AN EFFICIENT ALGORITHM TO CALCULATE THE CONNECTIVITY OF HYPER-RINGS DISTRIBUTED NETWORKS

Ashraf A. Aly[1], and Tom Altman[2]

[1]Departement of computer Science and Information Systems, Morehead State University, USA
[2] Department of Computer Science, University of Colorado at Denver, USA

ABSTRACT

The aim of this paper is develop a software module to test the connectivity of various odd-sized HRs and attempted to answer an open question whether the node connectivity of an odd-sized HR is equal to its degree. We attempted to answer this question by explicitly testing the node connectivity's of various odd-sized HRs. In this paper, we also study the properties, constructions, and connectivity of hyper-rings. We usually use a graph to represent the architecture of an interconnection network, where nodes represent processors and edges represent communication links between pairs of processors. Although the number of edges in a hyper-ring is roughly twice that of a hypercube with the same number of nodes, the diameter and the connectivity of the hyper-ring are shorter and larger, respectively, than those of the corresponding hypercube. These properties are advantageous to hyper-ring as desirable interconnection networks. This paper discusses the reliability in hyper-ring. One of the major goals in network design is to find the best way to increase the system's reliability. The reliability of a distributed system depends on the reliabilities of its communication links and computer elements.

KEYWORDS

Hyper-ring Network, reliability, Network Connectivity, Network Conductivity

1. INTRODUCTION

Although the hypercube is quite powerful from a computational point of view, there are some disadvantages to its use as an architecture for parallel computation. One of the most obvious disadvantages is that the node degree of the hypercube grows with its size. This means, for example, that processors designed for an N-node hypercube cannot later be used in a 2 N-node hypercube. Moreover, the complexity of the communications portion of a node can become fairly large as N increases. For example, every node in a 1024-processor hypercube has 10 neighbours, and every node in a one million-processor hypercube has 20 neighbours.

In this paper we present hyper-rings (HRs for short). The number of links between processors in a HR is roughly twice that of a HC with the same number of processors, but the proposed organization possesses a number of advantages over that of a HC for any N we can construct a HR with N processors, whereas a hypercube must contain exactly 2^n processors for some positive n (Han et al., 1988; Awad et al., 2013; Huo et al., 2010; Monton et al., 2008; Fan et al., 2014).

1.1 Hyper-Rings

Definition: A circulant graph, G = (V, E) is called a hyper-ring with N nodes (N-HR for short) if V = {0, 1,…, N-1} and E = {{u, v} 1 v-u modulo N is power of 2}, (Altman, et al., 1994). We use a node for a processor and an edge for a link. Hyper-rings (HRs), are a multi-machine organization that is, in a sense, a generalization of the standard hypercube architecture. An example of a 11-HR is shown in Figure. 1.

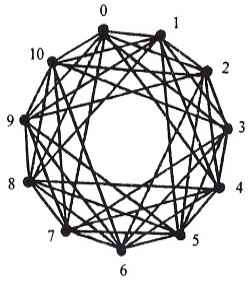

Figure 1. 11-HR

1.2. Advantages of HRs over HCs

Hyper-rings are regular graphs. Although the number of edges in a hyper-ring is roughly twice that of a hypercube with the same number of nodes, the diameter and the connectivity of the hyper-ring are shorter and larger, respectively than those of the corresponding hypercube. These properties are advantageous to hyper-rings as desirable interconnection networks.

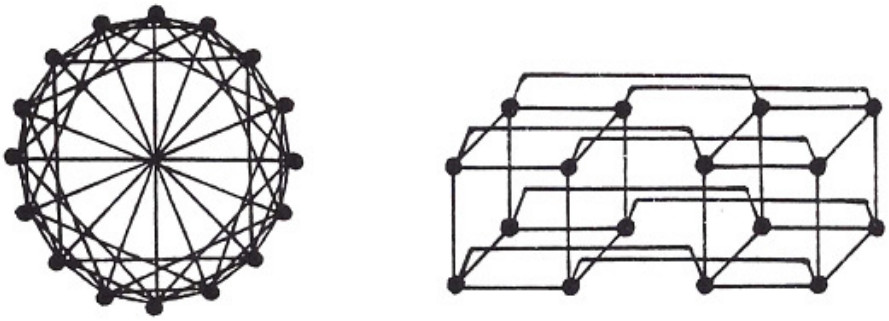

Figure 2. 16-HR and 16-HC.

A number of topologies have been proposed for interconnecting processors in large scaled distributed system. The hypercube (HC for short) is shown in Figure 2, is one of the most popular and efficient networks due to its topological richness. Hyper-rings and their variations have appeared in literature under several different names, including an optimal broadcasting scheme. Altman, et al., (1997) showed that the number of links between processors in a HR is roughly twice that of a HC with the same number of processors, but the proposed organization processes a number of advantages over that of a HC. In particular, for any N we can construct, a HR with N

processors, whereas a hypercube must contain exactly 2^n processors for some positive n. Altman et al., (1997) showed that advantage of HRs over HCs:

- The number of processors in HRs does not have to be a power of 2.
- The number of edges in a HR is roughly twice that of a same-sized HC.
- HCs and complete binary trees can be directly embedded in HRs.

1.3 The Number of Edges in HRs:

Let N be a positive integer. The binarity of N, written bin (N), is a function whose range is between 0 and [Log N] that returns the largest integer K, such that N is divisible by 2^k. Observe that any positive integer N may be represented as the following product.

$$N = 2^{\text{bin (N)}} M \quad (1)$$

Where M is an odd number. Altman et al., (1997) showed that the number of edges in HRs does not grow monotonically with the number of nodes. Theorem (Altman et al., 1994).

1.4 Connectivity in HRs

In the case of computer networks, the edge v node connectivity corresponds to the minimum number of communication links or computer centres. This criterion is most meaningful if all centres in the network are of equal importance. Otherwise, it is desirable for certain pairs of nodes in the graphs to have a larger edge or node connectivity than others. The minimum number of nodes in any i-j cut of a graph is equal to the maximum number of node disjoint paths between nodes n_i and n_j. Similarly, the edge connectivity between nodes n_i and n_j is equal to the maximum number of edge disjoint paths between that pair of nodes. Menger's fundamental theorem has simplified the determination of the node and edge connectivity between any pair of nodes in a graph (Aiello et al., 1991).

1.5 Problems of Edge Connectivity

We can calculate EC (G) by finding a maximum flow on a network G^1 obtained from G by replacing each undirected edge {u, v} in G by a pair of directed edges (u, v) and (u, v) in G^1 and assigning a unit capacity to each edge of G^1. The pairwise edge disconnecting set problems can then be solved using flow methods. The flow bearing paths of a flow on G^1 correspond directly to a set of edge disjoint paths in G. The edges of a minimum cut between a pair of vertices in G^1 determine a minimum cardinality edge disconnecting set in G between those vertices. We can calculate EC (G) by selecting an arbitrary vertex v in G and calculating the minimum of EC (G, u, v) overall distinct vertices u in G, and the given v.

1.6 Reliability in Distributed Systems

One of the major goals of the reliability engineer is to find the best way to increase the systems reliability. Usually, the system is a reliability measure of how well a system meets its design objective and is expressed as a function of the reliability of the subsystems or components. Thus,

there are many factors that influence the reliability of a system. It is well accepted that improving the reliability of the system can be achieved by one or a combination of the following techniques:

- reducing the complexity of the system
- improving the reliability of components or subsystems
- implementing large safety factors
- using redundant components (standby, switched)
- practicing a planned maintenance and repair schedule

Many of these techniques are contradictory to others and it is extremely important to consider all the aspects of the system. Computing exact reliability and performing optimization is a complex process and some simplifying assumptions about the system network configurations are made so that the problem can be handled with reasonable effort.

The reliability of a distributed system depends on the reliabilities of its communication links and computer elements, as well as on the distribution of its resources, such as programs and data files. Useful measure of reliability in distributed systems is the terminal reliability between a pair of nodes which is the probability that at least one communication path exists between these nodes. An interesting optimization problem is that of maximizing the terminal reliability between a pair of computing elements under a given budge constraint. Analytical techniques to solve this problem are applicable only to special forms of reliability expressions.

2. RELATED WORK

Hyper-cube and hyper-ring are two well-known network and appropriate for large scale multicomputer (Yanney et al.,1984; Raghavendra et al.,1985; Feng et al., 1996). The previous studies of hyper-ring and hyper-cube were limited to the case in which each node communicates uniformly with every other nodes. Hyper-rings and hyper-cube architectures are evaluated as interconnection networks for multicomputer. Comparisons are made between hyper-rings and hyper-cube networks under different communication types to show the advantages and disadvantages of these networks (Yanney et al., 1984; Raghavendra et al., 1985; Feng et al., 1996). This paper discusses the reliability in hyper-ring. One of the major goals of the reliability engineer is to find the best way to increase the system's reliability.

3. ALGORITHM

A graph G = (V, E) is called a hyper-ring with N nodes (N-HR for short) if V = {0,…,N-1} and E = {{u, v} 1 v-u modulo N is a power of 2}. We present and embedding of the n-dimensional hypercube into 2^n-hyper-ring.

3.1 Construction of HRs

Let N be a positive integer. The binary of N, written bin (N), is a function whose range is between 0 and [log N] that returns the largest integer K, such that N is divisible by 2. Observe that any positive integer N may be represented as the following product.

$$N = 2^{bin(N)} \; M \qquad (2)$$

Where M is an odd number. First, let us observe a simple (non-recursive) construction of N-HRs. The following procedure is straightforward and needs no further explanation (Altman, et al., 1994). Although the above is an effective and correct procedure, it gives us no insight about the actual number of edges placed in the HR. In addition, no obvious information about recursive structure relates HRs construction in this fashion. If E(N) denotes the number of edges in an N-HR, then

$$E(2N) = 2E(N) + 2N \quad (3)$$

The problem, of course, is the construction and the determination of the number of edges in the original N-HR. In fact, the problem may be reduced to examinations of M-HRs, where M is an odd positive integer in Equation (1). The non-recursive construction of N-HRs may be used for construction of such M-HRs, with a slight modification. Observe that for odd M of the form we can construct the M-HR by using the non-recursive construction procedure that ignores the last iteration of the inner for loop. Note that in this case the last iteration of the inner for loop produces the same connections as the connections produced in the first iteration of the inner for loop. That is, the inner for loop should be integrated one less time to produce the M-HR. For the construction of an arbitrary N-HR one simply needs to compute k = bin (N), determine the odd M in Equation (1), construct the appropriate M-HR, and recursively applied the doubling construction procedure exactly k times.

3.2 The Number of Edges in HRs

Let N be a positive integer. The binarity of N, written bin (N), is a function whose range is between 0 and [Log N] that returns the largest integer K, such that N is divisible by 2^k. Observe that any positive integer N may be represented as the following product.

$N = 2^{\text{bin (N)}} M$

Where M is an odd number. Altman et al., (1997) showed that the number of edges in HRs does not grow monotonically with the number of nodes. Theorem (Altman et al., 1994).

Number of edges in an N-HR, is equal to:

N (Log N-1/2)-------if HW (N) = 1
N (Log N-1/2)-------if HW (N) = 2
N (Log N-1/2)-------if HW (N) = 7, 3

Where HW (N) denotes the Hamming of weight of N.

Proof: If the Hamming weight of N is 1, then $N = 2^n$, $n \geq 1$. Observe that during the execution of the nonrecursive construction procedure of N-HR, no multiple edges could possibly be constructed until the very last iteration of the inner for loop. During that iteration, exactly half of the edges would be eliminated. It follows that the number of (nonreplicated) edges when N is a power of 2, is $N \log N - N/2(\log N - 1/2)$.

A Hamming weight of 2 implies that N can be decomposed into a sum of 2^p and 2^q. Moreover, this decomposition must be unique. Because $N = 2^p + 2^q$, for each pair of nodes that were 2^p and 2^q positions apart, exactly one extra edge is eliminated due to duplication in the nonrecursive

construction procedure of *N*-HRs. Hence, the number of edges is *N*[log *N*] - *N*, in this case, is equal to *N*[log *N*].

In the case where Hamming weight is 3 or more, the non-recursive construction procedure of HRs does not attempt to create any duplicated edge even if the condition is removed. Hence, the number of edges in this case is *N*[log *N*].

Example: Assume we have 11-HR as is shown in Figure 1. Where N is a positive integer, (number of nodes = n^0 of processes) HW is the Hamming weight.
n^0 of nodes = N = 11
$(11)_{10} = (1_0\ 11)_2$

The Hamming weight of N nodes = HW (11) = 3 = number of 1's in base 2. By using theorem 1, we can find the number of edges in 11-HR.

HW (11) = 3

In number of edges in 11-HR = *N*[log *N*] where HW (N) 7/3

In number of edges in 11-HR = *N*[log *N*]
Log N = Log_2 11 = 3.2
[3.2] = 4

From number of edges in 11-HR = 11 (4) = 44 edges, now we can find the number of edges connected to each node.

$$\text{Number of edges} = \frac{\text{number of nodes} * n^0 \text{ of edges connected to each node}}{2}.$$

$$\text{In number of edges connected to each node} - \frac{2*44}{11} = 8 \text{ edges}$$

3.3 Embedding of Hyper-Cubes into Hyper-Rings

Since hypercube are commonly used networks, the study of embedding of hypercube in hyper-rings will be useful. Altman et al., (1997) showed that for any n, the n-dimensional hypercube can be embedded in the hyper-ring with 2_n nodes as its sub-graph where: A graph G = (V, E) with N nodes is called the N hyper-ring (N-HR for short) if V = {0,…,N-1} and E = {{u, v} 1 v-u modulo N is a power of 2}.

The n-dimensional hypercube (n-HC for short) is a graph with N-2^n nodes labeled by 2^n binary numbers from 0 to 2^n-1. Where there exists an edge between two nodes if and only if their binary representations of their labels differ by exactly one bit. The HC can be used to simulate many networks since it contains or nearly contains those networks as sub-graph. This is one of the main reasons why the HC is a powerful network for parallel computation from the structures of N-HCs and N-HRs, it is clear that for N = 2^n, the N-HR contains N-HC as a sub-graph, making the HR an even more powerful network for parallel computation.

From the structures of N-HCs and N-HRs, it is clear that for $N = 2^n$, the N-HR contains N-HC as a sub-graph, making the HR an even more powerful network for parallel computation. Note that an N-HC is embeddable into an Q-HR with stretch factor of one as long as $Q \geq N$.

1 - dimensional grid (size 1 x K)

A 1-dimensional grid of size 1 x K can be embedded as a sub-graph into a K-HR in an obvious way. That is, there are no wasted nodes in the HR. Note that for HCs, if K is not a power of 2, e.g., $K = 2^m + 1$, then the HC will need $2^{m+1} = 2K-2$ nodes.

2- dimensional grid (size a x b)

To embed a grid of size a x b as a sub-graph into a HR, we need min ($a \times 2^{[\text{Log } b]}$], $bX2^{[\text{Log } a]}$) nodes.

3-dimensional grid (size a x b x c)

The number of nodes required in the HR to embed a x b x c grid is

min ($a \times 2^{[\text{Log } b]} \times 2^{[\text{Log } c]}$, $b \times 2^{[\text{Log } a]} \times 2^{[\text{Log } c]} \times 2^{[\text{Log } b]}$).

In general, when embedding K-dimensional grids into HRs we have one extra degree of freedom, i.e., the size of one of dimensions can be the same as the original grid which is not necessarily a power of 2. It follows that more nodes are wasted when the grid embedding is done in HCs than HRs (unless each of the grid dimensions is a power of 2).

6. MAXIMUM FLOW FOR ODD-SIZED HRS

Altman [3] showed that the edge connectivity of HR is equal to its nodal degree, which optimum. However, it is still an open question whether the node connectivity of an odd-sized HR is equal to its degree. We have attempted to answer this question by explicitly testing the node connectivity's of various odd-sized HRs. The maximum-flow problem is the simplest problem concerning flow networks. Which testing the greatest rate at which material can be shipped from the source to the sink without violating any capacity constraints. We shall use the classical method of Ford and Fulkerson for finding maximum flows.

6.1 The Ford-Fulkerson method

Ford-Fulkerson method for solving the maximum-flow problem. It depends on three ideas:

- Residual networks
- Augmenting paths
- Cuts of flow networks

These ideas are essential to the max-flow min cut theorem, which characterizes the value of a maximum flow in terms of cuts of the flow networks. The Ford-Fulkerson method is iterative. We start with f(u, v) = 0, for all u and v, giving an initial flow of value 0. We increase the flow value by finding an “augmenting path”, as a path from source S to the sink t along which we can push

more flow. We repeat this process until no-augmenting path can be found. This process yields a maximum flow.

-Residual networks

Consists of edges that can admit more net flow. Suppose we have a flow network G = (V, E) with source s and sink t. Let f be a flow in G, and consider a pair of vertices u and v. Residual capacity: is the amount of additional net flow we can push from u to v before exceeding the capacity

C(f) = (u,v).

C(f) = c(u,v)-f(u,v).

-Augmenting paths

Given a flow network G = (V, E) and a flow f. An augmenting path p is a simple path from s to t in the residual network G(f). The residual capacity of p, is the maximum amount of net flow that we can send along the edges of an augmenting path p.

-Cuts of flow networks

The ford-Fulkerson method repeatedly augments the flow along augmenting paths until a maximum flow has been found. The max-flow min cut theorem, tells us that a flow is maximum if and only if its residual network contains no augmenting path. A cut (S, T) of flow network G = (V, E) is a partition of V into S and T = V-S such that s S and t T. If f is a flow, then the net flow across the cut (S, T) is defined to be f (S, T). The capacity of the cut (S, T) is C(S, T).

6.1.2 The basic Ford-Fulketson algorithm

In each iteration of the Ford-Fulkerson method, we find any augmenting path p and augment flow f along p by the residual capacity C(p). We update the net flow f [u, v] between each pair u, v of vertices that are connected by an edge in either direction, we assume implicitly that f [u, v] =0. The capacity C (u, v) = 0 if (u, v) E. The residual capacity C (u, v) = C (u, v) - f (u, v). Frist we initialize the flow f to 0. The while loop repeatedly finds an augmenting path p in G(f), and augments flow f along p by the residual capacity C(p). When no augmenting paths exist, the flow f is a maximum flow.

-Breadth-First search (BFS).

To visit all nodes connected to node K in a graph, we put K onto a FIFO queue. Then enter into a loop where we get the next node from the queue. If it has not been visited, visit it and push all the unvisited nodes on its adjacency list. We continue doing that until the queue is empty (Alon et al., 1987). We have used the Ford-Fulkerson method to find the maximum flow for the odd-sized HRs, which is related to the node-connectivity of the HRs.

4. RESULTS

The aim of this paper is to develop a software module to test the connectivity of various odd-sized HRs and attempted to answer an open question whether the node connectivity of an odd-sized HR is equal to its degree. We attempted to answer this question by explicitly testing the node connectivity's of various odd-sized HRs. We have used the Ford-Fulkerson method to find the maximum flow for the odd-sized HRs, which is related to the node-connectivity of the HRs.

In this paper, so far we have confirmed our hypothesis for the first 1,000 odd-sized HRs.

The following is some of our results for the maximum flow output for various odd-sized HRs.

- Maximum flow for 3-HR.

Number of edges = 3
Augment path: 0-2
Augment path: 0-1-2
Max flow = 2
Nodal degree = 2

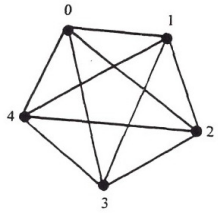

Figure 4. A 5-HR.

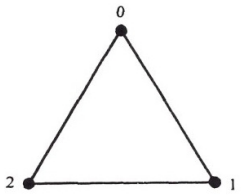

Figure 3. A 3-HR.

- Maximum flow for 5-HR.

Number of edges = 10
Augment path: 0-4
Augment path: 0-2-4
Augment path: 0-1-3-4
Max flow = 3
Nodal degree = 4
- Maximum flow for 11-HR

Number of edges = 44

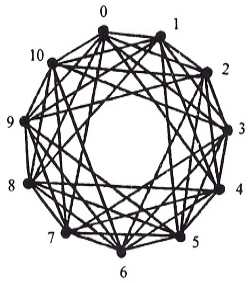

Figure 5. An 11-HR.

Augment path: 0-10
Augment path: 0-1-10
Augment path: 0-2-10
Augment path: 0-3-10
Augment path: 0-7-10
Augment path: 0-8-10
Augment path: 0-9-10
Augment path: 0-4-6-10
Maximum flow = 8
Nodal degree = 8

-Maximum flow for 29-HR

Number of edges = 116
Augment path: 0-1-28
Augment path: 0-2-3-28
Augment path: 0-4-26-28
Augment path: 0-8-23-28
Augment path: 0-28
Augment path: 0-27-28
Augment path: 0-25-28
Augment path: 0-21-20-28
Maximum flow = 8

-Maximum flow for 57-HR

Number of edges = 342
Augment path: 0-1-56
Augment path: 0-56
Augment path: 0-56
Augment path: 0-4-52-56
Augment path: 0-8-9-50-56
Augment path: 0-16-31-56
Augment path: 0-25-33-35-56
Augment path: 0-32-40-48-56
Augment path: 0-41-42-56
Augment path: 0-49-56
Augment path: 0-53-54-56
Augment path: 0-55-56
Maximum flow = 12

5. CONCLUSIONS

Although the Hyper-Cube is quite powerful from a computational point of view, there are some disadvantages to its use as an architecture for parallel computation. One of the most obvious disadvantages is that the node degree of the Hyper-Cube grows with its size. This means, for example, that processors designed for an N-node Hyper-Cube cannot later be used in a 2N-node Hyper-Cube. Moreover, the complexity of the communications portion of a node can become fairly large as N increases. For example, every node in a 1024-processor Hyper-Cube has 10 neighbors, and every node in a one million-processor Hyper-Cube has 20 neighbors.

In this paper we present hyper-rings (HRs for short).The number of links between processors in a HR is roughly twice that of a HC with the same number of processors, but the proposed organization possesses a number of advantages over that of a HC. Altman, et al., (1997) have shown that for any N we can construct a HR with N processors, whereas a Hyper-Cube must contain exactly 2n processors for some positive n.Although the number of edges in a hyper-ring is roughly twice that of a hypercube with the same number of nodes, the diameter and the connectivity of the hyper-ring are shorter and larger, respectively than those of the corresponding hypercube. These properties are advantageous to hyper-ring as desirable interconnection networks.

In this paper we developed a software module to test the connectivity of various odd-sized HRs and attempted to answer an open question whether the node connectivity of an odd-sized HR is equal to its degree. We attempted to answer this question by explicitly testing the node connectivity's of various odd-sized HRs. We have used the Ford-Fulkerson method to find the maximum flow for the odd-sized HRs, which is related to the node-connectivity of the HRs. In this paper, so far we have confirmed our hypothesis for the first 1,000 odd-sized HRs.

REFERENCES

[1] Sebek, M., Tonjes, R., and Kiss, Z (2016) "Complex Rotating Waves and Long Transients in a Ring - Network of Electrochemical Oscillators with Sparse Random Cross - Connections", arXiv:1501.02870v1 (2016).

[2] Fan, C., and Kenter, F., (2014) "Discrepancy inequalities for directed graphs , Discrete Applied Mathematics", 176 (2014) 30-42.

[3] Awad, I. Qatawneh, (2013), "Embedding Hex – Cells into Tree - Hypercube Networks", IJCSI International Journal of Computer Science, Vol. 10, Issue 3, No 2.

[4] Huo, H. Wei and Youzhi, X., (2010), "A Virtual Hypercube Routing Algorithm for Wireless Healthcare Networks", Chinese Journal of Electronics, Vol.19, No.1.

[5] Monton, E.and Hernandez, F., (2008). "Body area network for wireless patient monitoring", IET Communications, Vol.2, No.2, pp.215–222.

[6] Hong Luo, Yonghe L. and S.K. Das,(2007) "Routing correlated data in wireless sensor networks: A survey", IEEE Network, Vol.21, No.6, pp.40–47.

[7] Aly, Ashraf, (1999) "Reliable Communication Schemes in Hyper–rings Distributed Systems", Thesis/dissertation, Manuscript, Colorado University at Denver Library, OCLC Number: 44104731.

[8] Altman, T., (1997), "Reliable Schemes for Hyper – Rings, 28th International Southeastern Conference on Combinatorics", Graph theory and Computing; Boca Raton, Florida.

[9] Altman, T., Y. Igarashi and K. Obocata, (1994),"Hyper-Ring Connection Machines", Proc. IEEE 9th Annual. International Conference (1994) pp. 290-294.

[10] Altman, T., Y. Igarashi, and K. Obokata, (1995), "Hyper- Ring Connection Machines", parallel Computing 21, pp. 1327-1338.
[11] Aiello, B., and Leighton, F., (1991) “Coding Theory, Hypercube Embedding, and Fault Tolerance,” The 3rd Annual ACM Symposium on Parallel Algorithms and Architectures, pp.125-136.
[12] Alon, N., Barak, A., and Mauber, U., (1987) “On Disseminating Information Reliably without Broadcasting,” The 7th International Conference on Distributed Computing Systems, pp.74-81.

AUTHOR

Dr. ASHRAF ALY
Morehead state University
Morehead, KY 40351, USA