International Journal of Advancements in Electronics & Electrical Engineering– IJAEEE 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 6 : Issue 2 [ISSN : 2319-7498] - Publication Date: 25 June , 2018

Fault Tolerant Routing Algorithm for OTIS-2D-Torus Interconnection Networks

Jehad Ahmed Al-Sadi
Arab Open University

***Abstract*— This paper proposes an enhanced fault-tolerant routing algorithm for the OTIS-2D-tours interconnection network,. Many researchers work have been presented on the Torus related topics, but not often research addressed the OTIS-Torus. In the proposed algorithm, each node A starts by computing the first level unsafety set, S_1^A, composed of the set of unreachable direct neighbours within the OTIS-2D-tours topology. 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+1; diameter of the network. 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. Equipped with these unsafety sets we show how each node calculates numeric unsafety vectors and uses them to achieve efficient fault-tolerant routing for the OTIS-2D-tours.**

Keywords—- Interconnection Networks, OTIS-2D-Torus, Fault-Tolerant Routing Algorithm, Unsafety Vectors.

I. Introduction

The binary *n*-cube is one of the most famous topologies that have been combined within OTIS-network due to its attractive topological properties, e.g. regular structure, low diameter, and ability to exploit communication locality. Several experimental and commercial systems have been built using the factor cube network including the NCUBE-2 [1], Intel iPSC [2], Cosmic Cube [3], and SGI Origin 2000 multiprocessor [4].

The well-organized inter-processor communication is the key to high-quality system performance. Routing algorithms have huge effect on network performance, as it is in charge for selecting a network path between two nodes; source and destination; involved in a one-to-one communication. Routing in fault-tolerant and fault-free *n*-torus (or the torus for short) and its variants has been extensively studied in the past (e.g. see [5,- 8]) and hardly you may find any fault-free or even fault-tolerant routing algorithm in OTIS-Torus. 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). Few fault-free routing strategies have been proposed in the literature for the Torus [5]. Most of these algorithms have assumed that a node knows either only the status of its neighbours (such a model is called *local-information-based*) or the status of all the nodes (*global-information-based*). Local-information-based routing yields sub-optimal routes (if not routing failure) due to the insufficient information upon which the routing decisions are made. Global-information-based routing can achieve optimal or near optimal routing. However, high communication overhead is involved in such algorithms to maintain up-to-date fault information at all network nodes.

The main challenge is to design and build a simple and effective algorithm to represent limited global fault information that allows optimal or near-optimal routing. Up to my knowledge, this is the first attempt to design a limited-global-information-based algorithm for the OTIS-Torus based on the set of unsafety vectors.

The new proposed limited-global-information-based routing algorithm for the OTIS-*2D*-Torus based on the set of unsafety vectors utilizing the attractive topological properties of OTIS-Torus network [9] to achieve an efficient fault-tolerant routing. Each node in OTIS-Torus *P* starts by determining the set of unreachable immediate neighbours due to faulty nodes and links. This set is referred to as the first-level unsafety set at node *P* and is denoted S_1^P. Then, each node *P* performs an *k*-1 exchanges with its immediate neighbours to determine the *l*-level unsafety set S_l^P for all 1 ≤ *l* ≤ *n*, where *k* is an adjustable parameter between 1 and 2*n*+ 1 for an *n=2;* dimensional OTIS-*2D*-Torus where 2*n*+ 1; 5; is the longest path between any 2 nodes; the diameter. The *l*-level unsafety set S_l^P represents the set of all nodes at distance *l* from *P* which are faulty or unreachable from node *P* 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 *k* is the better the routing decisions are, but at the expense of more computation and communication overhead.

II. Notations and Definitions

The *2D*-dimensional undirected graph binary *2D*-Torus is one of the well known networks which have been used in real life systems [5, 9, 10].

The *2D*-Torus has n^2 vertices (nodes) where *n* is the number of node at each row or column. Each node *P* is labelled in the form $P=p_1p_0$, where each p_i digit satisfies: $0 \le p_i < n$. Two nodes $A=a_1a_0$ and $B=b_1b_0$ are joined by a link if, and only if, there exists *i*, $0 \le i < n$, such that $a_i = b_i \pm 1$ (mod *n*) and $a_j = b_j$ for $i \ne j$.

The *2D*-Torus has a degree of 2- Dimension; and a diameter of *n* where *n* is the number of nodes at each row or column. Figure

1 illustrates a 2D-Torus where *n*= 3, and 9 nodes. The shortest path between nodes *A* and *B* is equal to their *Lee distance* [11] given by

$$d_L(A,B) = w_1 + w_0 \text{, where}$$

$$w_i = \min_{0 \le i < 2}(|a_i - b_i|, n - |a_i - b_i|).$$

The *Hamming distance* between two nodes *A* and *B*, denoted *H*(*A*, *B*), is the number of digits at which their labels differ. A path between *A* and *B* is an *optimal path* if its length is equal to $d_L(A,B)$.

A routing algorithm *R* for a network *G* can be viewed as a function that returns the address of the next node to visit in order to achieve routing between a given source and a given destination. A fault-tolerant routing algorithm is a routing algorithm that is able to function in a network with faulty components (nodes and links).

Consider two nodes *A* and *D* where *A* is the source and *D* is the destination of a message exchange. Let $A^{(i+)}$ and $A^{(i-)}$ represent the two neighbours of node *A* along the i^{th} dimension and let $A^{(i\pm)}$ denote $A^{(i+)}$ or $A^{(i-)}$. The symbol $i\pm$ denotes the positive or negative direction along dimension *i*. If $a_i \neq d_i$, a neighbour $A^{(i\pm)}$ of *A* is called a *preferred neighbour* for routing from node *A* to *D* if $d_L(A^{(i\pm)}, D) = d_L(A, D) - 1$. We say in this case that $i\pm$ is a *preferred direction*. If $a_i \neq d_i$, a neighbour $A^{(i\pm)}$ such that $d_L(A^{(i\pm)}, D) \geq d_L(A, D)$ is called a *spare neighbour*. Neighbours other than preferred or spare are called *disturb neighbours*. For routing from *A* to *D*, a disturb neighbour $A^{(i\pm)}$ of *A* corresponds to the case $a_i = d_i$ and therefore the i^{th} digit is disturbed. Routing through a disturb neighbour increases the total routing distance by at least two over the minimum distance. Routing through a spare neighbour increases the total routing distance by at least one over the minimum distance. A minimal path can be obtained by performing a preferred direction move at every routing step. With respect to routing from node *A* to *D*, node *T* is called a preferred *transit node* if $d_L(T, D) < d_L(A, D)$.

We make the following assumptions for the proposed algorithm and performance study. Similar assumptions have been made in earlier related works, e.g. [5].

i) A faulty *k*-ary *n*-cube contains faulty nodes and/or links. The fault pattern remains fixed for the duration of calculations of unsafety sets. In other words, the faulty sets calculation has to be restarted if additional faults occur before completing the calculation.

ii) Each node can determine the status of its own links and the status of its neighbouring nodes.

iii) Node failures are fault-stop failures.

In an OTIS-2D Torus network, $N = n^4$ processors are divided into n^2 groups to form a two-dimensional lattice. Each group is basically a two-dimensional Torus (2D mesh with wraparound connections) with *n* rows and *n* columns. Let us denote node *A* is placed in the (p_1p_0) position of the (g_1g_0) group, then the combined address will become form node *A* is (g_1g_0,p_1p_0) *where* g_1g_0 is the group address and p_1p_0 is the node address within a specific group and for $0 \leq p_i,g_i < n$. Therfore, within each group, two nodes placed at (x_1x_0) and (y_1y_0) are connected if and only if $x_1 = (y_1 \pm 1) \bmod n$ and $x_0 = y_0$ or $x_0 = (y_0 \pm 1) \bmod n$ and $x_1 = y_1$. For group communication, the node $A(g_1g_0,p_1p_0)$ is connected to node $B(p_1p_0,g_1g_0)$ via an optical link. We assume that all the links are bidirectional. As an example the OTIS-*2D*-Torus is shown in Figure. 1 for *n* = 3. In this figure, the indices for each group are shown below it in boldface and the processor indices are shown adjacent to it. It possesses several topological properties that can be exploited in efficient mapping of parallel algorithms.

We differentiate the electronic and optical links as follows: (i) optical links have larger bandwidth than electronic links and (ii) transfer times including latency are different along optical and electronic links [6]. As the channel capacity, transmission property and mechanism of these two links are different; we keep a separate count for data movement on these links. We represent data movement on electronic link by electronic move and that on optical link by OTIS move for analyzing the time complexity of our proposed algorithms. These two moves actually provide the total communication latency required by the algorithms. We also count the number of different primitive mathematical operations per iteration for each of the algorithm to add for a better understandability of our proposed algorithms.

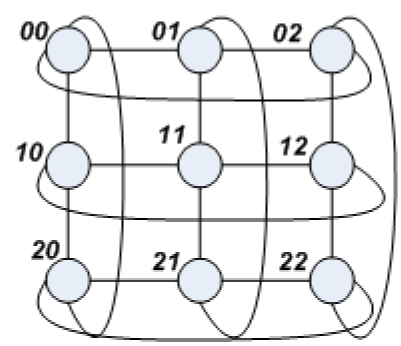
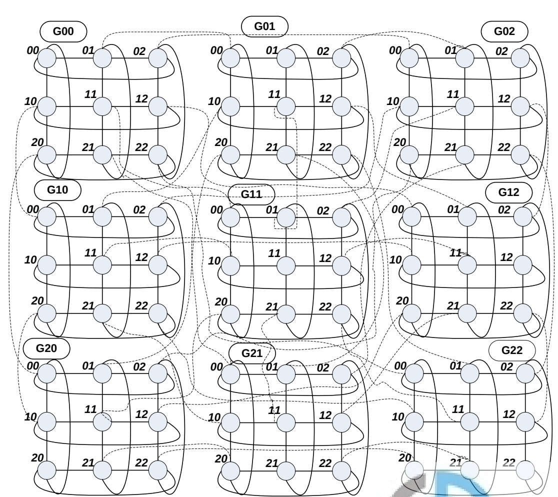


Figure 1: *2D*-Torus where *n*=3



International Journal of Advancements in Electronics & Electrical Engineering– IJAEEE 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 6 : Issue 2 [ISSN : 2319-7498] - Publication Date: 25 June , 2018

```
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 neighbours;
F_A = S_1^A ;
for k := 2 to m do // diameter
  {
    for i:=1 to n do
       if p_A^(i) ∉ F_A then   {
            send F_A to p_A^(i) ;
              receive F_A^(i) from p_A^(i) ;
              F_A = F_A ∪ F_A^(i);   }
      if p_A ≠ 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 ∪ F_<g_A,p_A> ;   }
 for k := 1 to m do
S_k^A = {<g_B,p_B> ∈ F_A | dist(<g_A,p_A>,<g_B,p_B>) = k}
End.
```

Figure 2: OTIS-*2D*-Torus where *n*=3

In the OTIS-2*D*-Torus the address of a node $u = \langle x, y\rangle$ from V is composed of two components, each component is formed of two digits $\langle x_1x_0 y_1y_0\rangle$. Figure 2 shows an 81 processor OTIS-Torus, the notation $\langle g, p\rangle$ 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, and only if, $g_1 = g_2$ and $(p_1, p_2)\in E_0$ (such that E_0 is the set of edges in torus network) or $g_1 = p_2$ and $p_1 = g_2$, in this case the two nodes are connected by transpose edge.

The distance in the OTIS-Torus 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 [12]:

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 and more than two, 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 g1 ≠ g2, 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:

$$\langle g1,\ p1\rangle \xrightarrow{E} \langle g1,\ g2\rangle \xrightarrow{O} \langle g2,\ g1\rangle \xrightarrow{E} \langle g2,\ p2\rangle.$$

The most important topological properties of the OTIS-2*D*-Torus including the following [13]:

1. **Size:** If the torus factor network of size N; 2^n; when is the number of nodes at each dimension; then the size of the OTIS-*2D*-Torus is N^2.
2. **Degree:** Let $\langle g, p\rangle$ be any node in OTIS-Torus. Then the degree (or *deg*) of the OTIS-*2D*-Torus is as follows:

$$\deg_{OTIS-2D-torus}(g,p) = \begin{cases} \deg_{G_0}(p) & if\ g = p \\ \deg_{G_0}(p)+1 & if\ g \neq p \end{cases}$$

Figure 3: The algorithm of_Unsafety_Sets that determines the faulty set for node *A*.

3. ***Number of Links***: Let l_0 be the number of links and N be the number of nodes in the torus network, then the number of links in the OTIS-*2D-Torus* = $(N^2 - N)/2 + l_0 \cdot N$.
4. ***Length***: Let $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ be two different nodes in the OTIS-*Torus*. To transmit data originated in the source node $\langle g_1, p_1\rangle$ to the destination node $\langle g_2, p_2\rangle$ we follow one of the three possible paths shown above *i*, *ii*, and *ii*. The length of the shortest path between the nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ is:

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

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 is the diameter of the torus network, the diameter of the OTIS-Torus is $2n+1$.

III. 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 OTIS-Torus calculates its unsafety sets.

The calculation of the unsafety sets is as follows:

Definition 2: The number of direct neighbours *np* of a node *A*, $<g_A,p_A>$, is defined as:

$$np = \begin{cases} n & if\ g_A = p_A \\ n+1 & Otherwise \end{cases}$$

Definition 3: 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., $\left|S_1^A\right| = np$. If for some node *A*, $\left|S_1^A\right| = np - 1$ then node *A* is called a *dead-end node*.

Each node uses the unsafety set to determine the faulty set F_A,

International Journal of Advancements in Electronics & Electrical Engineering– IJAEEE 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 6 : Issue 2 [ISSN : 2319-7498] - Publication Date: 25 June , 2018

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 neighbours. After determining F_A, node A calculates m unsafety sets denoted S_1^A, $S_2^A,\ldots,$ S_m^A (defined below), where m is an adjustable parameter between 1 and $2n+1$.

Definition 4: 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. Figure 3 gives an outline of the *Find_Unsafety_Sets* algorithm that node A uses it to determine it's faulty and unsafety sets.

With respect to a given destination node, D, in an OTIS- Torus network a neighbour $A^{(i)}$ of node A is called a

preferred neighbour for the routing from A to D if the ***i***-th bit of the processor address of $A \oplus D$ is 1 if both are in the same group or the ***i***-th bit of the processor address of $A \oplus$ the group address of D is 1. We say in this case that i is a *preferred dimension*. Neighbours other than preferred neighbours are called *spare neighbours*. Routing through a spare neighbour increases the routing distance by at least two over the minimum distance. In general, a preferred neighbour is one step closed to the destination while a spare neighbour 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.

IV.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* $U^{A,D} = (u_1^{A,D},\ldots,u_k^{A,D},\ldots,u_m^{A,D})$ where its k^{th} element is given by

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

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

```
Algorithm Unsafety_Vectors (M: message; <gc,pc>,<gd,pd>: node)
/* called by current node <gc,pc> to route the message M to
  its destination node <gd,pd> */

if <gc,pc> is source node then M.Route_distance = 0
if Route_distance<= dist(pc,pd)+dist(gc,gd) +( 2n+1) * No_FaultyNodes then {
M.Route_distance:=M.Route_distance + 1
if gc= gd and pc= pd then exit;      /* destination reached */
if gc= gd then route(<gc,pc>,<gd,pd>) /* curr & dest. at the same group */
if (dist(pc,pd)+dist(gc,gd)+2)<(dist(pc,gd)+dist(gc,pd)+1) and the two optical
   moves (gcpd→ pdgc, pdgd→ gdpd) are not faulty then
   { if pc= pd then move m to < pc, gc>
      else route(<gc,pc>,< gc, pd>) }
else if the optical move (gcgd→ gdgc) is not faulty then
   { if pc= gd then move m to < pc, gd>
      else route(<gc,pc>,<gc,gd>) }
  else if  gc ≠pc and the node<pc, gc > is not faulty
   then  send M to <pc, gc >    /* disturb the message*/
}
 else  looping

End.
Function route(<gc,pc>,<gd,pd>:node)
{if ∃ a preferred non-faulty neighbour A(i) with least
   ( A(i) ,D)-unsafety vector U^(A(i),D)  And  A(i)  is not dead-end

   then   send M to A(i)
   elseif ∃ a spare non-faulty neighbour 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  gc ≠pc and the node<pc, gc > is not faulty
  then  send M to <pc, gc >    /* disturb the message*/
    else failure          /* destination unreachable */ }
```

Figure 4: A description of the proposed Unsafety vectors routing algorithm.

Example 1: Consider the OTIS-2D-Torus depicted in Figure 3 where the source node A=1001, the destination node D=0000, 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 via optical link, note that the preferred neighbour 1000 is faulty. By performing the same operations the message will be routed through an electronic move to node 0100 then via an optical move to node 0001 and finally to its destination 0000.

Theorem 1*: Let $A^{(i)}$ and $A^{(j)}$ be two non faulty (A,D)-preferred neighbours of A. If all preferred neighbours of $A^{(j)}$ are faulty and at least one preferred neighbour 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).

V.Conclusion

This paper has proposed a specified fault-tolerant routing for the OTIS-2*D*-Torus 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 2*n* 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 2*n*+1. 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 OTIS-2*D*-Torus. 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 is to implement the proposed routing algorithm for all the different network sizes and conduct a performance analysis through extensive simulation experiments to show the superiority of the proposed algorithm using the set of unsafety vectors.

References

[1] "N- Cube Systems," N-cube Handbook, N-Cube, 1986.
[2] J. Rattler, "Concurrent Processing: A new direction in scientific computing," *Proc. AFIPS Conf*. 54, pp. 157-166, 1985.
[3] C.L. Seitz, "The cosmic cube," CACM 28, 22-23, 1985.
[4] Silicon Graphics, "Origin 200 and Origin 2000," *Technical Report*,1996.
[5] C.P. Ravikumar, C.S. Panda, Adaptive routing in k-ary n-cubes using incomplete diagnostic information, Microprocessors and Microcomputers, vol. 20, 1997, pp. 351- 360.
[6] Keny T. Lucas · Prasanta K. Jana, Parallel algorithms for finding polynomial Roots on OTIS-torus, J Supercomput (2010) 54: 139–153.
[7] S. Yadav, C. Rama Krishna, CCTorus: A New Torus Topology for Interconnection Networks, Int'l Conference on Advanced Computational Technologies & Creative Media (ICACTCM), 2014, pp 8-14
[8] P. Chauhan, M. Bhardwaj, Analysis the Performance of Interconnection Network Topology C^2 Torus Based on Two Dimensional Torus, International Journal of Emerging Research in Management &Technology, Vol. 6, No. 6, 2017, pp 169,173.
[9] Abdul Hannan Akhtar, Keny T. Lucas, Enumeration Sort on OTIS k-Ary n-Cube Architecture, IJECS Volume 3 Issue 7. July 2014 page no. 7173-7176
[10] O. AbuAlghanam, M. Qatawneh, H. Ofeishat, O. Adwan, A. Huneiti, A New Parallel Matrix Multiplication Algorithm on Tree-Hypercube Network using IMAN1 Supercomputer, International Journal of Advanced Computer Science and Applications, Vol. 8, No. 12, 2017, pp 201-205.
[11] A. Krishnamoorthy, P. Marchand, F. Kiamilev, and S. Esener, "Grain-size Considerations for Optoelectronic Multistage Interconnection Networks," *Applied Optics*, vol. 31, no. 26, pp. 5480- 5507, 1992.
[12] C.F. Wang, "Algorithms for the OTIS Optoelectronic Computers,", *PhD thesis*, Dept. of Computer Science, University of Florida, 1998.
[13] K. Day and A. Al-Ayyoub, "Topological Properties of OTIS-Networks," *IEEE Trans. On Paralle and Distributed Systems*, vol. 13, no. 4, pp. 359-366, 2002.

About Author (s):

Prof. Jehad Al-Sadi received his Ph.D. in Computer Science from the University of Glasgow, U.K. 2002. He is currently a professor and Assistant director at Arab Open University, Jordan. His research interests are: interconnection networks, optical networks, fault-tolerant routing, e-learning, and learning management systems.