# IMPROVED GOSSIPING ALGORITHM FOR OTIS *K*-ARY *N*-CUBE ARCHITECTURE

**Abdul Hannan Akhtar***

**Keny T. Lucas****

**Abstract:**

The OTIS (Optical Transpose Interconnection System) has been one of the popular interconnection networks for researchers and has been widely exploited to develop various parallel algorithms. In this paper, we have proposed gossiping algorithm for the OTIS *k*-ary *n*-cube architecture. Gossiping is a common requirement for some of the problems like polynomial interpolation, matrix multiplication, prefix computation and enumeration sorting. The proposed algorithm is based on some predefined data routing functions. The time complexity of the proposed algorithms on the OTIS architecture is given in terms of electronic moves and OTIS moves. It needs $O(k^n)$ OTIS moves and $O(k^{2n})$ electronic moves.

**Keywords:** OTIS *k*-ary *n*-cube, interconnection network, parallel algorithm, gossiping, time complexity.

* Government Polytechnic, Ranchi -834001

** Xavier Institute of Polytechnic and Technology, Namkum, Ranchi – 834010

## 1. Introduction

The OTIS (Optical transpose Interconnection System) [1], [2] is a hybrid system that exploits the qualities of electronic links as well as optical links for developing multicomputer. The optical links are superior to electronic links in terms of power, speed and crosstalk properties if the connect distance between processors is more than a few millimeters. The electronic links are preferred to optical links for smaller distance between processors [3], [4]. In an OTIS model, the total number of processors in the network is divided into groups and each group can be assumed to be a microchip. All the processors within a group are connected through the electronic links whereas the processors of different groups are connected through the optical links. The number of groups in the network can be equal to the number of processors in each group for maximized bandwidth and minimized power consumption [1]. The interconnection pattern of processors within each group determines the overall model of such system, i.e. an OTIS-G has G interconnection pattern for all of its groups. Some of the OTIS models are OTIS-Ring, OTIS-Mesh, OTIS-Hypercube, OTIS-Torus, OTIS-Mesh of trees [2], [5]. In the recent years, many parallel algorithms have been proposed for various OTIS models that includes image processing [6], matrix multiplication [7], [33] , basic operations [8], BPC permutation [9], sorting [10], [34], [35], [36], randomized algorithm [11], extreme finding [12], decentralized consensus protocol [13], polynomial interpolation [14], [15], [own], polynomial root finding [14], [37], construction of conflict graph [16] , gossiping [17], [18], [32] etc.

The efficiency of many real-life problems, such as matrix multiplication, polynomial interpolation, root finding, scheduling, sorting etc. depend on the efficiency of data exchange among the processing nodes of the architectures. The proposed algorithm gives the general picture for $n$-dimensional architecture whereas in [32], the algorithm is presented only for $n$=3. The algorithm presented here for $k^{2n}$ processing nodes requires O($k^n$) OTIS moves and O($k^{2n}$) electronic moves. The rest of the paper is organized as follows. Section 2 describes the topology of the OTIS $k$-ary $n$-cube. Our proposed algorithm is discussed in section 3 followed by conclusion in section 4.

**2. Topology of OTIS *k*-ary *n*-cube** An OTIS *k*-ary *n*-cube network consists of $k^{2n}$ processors together in the network. There are $k^n$ roups in the network and each group has $k^n$ processors. Each dimension within the group hasprocessors and as each group is *n*-dimensional structure, there are $k^n$ processors in each group.As evident from Fig. 1, these $k^{2n}$ processors are divided into $k^n$ groups and each group nocessors within it. In this architecture, each group is *k*-ary *n*-cube with three dimensional grid structures. The two most variant of *k*-ary *n*-cube are hypercube ($k$ = 2) and torus ($n$ = 2 or 3) that have attracted researchers to develop parallel algorithms for real life problems. The hypercube architecture has been used in iPSC/2 [27] and iPSC/860 [28] whereas the torus has been used in J-Machine [29], CRAY-3TD [30] and CRAY-3TE [31] parallel computers. The electronic links are used to connect intra-group

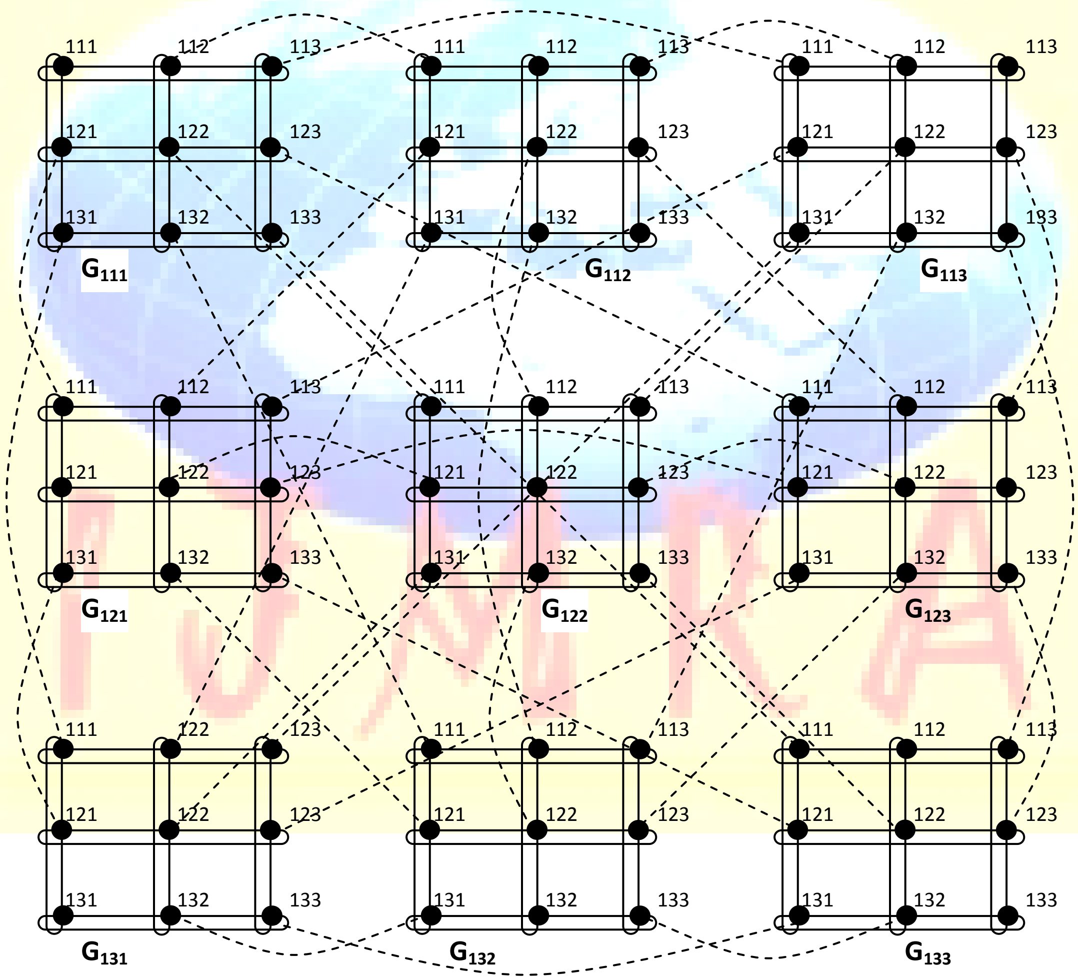


Figure 1. Topology of OTIS *k*-ary *n*-cube for $k$ = 3 and $n$ = 3 for $g_1$=1 and $p_1$=1

processors whereas the processors of a group are connected to the processors of other groups through the optical links using optical transpose rule. Let each processor be denoted by

$(G_{g_1 g_2 g_3 \ldots g_x} P_{p_1 p_2 p_3 \ldots p_x})$, then each optical link connects processor $(G_{g_1 g_2 g_3 \ldots g_x} P_{p_1 p_2 p_3 \ldots p_x})$ and $(G_{p_1 p_2 p_3 \ldots p_x} P_{g_1 g_2 g_3 \ldots g_x})$ where $1 \le g_1, g_2, g_3, g_x, p_1, p_2, p_3, p_x \le k$. The topology of OTIS $k$-ary $n$-cube for $k$ = 3; $n$ = 3 for $g_1$=1 and $p_1$=1 is shown in Fig. 1, where $k$ represents the number of processors in each dimension and $n$ represents the dimension of each group. The optical links are represented by dotted lines and the electronic links are represented by solid lines in Fig. 1.

## 3. Proposed Algorithm

Our proposed algorithm is valid for n dimensional OTIS $k$-ary $n$-cube architecture as compared to algorithm presented in [32] for 3-dimensional architecture. First, we would like to define some primitive data routing functions that will be used in the algorithm.

```
Rotate (i)
{
        /* For all groups, do in parallel */
        {
                /* For all processors in the ith dimension, do in parallel */
                        {
                                Move the content of Register B clockwise through the ring
                                Move the content of Register C anticlockwise through the ring
                        }
        }
}
```

The procedure *Rotate* ($i$) is responsible to move the data of one processor to its adjacent processors in the $i^{th}$-dimension by moving the content of Register $B$ in the clockwise and the content of Register $C$ in anticlockwise direction through the ring as shown in Fig. 2.

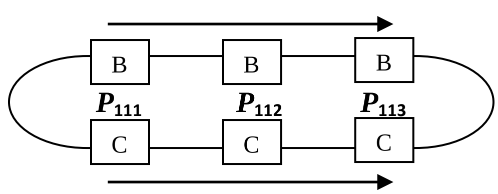


Figure 2. Moving $B$ clockwise and $C$ anticlockwise

***LocalBroadcast* (*n*)**

```
{
    /* For all groups, do in parallel */
    {
        /* For all processors, do in parallel */
        {
            For t=1 to k-1
            {
                Rotate (n)
                LocalBroadcast (n-1)
            }
        }
    }
}
```

The *LocalBroadcast* ($n$) procedure achieves gossiping within each group of OTIS $k$-ary $n$-cube architecture. As there are $k^n$ processors within each group, the time complexity of the procedure *LocalBroadcast* ($n$) can be expressed as $O(k^n)$ electronic moves. Now, it is required to make the data of each processor of a group to all the processors of the rest of the groups. This can be achieved by inter-group data movement followed by gossiping within the group. The inter-group data movement is given in the procedure *GroupMove* ($i$) as given below.

***GroupMove* (*i*)**

```
{
    /* For all groups, do in parallel */
    {
        /* For all processors, do in parallel*/
        {
            OTIS move on Register B and C
            Rotate (i)
            OTIS move on Register B and C
```

}

}

}

This will move the data of each processor from one group to the respective processors of another group in the $i^{th}$ -dimension. The *LocalBroadcast* ($n$) followed by each *GroupMove* ($i$) finally solves gossiping problem for the given OTIS $k$-ary $n$-cube architecture. Thus the final algorithm can be given as follows:

**Algorithm *Gossiping***

Step 1: Data Initialization

/* For all groups, do in parallel */

/* For all processors, do in parallel */

$B_{p_1p_2p_3\ldots p_k} = \alpha_{p_1p_2p_3\ldots p_k}$

$C_{p_1p_2p_3\ldots p_k} = \beta_{p_1p_2p_3\ldots p_k}$

Step 2: ***GroupBroadcast* (*n*)**

```
{
    GroupBroadcast (n-1)
    /* For all the groups, do in parallel */
    {
        For t=1 to k-1
        {
            GroupMove (n)
            LocalBroadcast (n)
            GroupBroadcast (n-1)
        }
    }
}
```

***Time complexity***: The procedure *Rotate* requires $O(k)$ electronic moves. *LocalBroadcast* needs $O(k^n)$ electronic move as there are $k^n$ processors within the group. Each *GroupMove* takes one electronic move and two OTIS moves. Our proposed algorithm *Gossiping* takes $O(k^{2n})$ electronic moves and $O(k^n)$ OTIS moves.

## 4. Conclusion

The proposed algorithm in this paper can be compared with the algorithm proposed in [32]. The algorithm presented in [32] requires $k^4 - 0.5k^2 + 0.5k$ electronic moves and $2(k^2\text{-}1)$ OTIS moves and offers validity only for $n$=3. The algorithm presented here is scalable for any value of $n$ for OTIS $k$-ary $n$-cube architecture and requires $O(k^n)$ OTIS moves and $O(k^{2n})$ electronic moves.

## References


[1] Feldman M. et al., 1990, Comparision between Eelectrical and Free-space Optical Interconnects based on Power and Speed Consideration, Applied Optics, 27(9), pp 1742-1751.

[2] Krishnamoorthy A. et al., 1992, Grain-size Consideration for Optoelectronic Multistage Interconnection Networks, Applied optics, 31(26), pp 5480-5507.

[3] Kiamilev F. et al., 1991, Performance Comparision between Optoelectronic and VLSI Multistage Interconnection Networks, Journal of Light Wave Technology, 9(12), pp 1674 - 1692.

[4] Marsden G. C. et al., 1993, Optical Transpose Interconnection System Architecture, Optics Letters, 18(13), pp 1083-1085.

[5] Hendrick W. et al., 1995, Modeling and Optimization of the Optical Transpose Interconnection System, Optoelectronic Technology Center, Program Review, Cornell University.

[6] Zane F. et al., "Scalable Network Architecture using the Optical Transpose Interconnection System (OTIS)", International Conference on Massively Parallel

Processing using Optical Interconnections (MPPOI' 96), San Antonio, Texas, pp 114-121.

[7] Rajasekaran S. et al., 1998, Selection and Sorting on OTIS-Mesh, IEEE Transaction on Parallel and Distributed Systems, 9(9), pp 833-840.

[8] Sahni S. and Wang C. –F., "BPC Permutations on the OTIS Mesh Optoelectronic Computer", International Conference on Massively parallel Processing using Optical Interconnections, pp 130-135.

[9] Wang C. –F. and Sahni S. (1998). OTIS Optoelectronic Computers, Parallel Computation using Optical Interconnections, Kluwer Academic Publication, Norwell, MA.

[10] Wang C. –F. and Sahni S., 1998, Basic operations on the OTIS-Mesh Optoelectronic Computer, IEEE Transactions on Parallel and Distributed Systems, 9(12), pp 1226-1236.

[11] Wang C. –F. and Sahni S., 2001, Matrix Multiplication on the OTIS-Mesh Optoelectronic Computer, IEEE Transactions on Computers, 50(7), pp 635-646.

[12] Wang C. –F. and Sahni S., 2000, Image Processing on the OTIS-Mesh Optoelectronic Computer, IEEE Transactions on Parallel and Distributed Systems, 11(2), pp 97-109.

[13] A. Osterloh, "Sortin on OTIS-Mesh", International Parallel and Distributed Processing Symposium (IPDPS 2000), pp 269-274.

[14] Jana P. K., 2006, Polynomial Interpolation and Polynomial Root Finding on OTIS-Mesh, Parallel Computing, 32(4), pp 301-312.

[15] Sahni S. and Wang C. –F., 1998, BPC Permutations on the OTIS-Hypercube Optoelectronic Computer, Informatica, 22, pp 263-269.

[16] Lucas K. T. et al., "Parallel Algorithms for the Conflict Graph on the OTIS-Triangular Array", International Conference on Distributed Computing and Networking, Springer-Verlag Berlin Heidelberg, pp 274-279.

[17] Lucas K. T., "The Gossiping on OTIS-Hypercube Optoelectronic Parallel Computer", International Conference on Parallel and Distributed Techniques and Applications, Las Vegas, Nevada, USA, pp 185-189.

[18] Dey K., 2004, Optical Transpose *k*-ary *n*-cube Networks, Journal of System Architecture, 50, pp 697-705.

[19] Coudert D. et al., "Isomorphism and Free Space Optical Networks", IEEE International Parallel and Distributed Processing Symposium ( IPDPS 2000), pp 769 – 774.

[20] C. –F. Wang., (1998). Algorithms for the OTIS Optoelectronic Computer", Ph. D. Thesis, Department of Computer Science, University of Florida.

[21] V. Kumar et al. (1993). Introduction to Parallel Computing – Design and Analysis of Algorithms, Benjamin-Cummings Publishing Company Inc.

[22] Chen M. S. et al., "Decentralized Consensus Protocol with Multiport Communication", International. Conference on Distributed Computing Systems, pp 356-365.

[23] Gonzalez T. F., 2003, An Efficient Algorithm for Gossiping in the Multicasting Communication Environment, IEEE Transactions on Parallel and Distributed Systems, 14(7), pp 701-708.

[24] Day K., 2002, Topological Properties of OTIS Networks, IEEE Transactions on Parallel and Distributed Systems, 13(4), pp 359-366.

[25] Hedetniemi S. M. et al., 1998, A Survey of Gossiping and Broadcasting in Communication Networks, NETWORKS, 18, pp 129-134.

[26] Fraigniaud P. and Lazard E., 1994, Methods and Problems of Communication in Usual Networks, Discrete Applied Mathematics, 53, pp 79-133.

[27] Nugent S. F., "The iPSC/2 Direct-Connect Communication Technology", International Conference on Hypercube Concurrent Computers and Applications, pp 51-60.

[28] Vanvoorst B. et al., "Workload of an iPSC/860", Scalable High-Performance Computing Conference, pp 1-31.

[29] M. Noakes et. al., “The J-Machine multicomputer: an architectural evaluation”, International Symposium on Computer Architecture, pp 224-235.

[30] Kessler R.E. and Schwarzmeier J. L., “CRAY T3D: A New Dimension for Cray Research,” CompCon, Housten, Texas, pp 176-182.

[31] Anderson E. et al., “Performance of the Cray T3E”, ACM/IEEE Conference on Supercomputing, San Jose, California, pp 1-7.

[32] Lucas K. T., “The Gossiping Algorithm for OTIS *k*-Ary *n*-Cube Parallel Computer”, International Conference on Parallel and Distributed Processing Techniques and Applications, pp 253-257.

[33] Mallick D. K. and Jana P. K., “Matrix Multiplication on OTIS *k*-Ary 2-Cube Network”, International Conference on Parallel and Distributed Processing Techniques and Applications, pp 224-228.

[34] Lucas K. T. and Jana P. K., 2010, Sorting and Routing on OTIS-Mesh of Trees, Parallel Processing Letters, 20(2), pp 145-154.

[35] Lucas K. T., “Parallel Enumeration Sort on OTIS-Hypercube”, IC3, pp 21-31.

[36] Lucas K. T., “Parallel algorithm for Sorting on OTIS-Ring Computer”, Bangalore Annual Compute Conference (COMPUTE 2009), pp 1-5.

[37] Lucas K. T. and Jana P. K., 2010, Parallel Algorithms for Finding Polynomial Roots on OTIS-Torus, Journal of Supercomputing, 54(2), pp 139-153.

[38] Lucas K. T., 2009, Parallel Algorithm for Prefix Computation on OTIS *k*-ary *n*-cube Parallel Computer, International Journal of Recent Trend in Engineering, 1(1), pp 560-562.