOC network [1] to achieve an efficient fault-tolerant routing. Each node in EOC A starts by determining the set of unreachable immediate neighbors due to faulty nodes and links. This set is referred to as the first-level unsafety set at node A and is denoted $S_1^A$. Then, each node A performs an m-1 exchanges with its immediate neighbors to determine the k-level unsafety set $S_k^A$ for all $1 \leq k \leq m$, where m is an adjustable parameter between 1 and 2n for an n dimensional EOC where 2n is the longest optimal path in the network; the diameter. The k-level unsafety set $S_k^A$ represents the set of all nodes at distance k from A which are faulty or unreachable from node A due to faulty links which causing a network partitioning. Equipped with these unsafety sets, each node calculates numeric unsafety vectors and uses them to achieve efficient fault-tolerant routing algorithm. The larger the value of m is the better the routing decisions are, but at the expense of more computation and communication overhead.

## 3. Notations and Definitions

The n-dimensional undirected graph binary n-cube $Q_n$ is one of the well known networks which have been used in real life systems [10, 12, 15, 16]. The undirected graph n-cube with 2n vertices, representing nodes, which are labeled by the 2n binary strings of length n. Two nodes are joined by an edge if, and only if, their labels differ in exactly one bit position. The label of node A is written an an-1…a1, where ai ∈ {0, 1} is the ith bit (or bit at ith dimension) [14].

From the above definition the neighbor of a node A along the ith dimension is denoted A(i). A faulty n-cube contains faulty nodes and/ or links.

The Extended Optoelectronic-Cube is obtained by "multiplying" a cube topology by itself. The vertex set is equal to the Cartesian product on the vertex set in the factor cube network. The edge set consists of edges from the factor network and new edges called the transpose edges. The formal definition of the Extended Optoelectronic-Cube is given below.

***Definition 1:*** Let *n-cube* = $(V_0, E_0)$ be an undirected graph representing a cube network where $n$ is the cube degree. The Extended Optoelectronic-Cube = $(V, E)$ network is represented by an undirected graph obtained from *n-cube* as follows $V = \{\langle x, y\rangle \mid x, y \in V_0\}$ and $E = \{(\langle x, y\rangle, \langle x, z\rangle) \mid$ if $(y, z) \in E_0\} \cup \{(\langle x, y\rangle, \langle y, x\rangle) \mid x, y \in V_0\} \cup \{(\langle x, x\rangle, \langle y, y\rangle) \mid x, y \in V_0 \cap x \text{ is an opposite of } y\}$

***Definition 2:*** Let $x$, $y$ be group addresses of two nodes in an EOC labeled as series of bits $\langle x_n \ldots x_2 x_1\rangle$, $\langle y_n \ldots y_2 y_1\rangle$ consequently where each bit is either 0 or 1. $x$ is called *an opposite of y* if and only if they are differ only in the first bit.

In the EOC the address of a node u = ⟨x, y⟩ from V is composed of two components. Fig. 1 shows a 16 processor EOC, the notation ⟨g, p⟩ is used to refer to the group and processor addresses, respectively. Two nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ are connected if one of the following cases occurs:

1- If $g_1 = g_2$ and $(p_1, p_2) \in E_0$ (such that $E_0$ is the set of edges in $n$-cube network), in this case the two nodes are connected by an electronic edge.

2- If $g_1 = p_2$ and $p_1 = g_2$, in this case the two nodes are connected by a transpose edge.

3- If $g_1 = p_1$ and $g_2 = p_2$ and $g_1$ is an opposite of $g_2$, in this case the two nodes are connected by a transpose edge too.

***Definition 3:*** The Topological properties of the EOC are defined as follows:

*1- **Size***: If the cube factor network of size $N$, then the size of the EOC is $N^2$.

*2- **Degree***: Let $\deg_{G_0}$ be the degree of the $n$-cube network and let $\langle g, p\rangle$ be any node in an EOC. Then the degree of the EOC is as follows:

$$\deg_{OTIS-CUBE}(g, p) = \deg_{G_0}(p) + 1 = n + 1$$

*3- **Number of Links***: The number of links in EOC = $(n+1)2^{2n}/2$ where $n$ is the dimension of the cube factor network.

The distance in the EOC is defined as the shortest path between any two processors, $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$, and involves one of the following forms:

*i-* When $g_1 = g_2$ then the path involves only electronic moves from source node to destination node.

*ii-* When $g_1 \neq g_2$ and if the number of optical moves is an even number of moves, then the paths can be compressed into a shorter path of the form: $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, p_2\rangle \xrightarrow{O}$

$\langle p_2, g_1\rangle \xrightarrow{E} \langle p_2, g_2\rangle \xrightarrow{O} \langle g_2, p_2\rangle$ where the symbols $O$ and $E$ stand for optical and electronic moves respectively.

*iii-* When $g_1 \neq g_2$, and the path involves an odd number of Optoelectronic moves. In this case the paths can be compressed into a shorter path of the form: $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, g_2\rangle \xrightarrow{O} \langle g_2, g_1\rangle \xrightarrow{E} \langle g_2, p_2\rangle$.

*iv-* When $g_1$ is opposite of $g_2$, and if the number of optical moves is an even number of moves, then the paths can be compressed into a shorter path of the form: $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, p_2\rangle \xrightarrow{O} \langle p_2, g_1\rangle \xrightarrow{E} \langle p_2, g_2\rangle \xrightarrow{O} \langle g_2, p_2\rangle$ where the symbols $O$ and $E$ stand for optical and electronic moves respectively.

*v-* When $p_1$ is opposite of $p_2$, and the path involves an odd number of OTIS moves. In this case the paths can be compressed into a shorter path of the form: $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, g_2\rangle \xrightarrow{O} \langle g_2, g_1\rangle \xrightarrow{E} \langle g_2, p_2\rangle$.

*4- **Length***: Let $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ be two different nodes in the EOC. To transmit data originated in the source node $\langle g_1, p_1\rangle$ to the destination node $\langle g_2, p_2\rangle$ it follows one of the five possible paths shown above *i*, *ii*, …, *v*. The length of the shortest path between the two nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ is:

$$Length = \begin{cases} d(p_1, p_2) & if\ g_1 = g_2 \\ \min(d(p_1,g_1)+d(g_{1\text{Opposite}},p_2)+1, d(p_1,g_1)+d(p_{1\text{Opposite}},g_2)+2) & if\ g_1 = g_{2\text{Opposite}} \text{ or } p_1 = p_{2\text{Opposite}} \\ \min(d(p_1,g_2)+d(p_2,g_1)+1, d(p_1,p_2)+d(g_1,g_2)+2) & if\ g_1 \neq g_2 \end{cases}$$

where $d(p_1, p_2)$ is the length of the shortest path between any two processors $\langle g_1, p_1\rangle$ and $\langle g_1, p_2\rangle$.

*5- **Diameter***: Let $n$ be the diameter of the $n$-cube network, then the diameter of the EOC is $2n$.

# 4. The Unsafety Vectors Fault-Tolerant Routing Algorithm

In this section, we introduce the adapted fault-tolerant routing algorithm, based on the concept of unsafety sets (defined below). Before presenting the new algorithm, we first discuss how a node in the EOC calculates its unsafety sets.

The calculation of the unsafety sets is as follows:

*Definition 4:* The number of direct neighbors $np$ of a node $A$, $<g_A, p_A>$, is equal to $np=n+1$that consist of $n$ neighbors via electronic move and one neighbor via optical move.

*Definition 5:* The *first-level unsafety set* $S_1^A$ of a node $A$ is defined as

$$S_1^A = \bigcup_{1 \le i \le np} f_A^i \text{, where } f_A^i \text{ is}$$

given by

$$f_A^i = \begin{cases} \{A^{(i)}\} & if\ A^{(i)}\ is\ faulty \\ \phi & otherwise \end{cases}$$

It should be clear that an *isolated node A* is associated with first-level unsafety set containing $np$ addresses of faulty nodes, i. e., $|S_1^A| = np$. If for some node $A$, $|S_1^A| = np-1$ then node $A$ is called a *dead-end node*.

Each node uses the unsafety set to determine the faulty set $F_A$, which comprises those nodes which are either faulty or unreachable from $A$ due to faulty nodes or links. This is achieved by performing $m$-1 exchanges with the reachable neighbors. After determining $F_A$, node $A$ calculates $m$ unsafety sets denoted $S_1^A$, $S_2^A$,…, $S_m^A$ (defined below), where $m$ is an adjustable parameter between 1 and $2n$.

*Definition 5:* The $k$-level unsafety set $S_k^A, 1 \le k \le m$, for node $A$ is given by

$$S_k^A = \{B \in F_A \mid d(A,B) = k\}$$

The $k$-level unsafety set $S_k^A$ represents node $A$'s view of the set of nodes at distance $k$ from $A$ which are faulty or unreachable from $A$ due to faulty nodes and links. Notice that if the network is disconnected due to faulty nodes and links, $A$'s view about unreachable nodes may not be accurate. In this case massage of Unreachability may occur. Fig. 2 gives an outline of the *Find_Unsafety_Sets* algorithm that node $A$ uses it to determine it's faulty and unsafety sets.

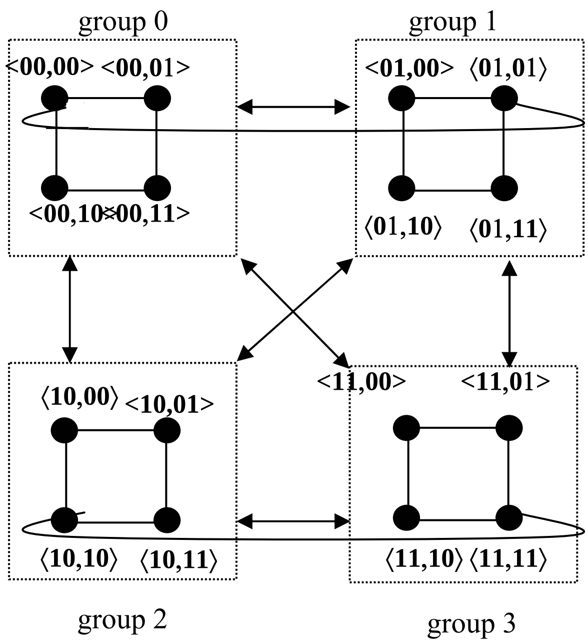


**Fig. 1. 16-processor EOC**

With respect to a given destination node, $D$, in a cube network a neighbor $A^{(i)}$ of node $A$ is called a *preferred neighbor* for the routing from $A$ to $D$ if the $i^{th}$ bit of $A \oplus D$ is 1. We say in this case that $i$ is a *preferred dimension*. Neighbors other than preferred neighbors are called *spare neighbors*. Routing through a spare neighbor increases the routing distance by two over the minimum distance. In general, a preferred neighbor is one step closed to the destination while a spare neighbor increases the routing distance two or more steps over the minimum distance depending of the type of the next move (electronic or optical). An optimal path can be obtained by routing through all preferred dimensions in some order. A node $T$ is called an ($A$, $D$)-*preferred transit node* if any preferred dimension for the routing from $A$ to $T$ is also a preferred dimension for the routing from $A$ to $D$.

*Algorithm Find_Unsafety_Sets* ($<g_A, p_A>$*: node*)
*/* called by node A to determine its faulty set* $F_A$ **/*

$S_1^A$ = *set of faulty or unreachable immediate neighbors;*
$F_A = S_1^A$;
*for k := 2 to m do*
{
*for i :=1 to n do*
*if* $P_A^{(i)} \notin F_A$ *then*
{
*send* $F_A$ *to* $P_A^{(i)}$;
*receive* $F_A^{(i)}$ *from* $P_A^{(i)}$;
$F_A = F_A \cup F_A^{(i)}$;
}
*if* $p_A \neq g_A$
{
*send* $F_A$ *to* $<g_A, p_A>$;
*receive* $F_{<g_A, p_A>}$ *from* $<g_A, p_A>$;
$F_A = F_A \cup F_{<g_A, p_A>}$;
}
*else* {
*send* $F_A$ *to* $<g_{A\ opposite}, p_{A\ opposite}>$;
*receive* $F_{<g_{A\ opp}, p_{A\ opp}>}$ *from* $<g_{A\ opp}, p_{A\ opp}>$;
$F_A = F_A \cup F_{<g_{A\ opp}, p_{A\ opp}>}$;
}
}
*for k := 1 to m do*
$S_k^A = \{<g_B, p_B> \in F_A \mid dist(<g_A, p_A>, <g_B, p_B>) = k\}$
*End*

**Fig. 2. The find_unsafety_sets algorithm that determines the faulty set for node *A*.**

*Example 1:* Consider a two-dimensional EOC with four faulty nodes (faulty nodes are represented as black nodes), as shown in Fig. 3. Table 1 shows the corresponding first-level unsafety set, $S_1^A$, associated with each node $A$. The Find_Unsafety_Sets algorithm calculates the sets $S_k^A$ for all $1 \le k \le m$ after calculating $F_A$. To achieve this, ($m$-1) exchanges of fault information are performed among neighboring nodes.

Let $m = 2n$ and for the sake of specific illustrations let us compute the unsafety sets associated with node $A$ = 0000. First, the node assigns the addresses of its immediate faulty neighbors to its faulty set $F_A$. Then each node performs $2n$ exchanges of the new elements of its faulty set $F_A$ with the immediate non-faulty neighbors. After determining $F_A$, node $A$ calculates $m$ unsafety sets denoted, $S_1^A$, $S_2^A$, …, $S_m^A$ according to the distance between node $A$ and each element of $F_A$. So, the faulty set for node $A$ in our example, given in decimal representation, $F_A$= {1, 10, 12, 15}, and the unsafety sets are $S_1^A$ = {1}, $S_2^A$ = {}, $S_3^A$ = {10, 12}, $S_4^A$ = {}, and $S_5^A$ = {15}.

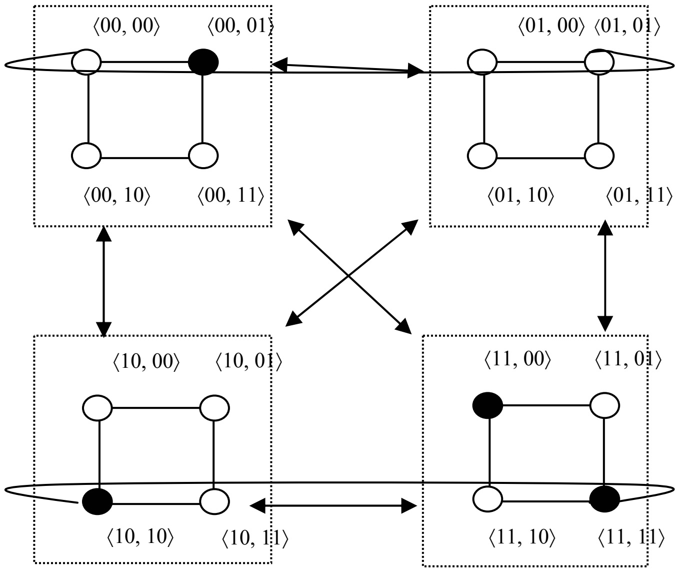


**Fig. 3. A 2-dimensional EOC with four faulty nodes.**

**Table 1. The unsafety sets of nodes in EOC ($n$ = 2) with 4 faulty nodes.**

| Node | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| $S_1^A$ | {1} | Faulty | {} | {1} | {1} | { } | { } | { } |

| Node | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|
| $S_1^A$ | {10} | { } | Faulty | {10} | Faulty | {12,15} | {12,15} | Faulty |

# 4. The Unsafety Vectors Routing Algorithm

For a given source-destination pair of nodes ($<g_A, p_A>$, $<g_D, p_D>$), we define the *(A, D)-unsafety vector* $U_k^{A,D} = (u_1^{A,D}, \ldots, u_k^{A,D}, \ldots, u_m^{A,D})$ where its $k^{th}$

element is given by $u_k^{A,D} = |\{T \in S_k^A$, such that $T$ is an $(A, D)$-preferred transit node$\}|$.

In other words, $u_k^{A,D}$ is the number of faulty or unreachable $(A, D)$-preferred transit nodes at distance $k$ from $<g_A, p_A>$. $u_k^{A,D}$ can be viewed as a measure of routing unsafety at distance $k$ from $<g_A, p_A>$, hence the name *unsafety vectors* for $U^{A,D}$. We also define an ordering relation '<' for numeric vectors as follows.

*Definition 5:* For any two numeric vectors $U = (u_1, u_2,\ldots, u_m)$ and $V = (v_1, v_2, \ldots, v_m)$, $U < V$ iff $\exists\, i$, $1 \le i \le m$, such that $u_i < v_i$, and $u_j = v_j$ for all $j < i$.

Fig. 4 shows the *Unsafety_Vectors* algorithm that each node in the network applies to route a message towards its destination node $<g_D, p_D>$.

*Example 2:* Consider the cube depicted in Fig. 3 where the source node $A$ = 0100, the destination node $D$ = 1011, and let $m$ = 1. According to the unsafety vectors algorithm, the source node $A$ will route message to a preferred neighbor associated with the least number of preferred faulty nodes in its unsafety sets, which is node 0110. By performing the same operations the message will be routed through an optical move to node 1001 then finally to its destination 1011.

*Theorem 1:* Let $A^{(i)}$ and $A^{(j)}$ be two non faulty (A, D)-preferred neighbors of A. If all preferred neighbors of $A^{(j)}$ are faulty and at least one *preferred* neighbor of $A^{(i)}$ is non faulty then the unsafety vectors algorithm does not route messages of destination D via $A^{(j)}$.

*Proof:* Since $u_1^{A^{(i)},D} < u_1^{A^{(j)},D}$ then $U^{A^{(i)},D} < U^{A^{(j)},D}$. Therefore, $U^{A^{(j)},D}$ is not the minimal such vector (for the preferred neighbors). □

*Algorithm Unsafety_Vectors (M: message; $<g_c, p_c>$, $<g_d, p_d>$: node)*
*/* called by current node $<g_c, p_c>$ to route the message M to its destination node $<g_d, p_d>$ */*
*if $<g_c, p_c>$ is source node then*
*M.Route_distance = 0*

*if Route_distance <= $dist(p_c, p_d)+dist(g_c, g_d)+(2n)$* No_FaultyNodes then*
*{*
*M.Route_distance:= M.Route_distance+1*
*if $(g_c = g_d)$and $(p_c = p_d)$ then*
*exit; /* destination reached */*
*if $g_c = g_d$ then*
*route($<g_c, p_c>$,$<g_d, p_d>$)*
*/* curr& dest.at the same group */*
*if ($g_c$=$g_{d\ opposite}$) and ($dist(p_c,g_c)+dist(g_d,p_d)+1 < dist(p_c,g_d) + dist(g_c,p_d)+1$) and the optical move ($g_c,g_c \rightarrow g_d,g_d$) is not faulty then*
*{*
*if $p_c$=$g_c$ then move m to $<g_d,g_d>$*
*else route($<g_c, p_c>$,$<g_c, g_c>$)*
*else if ($p_{c\ opposite}$=$p_d$)and($dist(g_c,p_c)+ dist(p_{c\ opposite},g_d)+2 < dist(p_c,g_d)+dist(g_c,p_d)+1$) then*
*{ if ($p_c$=$g_c$) and the two optical moves ($g_c,g_c \rightarrow p_d,p_d$, $p_d,g_d \rightarrow g_d,p_d$ ) are not faulty then move m to $<p_d,p_d>$*
*if ($p_c \ne g_c$) and the two optical moves ($g_c,g_d \rightarrow g_d,g_c$, $p_c,p_c \rightarrow p_d,p_d$ ) are not faulty then move m to $<g_d,p_d>$*
*}*
*elseif ($dist(p_c, p_d)+dist(g_c, g_d)+2) < dist(p_c,g_d)+ dist(g_c,p_d) +1$) and the two optical moves ($g_cp_d \rightarrow p_dg_c$, $p_dg_d \rightarrow g_dp_d$) are not faulty then*
*{ if $p_c = p_d$ then*
*move m to $< p_c, g_c>$*
*else route($<g_c, p_c>$,$< g_c, p_d>$)*
*}*
*else if the optical move ($g_cg_d \rightarrow g_dg_c$) is not faulty then {*
*if $p_c = g_d$ then*
*move m to $< p_c, g_d>$*
*else route($<g_c, p_c>$,$<g_c, g_d>$)*
*}*
*else if $g_c \ne p_c$ and the node$<p_c, g_c>$ is not faulty then*
*send M to $<p_c, g_c>$ /* disturb the message*/*
*else looping*
*}*

*End.*
*Function route($<g_c, p_c>$,$<g_d, p_d>$:node)*
*{*
*if $\exists$ a preferred non-faulty neighbor $A^{(i)}$ with least ($A^{(i)}$, D)-unsafety vector $U^{A(i),D}$ and $A^{(i)}$ is not a dead-end node then send M to $A^{(i)}$*
*else if $\exists$ a spare non-faulty neighbor $A^{(j)}$ with least ($A^{(j)}$, D)-unsafety vector $U^{A(j),D}$ and $A^{(j)}$ is not dead-end then*
*send M to $A^{(j)}$*
*else if $g_c \ne p_c$ and the node$<p_c, g_c>$ is not faulty then*
*send M to $<p_c, g_c>$ /* disturb the message*/*
*else failure /* destination unreachable */ }*

**Fig. 4. A description of the proposed unsafety vectors routing algorithm.**

Fig. 4 shows the fault-tolerant *routing* algorithm that each node $<g_c,p_c>$.in the network applies to route a message towards its destination node $<g_d,p_d>$. The algorithm checks first whether the source and the destination nodes are in the same group or not. If both nodes are in the same group then the cube routing rules are applied by selecting a preferred neighbor to guarantee an optimal routing toward the destination. Otherwise, the algorithm selects a move that leads to make an optical move to reach the destination's group, then, to reach the destination node.

## 5. Conclusion

This paper has proposed a new fault-tolerant routing based on the concept of unsafety vectors. As a first step in this algorithm, each node $A$ determines its view of the faulty set $F_A$ of nodes, which are either faulty or unreachable from $A$. This is achieved by performing at most $2n$ exchanges with the reachable neighbours. After determining $F_A$, node $A$ calculates $m$ unsafety sets denoted , $S_1^A$, $S_2^A$,…, $S_m^A$ where $m$ is an adjustable parameter between 1 and $2n$. The $m$-level unsafety set represents the set of all nodes at distance $m$ from $A$ which are faulty or unreachable from $A$ due to faulty links or nodes.

Equipped with these unsafety sets each node calculates unsafety vectors and uses them to achieve fault-tolerant routing in the EOC. The larger the value of $m$ is the better the routing decisions are, but at the expense of more communication overhead. An extension for this work in the future is to implement the proposed routing algorithm for all different network sizes and to conduct a performance analysis through extensive simulation experiments to prove the superiority of the routing.

*References:*


[1] Al-Sadi J., An EOC Interconnection Network, Proceedings of the IADIS International Conference on Applied Computing, Algarve, Portugal, February 22-25, 2005, Vol. II, pp. 167 – 172

[2] AL-Sadi J., Awwad A., A new Fault-Tolerant Routing Algorithm for Optoelectronic-Cube Using Unsafety Vectors, International Journal of Computers and Applications. Vol. 27, No. 4, 2005.

[3] Al-Sadi J., Day K., and Ould-Khaoua M., “Unsafety Vectors: A New Fault-Tolerant Routing for the Binary N-Cube,” *Journal of Systems Architecture*, vol. 47, no. 9, pp. 783-793, 2002,

[4] Awwad A., Al-Ayyoub A., and Ould-Khaoua M., “Efficient Routing Algorithms on the OTIS–Networks,” *in Proceedings of the 3rd International Conference on Information Technology* ( ACIT’2002), The University of Qatar, Doha, pp. 138-144, December 2002.

[5] Gaughan P. T., Yalamanchili S., “Adaptive Routing Protocols for Hypercube Interconnection Networks,” *Computer*, vol. 26, no. 5, pp. 12-24, 1993.

[6] Graham S. and Seidel S., “The Cost of Broadcasting on Star Graphs and K-Ary Hypercubes,” *IEEE Transactions Computers*, vol. 6, pp. 756-759, 1993.

[7] Hendrick W., Kibar O., Marchand P., Fan C., Blerkom D., McCormick F., Cokgor I., Hansen M., and Esener S., “Modeling and Optimisation of the Optical Transpose Interconnection System,” *in Optoelectronic Technology Centre*, Cornell University, September 1995.

[8] Krishnamoorthy A., Marchand P., Kiamilev F., and Esener S., “Grain-size Considerations for Optoelectronic Multistage Interconnection Networks,” *Applied Optics*, vol. 31, no. 26, pp. 5480- 5507, 1992.

[9] Marsden G., Marchand P., Harvey P., and Esener S., “Optical Transpose Interconnection System Architecture,” *Optics Letters*, vol.18, no.13, pp. 1083-1085, 1993.

[10] N-Cube Systems, *N-cube Handbook*, 1986.

[11] Ni M. and McKinley P. K., “A Survey of Routing Techniques in Wormhole Networks,” *Computer*, vol. 26, no.2, pp. 62-76, 1993.

[12] Rattler J., “Concurrent Processing: A new Direction in Scientific Computing,” *in Proceedings of the AFIPS Conference*, pp. 157-166, 1985.

[13] Saad Y. and Schultz M. H., “Data Communication in Hypercubes,” *Technical Report* YALEU/ DCS/ RR-428, Yale University, 1985.

[14] Saad Y. and Schultz M. H., “Topological Properties of Hypercubes,” *IEEE Transactions Computers*, vol. 37, no. 7, pp. 867-872, 1988.

[15] Seitz C. L., “The cosmic cube,” *Communications of ACM*, vol. 28, no.1, pp. 22-23, Jan 1985.

[16] Silicon Graphics, “Origin 200 and Origin 2000,” *Technical Report*, 1996.

[17] Sullivan H., Bashkow T., and Klappholz D., “A Large Scale, Homogeneous, Fully Distributed Parallel Machine,” *in Proceedings of the 4th Annual Symposium Computer Architecture*, pp. 105-124, 1977.

[18] Xue Y. and Nahrstedt K., "Fault Tolerant Routing in Mobile Ad hoc Networks," in Proc. of IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, Louisiana, March, 2003.

[19] Zane F., Marchand P., Paturi R., and Esener S., “Scalable Network Architecture Using the Optical Transpose Interconnection System (OTIS),” *Journal of Parallel and Distributed Computing*, vol. 60, pp. 521-538, 2000