International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:9, 2014

The Load Balancing Algorithm for the Star Interconnection Network

Ahmad M. Awwad, Jehad Al-Sadi

Abstract**—**The star network is one of the promising interconnection networks for future high speed parallel computers, it is expected to be one of the future-generation networks. The star network is both edge and vertex symmetry, it was shown to have many gorgeous topological proprieties also it is owns hierarchical structure framework. Although much of the research work has been done on this promising network in literature, it still suffers from having enough algorithms for load balancing problem. In this paper we try to work on this issue by investigating and proposing an efficient algorithm for load balancing problem for the star network. The proposed algorithm is called Star Clustered Dimension Exchange Method SCDEM to be implemented on the star network. The proposed algorithm is based on the Clustered Dimension Exchange Method (CDEM). The SCDEM algorithm is shown to be efficient in redistributing the load balancing as evenly as possible among all nodes of different factor networks.

Keywords**—**Interconnection networks, Load balancing, Star network.

I. Introduction

THE star graph was proposed by Akers and et al as one of the attractive topologies, it was proposed as an alternative to the cube network [1]. The star graph is shown to have excellent topological properties for comparable network sizes of the cube network [2]. The star graph showed to have many good properties over many networks including the well-known cube network including: smaller diameter, smaller degree, and smaller average diameter. The star graph proved to have a hierarchical structure which will enable it building large network size of smaller ones, the star graph is both edge and vertex symmetry [1].

Although some algorithms proposed for the star graph such as distributed fault-tolerant routing algorithm [3]. The proposed algorithm adapts the routing decisions in response to node failures; also the star network is shown to have fault tolerance properties [3]. Anyway one of the main problems that the star network still suffers from is having enough algorithms for load balancing problem.

To our knowledge there is no enough results proposed in literature about implementing and proposing efficient algorithms for load balancing on star network. In this paper we try to fill this gap by proposing and embedding the SCDEM algorithm on the star graph which is based on the CDEM algorithm which was shown to be attractive on OTIS-Hypercube network by redistributing the load as evenly among different processors [4]. Efficient implementation of the SCDEM algorithm on the star network will make the star network more acceptable network for real life application in connection to load balancing problem. The rest of the paper is organized as follows: In Section II we present the necessary basic notations and definitions, in Section III we introduce some of the related work on load balancing, in Section IV we present and discuss the implementation of the SCDEM algorithm on the star graph, also we present an example of SCDEM on S_4 star network, finally Section V concludes this paper.

Dr. Awwad works at University of Petra – Amman - Jordan FIT– CS Dept. (e-mail: awwad@uop.edu.jo).
Dr. Al-Sadi works at Arab Open University – FIT- CSD – Amman (e-mail: j_alsadi@aou.edu.jo).

II. Definitions and Basic Topological Properties

During the last decade a huge number of interconnection networks for High Speed Parallel Computers (HSPC) have been investigated and proposed in literature [5]-[7]. As an example one of these networks was the hypercube interconnection network, also this network is known as the binary *n*-cube. The *star graph* [1] is another example, which has been proposed as an attractive alternative to the hypercube network. Since its appearance the star network has attracted a lot of research efforts. Some properties of this network have been studied in the literature including its basic topological properties [1], parallel path classification [4], node connectivity [2] and embedding [8]. The authors Akers and Krishnamurthy [3] have proved that the star graph has several advantages over the hypercube network including a lower degree for a fixed network size of the comparable network sizes, a smaller diameter, and smaller average diameter. Furthermore they showed that the star graph is maximally fault tolerant edge, and vertex symmetric [3].

The structure of the star network can play an effective step for proposing any algorithms on it. The authors in [21] Menn and Somani have shown that the star graph may be seen as $n \times (n\text{-}1)!$, where the rows and the columns in this framework are $(n\text{-}1)$-star and an *n*-linear array correspondingly. Also, Ferreria and Berthome [9] claimed that the star graph may be seen also as a rectangular framework $R \times C$ (Rows by Columns) where the rows are substar-$S_{n\text{-}2}$ and the columns are n $(n\text{-}1)$ nodes on each of its column.

However, there has been relatively a limited research efforts have been dedicated to design efficient algorithms for the star graph including computing fast Fourier transforms [10], broadcasting [11], selection and sorting [12], [21], and Matrix Multiplications [13] and load balancing [14], [15]. In an attempt to overcome this problem we present an efficient algorithm for load balancing problem on star graph to redistribute the load balancing among all processors of the network as evenly as possible.

Definition 1: The n-star graph, which is denoted by S_n, has $n!$ nodes each labelled with a sole permutation $\langle n \rangle = \{1,\ldots,n\}$. Any two nodes of S_n are connected if, and only if, their corresponding permutations differ exactly in the first position and any other position.

Fig. 1 shows the 4-star graph with 4 groups each containing 6 vertices (i.e. four copies of 3-star graphs). The degree, α, and the diameter, δ, of the star graph are as follows [1], [16]:

α, of the n-star graph = n-1, where n>1.

δ, of n-star graph = $\lfloor \frac{3}{2}(n\text{-}1) \rfloor$.

III. Background and Related Work

The attractive results shown and proved by researchers in literature of the star graph make it a one of the strongest competitor's topology for High Speed Parallel Computers (HSPC) and a strong candidate network for real life applications. This fact has motivated us to investigate the load balancing problem on the star network since the star graph suffers from limited number of efficient algorithms proposed for it in general and load balancing problem as specific case. The load balancing problem has been investigated on various types of infrastructure ranging from electronic networks [15] and OTIS networks [4].

Load balancing problem is one of the well-known and important types of problems which were studied from different point views and different approaches. This problem was studied and investigated by Ranka, Won, and Sahni [17]; they proposed and introduced the Dimension Exchange Method (DEM) on the hypercube topology. The DEM algorithm was based on the idea of finding the average load of neighbours' nodes, such that the dimension of the network is n, the neighbours which are connected on the n^{th} dimension they will exchange their loads to redistribute the load and achieve evenly load balancing as possible, the processor which have more load will broadcast the extra amount of the load to its direct neighbour node. The main advantage of the Dimension Exchange Method is that every node will be able to redistribute tasks to its direct neighbours to reach even load balancing among all nodes. Ranka et al. have achieved that in the worst case in the DEM method to redistribute load balancing was log2n on the cube network [17].

The researchers Zaho, Xiao, and Qin have presented hybrid scheme of diffusion and dimension exchange called DED-X for load balancing on Optical Transpose Interconnection System (OTIS) [18], [19]. The proposed algorithm works by dividing the load balancing task to three different phases. The results achieved on OTIS networks showed that the load balance efficiently redistributed almost evenly. On the other hand the achieved results of the simulation from Zaho et al of the proposed algorithms on load balancing has shown a considerably major advancement in enhancement of efficiency and stability [18], [19]. In another research done by Zaho and Xiao they have presented different DED-X schemes for load balancing on homogeneous OTIS networks and they proposed new algorithm structure called Generalized Diffusion-Exchange- Diffusion Method, the proposed scheme enabled load balancing on Heterogeneous OTIS networks [20].

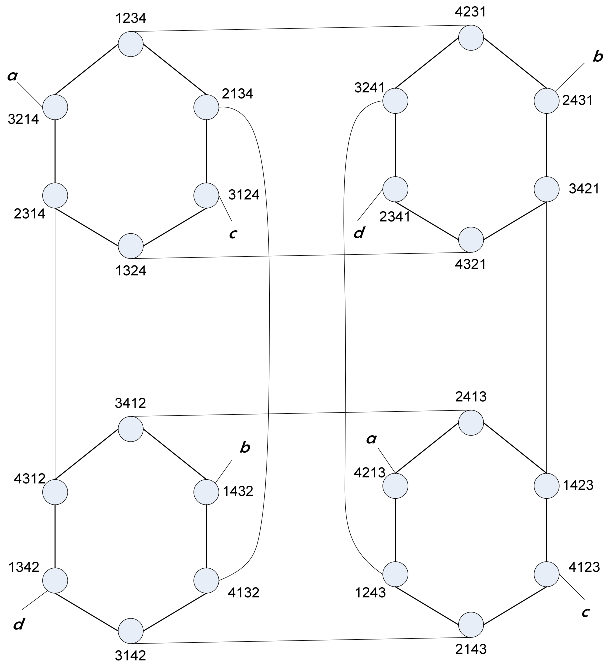

Fig. 1 The 4-Star Graph

Furthermore Zaho, Xiao, and Qin have shown that the usability of the new proposed load balancing methods to be better than the X traditional load balancing algorithm [20].

The main objective of this paper is to propose and present a new load balancing algorithm for the star networks named Star Clustered Dimension Exchange Method (SCDEM) based on the algorithm [4].

The Star Clustered Dimension Exchange Method for Load Balancing on the Star Network

The algorithm we present in this paper SCDEM is based on the Clustered Dimension Exchange Method CDEM for load balancing for Optical Transpose Interconnection system on Hypercube factor network [4]. The worst time case complexity of CDEM for load balancing on OTIS-Hypercube was O(Sqrt(p)*M log2 p). Also the number of communication steps which is required by CDEM proved to be 3log2 p [4].

The main achievement of the new presented SCDEM is to obtain even load balancing for the Sn network by redistributing number of tasks between different nodes on different groups. The number cooperating moves needed between different nodes in the SDEM is 2n-1, where n is the degree of Sn. Fig. 2 presents the SCDEM algorithm for load balancing problem on n! Processors of Sn.

The SCDEM load balancing algorithm is based on the following 3 phases:

Phase1. The load balancing of neighbour nodes of the 1st stage is achieved by redistributing the load balancing of all direct neighbour nodes, and only if, their corresponding permutations differ exactly in the 1st and 2nd position.

Then redistribute the load balancing of any two neighbour nodes that are connected if, and only if, their corresponding permutations differ exactly in the first and 3rd position.

Keep redistributing the load balancing of any two neighbour nodes "as above" that are connected if, and only if, their corresponding permutations differ exactly in the first and nth position.

Phase2. Repeat phase one more time.

Phase3. For each direct neighbour's processors pi and pj find the maximum difference weight of the directed weights and redistribute the weight among the nodes of highest difference following the steps 3-12.

Note that n-1 is the number of neighbours of any processor in S_n.

```
1.  for $n = 1$; $n \le n-1$; $n$++
2.  for all neighbour nodes $p_i$ and $p_j$ which they differ in 1st and $n$+1
    position of $S_n$ do in parallel
3.      Give-and-take $p_i$ and $p_j$ total load sizes of the two nodes
4.      TheAverageLoad $p_{i,j}$ = Floor (Load $p_i$ + Load $p_i$)/2
5.      if ( Totalload $p_i$ >= excess AverageLoad $p_{i,j}$ )
6.          Send excess load $p_i$ to the neighbour node $p_i$
7.          Load $p_i$ = Load $p_i$ – extra load
8.          Load $p_j$ = Load $p_j$ + extra load
9.      else
10.         Receive extra load from neighbour $p_j$
11.         Load $p_i$ = Load $p_i$ + extra load
12.         Load $p_j$ = Load $p_j$ – extra load
13. Repeat steps 3 to 12 one more time
14. for all adjacent processors of $p_i$ and find the max $|p_i - p_{j,}|$ such
    that $p_j$ is the set of all neighbours of pi where $1 \le j \le n-1$.
15. redistribute the weight as in steps as in steps 3 to 12 between $p_i$ and
    the node with the max $|p_i - p_j|$.
```

Fig. 2 The SCDEM load balancing Algorithm

SCDEM algorithm works on redistributing load balancing among all processors of the network, phases: one, two and three are done in parallel.

Phase1. The load balancing between the processors of S_n based on SCDEM algorithm is exchanged as in steps 2 to 12 in parallel, in first step the load exchange will be between all the processors in which they differ in 1st position and 2nd position for all the factor networks of S_n i.e. S_{n-1}. Then the same process will be repeated continually until it reach the neighbours p_j that is n positions far away from p_i.

Phase2. To enhance the load balancing efficiency between different processors of n factor networks, the algorithm suggests repeating the steps 2 to 12 as mentioned above.

Phase3. As a final phase all adjacent processors which they differ in first position and any other position i.e. p_i and p_j. The algorithms will find the maximum difference among all the weights of these neighbors and redistribute the weight between p_i and p_j with *max* difference, $|p_i - p_j|$ following the steps 2-12 of the SCDEM algorithm.

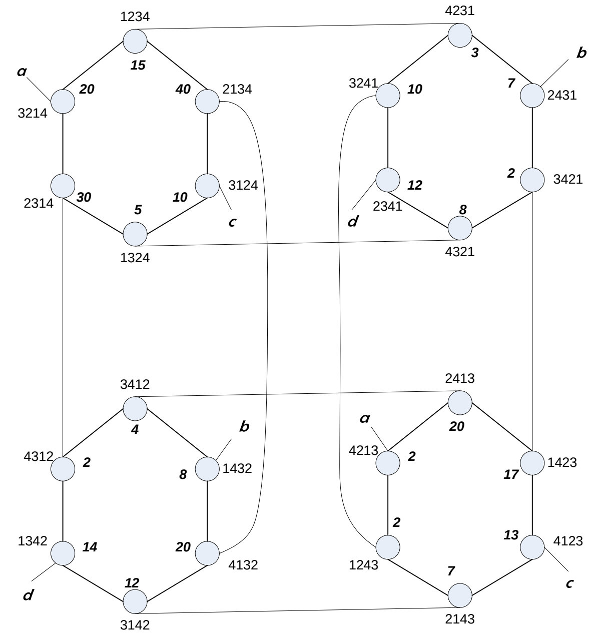

Fig. 3 4-star network – load balancing - initial state

Example: - To explain the SCDEM algorithm presented in Fig. 2, the following example implements the load balancing algorithm on the different factor networks of S_4.

Fig. 3 shows the four factor networks 4-S_3 of the Network S_4, each factor network has 6 processors with a specific load assigned to it. Since the degree of S_n is n-1 it follows that each node connected to three other direct nodes, two of the inner group and one in outer group. The number which was assigned next to the processor presents the starting load.

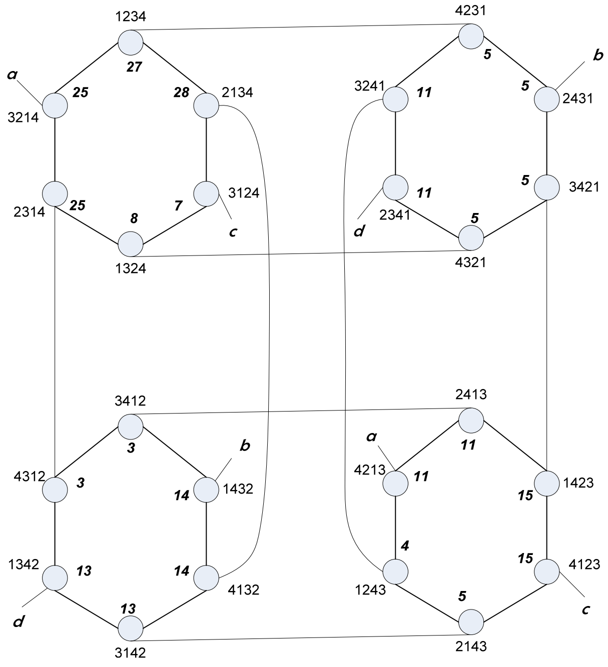

Fig. 4 4-Star network – load balancing phase 1-A

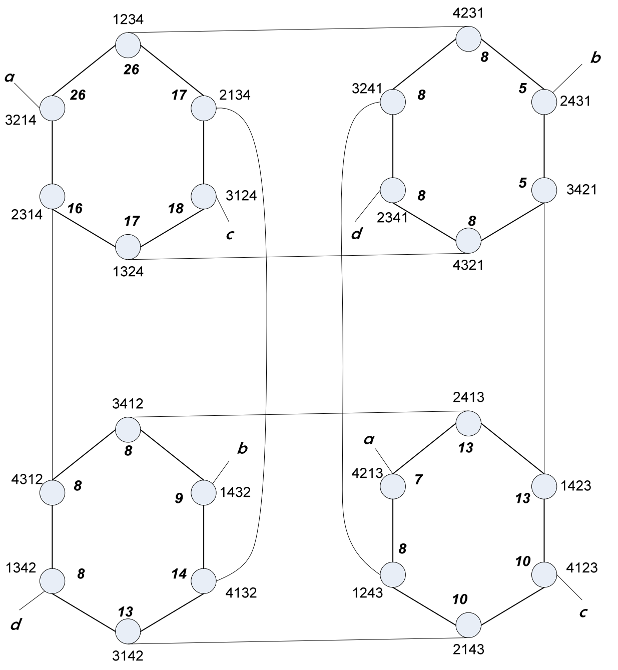

Fig 5 4-Star network – processors load balancing phase 1-B

First we start by implanting phase 1 of the algorithm by following the steps 2-12. Figs. 4-6 reflect phase1: A, B and C of the SCDEM algorithm, each pair of nodes which they differ in 1st position, 2nd, 3rd and 4th position. At the end of this phase Fig. 6 shows the new load balancing distribution of the phase 1 of the algorithm.

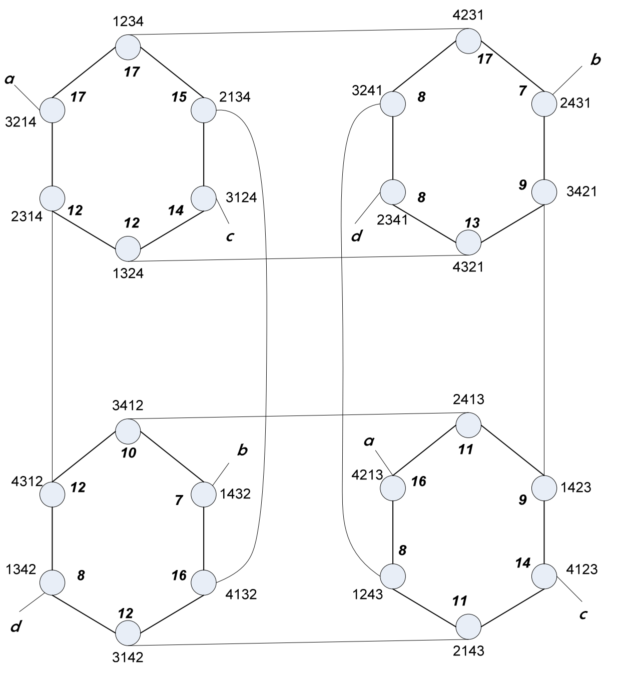

Fig. 6 4-Star network – load balancing phase 1-C

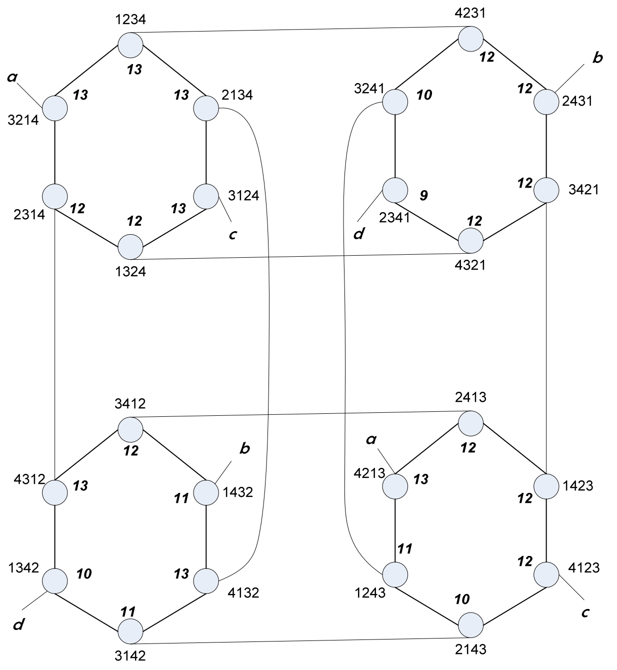

Fig. 7 4-Star network –load balancing – phase 2

In phase 2 we repeat the same steps one more time to redistribute the load balancing among the neighbours' nodes as suggested in SCDEM algorithm presented in Fig. 2 by following the steps 2- 12. Fig. 7 shows the load balancing of all processors at the end of phase 2.

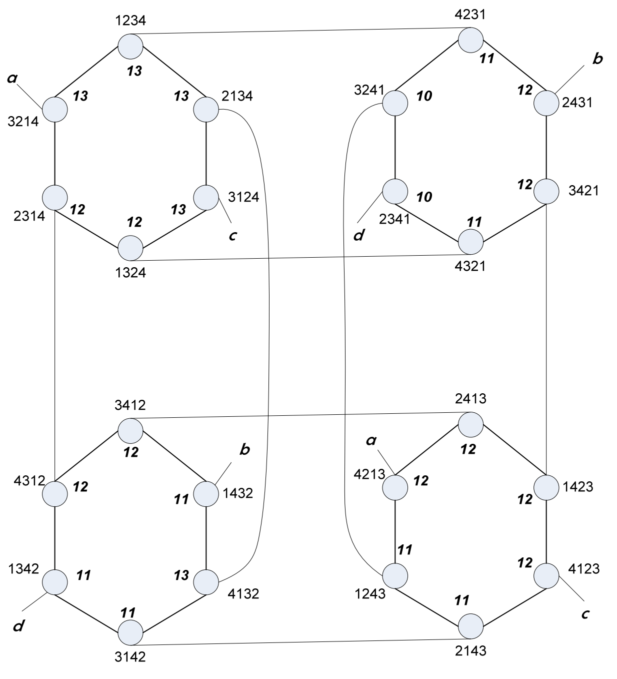

Fig. 8 4-Star network – load balancing – phase 3

Finally in phase 3 all adjacent nodes which differ in first position and any other position i.e. p_i and p_j will redistribute their load balancing by first finding the maximum difference between their weights then by implementing the steps of the algorithm between the nodes with maximum difference (Fig. 8)

The new achieved distribution of the load balance shown to be efficient and optimal. The final distribution is achieved in 2*n*-1 communication steps where *n* is the degree of the star network.

IV. Conclusion

This paper presented an efficient algorithm for distributing the load balancing of nodes in star network. The proposed algorithm is called Star Clustered Dimension Exchange Method (SCDEM) is based on the well-known algorithm which was proposed by Mahafza et al (CDEM). The proposed algorithm SCDEM resulted in almost even distributions load balancing among the all nodes of star network. The algorithm is able to redistribute load balancing among all nodes in 2*n*-1 communication steps which is considered efficient.

As future extension of this research work we will do some analytical estimation including: execution time, load balancing accuracy, communication steps and speed to prove the SCDEM efficiency mathematically.

References

[1] S. B. Akers, D. Harel and B. Krishnamurthy, "The Star Graph: An Attractive Alternative to the n-Cube" *Proc. Intl. Conf. Parallel Processing*, 1987, pp. 393-400.

[2] K. Day and A. Tripathi, "A Comparative Study of Topological Properties of Hypercubes and Star Graphs", IEEE Trans. Parallel & Distributed Systems, vol. 5.

[3] Kaled Day and Abdel-Elah Al-Ayyoub, "Node-ranking schemes for the star networks", Journal of parallel and Distributed Computing, Vol. 63 issue 3, March 2003, pp 239-250.

[4] B.A. Mahafzah and B.A. Jaradat, "The Load Balancing problem in OTIS-Hypercube Interconnection Network", J. of Supercomputing (2008) 46, 276-297.

[5] S. B. Akers, and B. Krishnamurthy, "A Group Theoretic Model for Symmetric Interconnection Networks," *Proc. Intl. Conf. Parallel Proc.*, 1986, pp. 216-223.

[6] K. Day and A. Al-Ayyoub, "The Cross Product of Interconnection Networks", *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 2, Feb. 1997, pp. 109-118.

[7] A. Al-Ayyoub and K. Day, "A Comparative Study of Cartesian Product Networks", *Proc. of the Intl. Conf. on Parallel and Distributed Processing: Techniques and Applications*, vol. I, August 9-11, 1996, Sunnyvale, CA, USA, pp. 387-390.

[8] I. Jung and J. Chang, "Embedding Complete Binary Trees in Star Graphs," *Journal of the Korea Information Science Society*, vol. 21, no. 2, 1994, pp. 407-415.

[9] Berthome, P., A. Ferreira, and S. Perennes, "Optimal Information Dissemination in Star and Panckae Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 12, Aug. 1996, pp. 1292-1300.

[10] P. Fragopoulou and S. Akl, "A Parallel Algorithm for Computing Fourier Transforms on the Star Graph," *IEEE Trans. Parallel & Distributed Systems*, vol. 5, no. 5, 1994, pp. 525-31.

[11] Mendia V. and D. Sarkar, "Optimal Broadcasting on the Star Graph," *IEEE Trans. Parallel and Distributed Systems,* Vo;. 3, No. 4, 1992, pp. 389-396.

[12] S. Rajasekaran and D. Wei, "Selection, Routing, and Sorting on the Star Graph," *J. Parallel & Distributed Computing,* vol. 41, 1997, pp. 225-33.

[13] S. Lakshmivarahan, and S.K. Dhall, "Analysis and Design of Parallel Algorithms Arithmetic and Matrix Problems," McGraw-Hill Publishing Company, 1990.

[14] N. Imani et al, "Perfect load balancing on star interconnection network", J. of supercomputers, Volume 41 Issue 3, September 2007. pp. 269 – 286.

[15] Jehad Al-Sadi, "Implementing FEFOM Load Balancing Algorithm on the Enhanced OTIS-n-Cube Topology", Proc. of the Second Intl. Conf. on Advances in Electronic Devices and Circuits - EDC 2013, 47-5.

[16] K. Day and A. Al-Ayyoub, "The Cross Product of Interconnection Networks", *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 2, Feb. 1997, pp. 109-118.

[17] Ranka, Y. Won, S. Sahni, "Programming a Hypercube Multicomputer", IEEE Software, 5 (5): 69 – 77, 1998.

[18] Zhao C, Xiao W, Qin Y (2007), "Hybrid diffusion schemes for load balancing on OTIS networks", In: ICA3PP, pp 421–432

[19] G. Marsden, P. Marchand, P. Harvey, and S. Esener, "Optical Transpose Interconnection System Architecture," *Optics Letters*, 18(13), 1993, pp. 1083-1085.

[20] Qin Y, Xiao W, Zhao C (2007), "GDED-X schemes for load balancing on heterogeneous OTIS networks", In: ICA3PP, pp 482–492.

[21] A. Menn and A.K. Somani, "An Efficient Sorting Algorithm for the Star Graph Interconnection Network," Proc. Intl. Conf. on Parallel Processing, 1990, pp.1-8.