Corner-First Tree-Based Region Broadcasting in Mesh Networks

Hadeel Haddad and Muhammed Mudawwar

Computer Science Department
The American University in Cairo

Abstract

In direct interconnection networks, the collective communication operation one to all, which is usually referred to as broadcasting, can be generalized to allow one source node to send a message to a rectangular region of nodes, rather than to all nodes. Most of the proposed routing algorithms for direct mesh and torus networks use a broadcast tree of unicast messages. The minimum spanning tree-based region broadcasting is not deadlock free, unless the network is partitioned into many virtual sub-networks, where the number of virtual channels grows exponentially with the dimension of the network [3]. This paper proposes two versions of the minimum spanning tree region-broadcasting algorithm that are based on the idea of starting always at a corner of a region. The first algorithm uses always a fixed corner, while the second one uses the nearest corner. The two proposed algorithms are deadlock free and use virtual cut through for buffering blocked packets. Both broadcast algorithms can be safely mixed with unicast routing algorithms.

Keywords: Tree-based region broadcasting, minimum spanning tree, direct mesh networks, corner-first region broadcast algorithms, virtual cut through.

1. Introduction

Collective communication, such as broadcasting and multicasting, are very important for developing parallel programs. Interconnections networks are a critical component of distributed shared-memory multi-processors and message-passing multi-computers because parallel application performance is affected by their performance. Network performance is measured by network latency and throughput.

Most existing systems support only unicast communication in hardware. In these environments, broadcast must be implemented in software by sending multiple unicast messages. Sending a separate copy of the message from the source node to all nodes in the region may require excessive time. An alternative approach is to use a broadcast tree of unicast messages. In unicast based tree the source node actually sends the message to only a subset of the destination nodes. Each recipient of the message forwards it to some subset of the destination nodes that have not yet received it. This process continues until all destinations have received the message. An example of unicast based broadcast algorithm is the Recursive Doubling algorithm [1], [2].

Broadcast can be supported by hardware by replicating the message at intermediate routers. This intermediate reception capability allows a router to deliver an incoming message to the local host while simultaneously forwarding it to another router. An example of a tree based broadcast algorithm is the Minimum Spanning Tree MST for meshes [3], [4]. The MST algorithms are not deadlock free unless network is partitioned into virtual networks. Although network partitioning avoids deadlock, it has a major disadvantage. The required number of virtual sub-networks grows exponentially with the number of dimensions of the mesh [5], [10]. In addition, most proposed broadcast routing algorithms discuss the case where the source node is part of the broadcast region and ignore the case where the source node is outside the region. In this paper, two modified versions of MST broadcast algorithm are discussed, the Fixed Corner and the Nearest Corner minimum spanning tree. These two algorithms are deadlock free for mesh topology and can be mixed safely with unicast messages. The source node may reside inside or outside the broadcast region. The fixed corner is deadlock free even with one virtual channel class, while the nearest corner requires two virtual channel classes for direct mesh networks with arbitrary dimension.

2. Tree-Based Broadcast Algorithms

Broadcast algorithms are classified as being either tree-based or path-based. This paper discusses only on tree-based broadcast algorithms and does not consider the path-based ones.

2.1 The Minimum Spanning Tree (MST) Algorithm

One approach to multicast routing is to replicate a message and forward each copy on a different channel. The path followed by each copy may further branch in this manner until the message is delivered to all destination nodes. In such tree-based routing algorithm, the destination set is partitioned at the source, and separate copies are sent on one or more outgoing links. In this algorithm, a router receiving a message on an input channel is allowed to forward this message to all output channels along the higher dimensions as well as to an output channel along the same dimension and in the same direction. However, it is not allowed to route a message to an output channel along a lower dimension, or along the same dimension but in the opposite direction. Figure 1 represents a 4x4 2D Mesh, in which the source node is the shaded square. The source node injects four copies of the message, each for a different direction and dimension: +X, +Y, -X, and –Y. The copies routed in the X dimension are responsible of broadcasting the message to +X, +Y, -Y. While those copies routed to Y are only allowed to broadcast in the Y dimension.

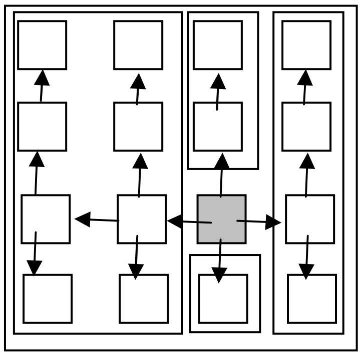


Figure 1: Tree Based Broadcast in a 4x4 Mesh

This routing algorithm is not deadlock free for wormhole routing unless the network is partitioned into disjoint virtual sub-networks. The disadvantage of this approach is that the number of virtual channels grows exponentially with number of dimensions of the network [5]. Also, this algorithm is not deadlock free for virtual cut through routing unless certain restrictions are applied [6], [7].

2.2 The Recursive Doubling (RD) Algorithm

In the RD algorithm, the source node first sends the message halfway across a dimension, partitioning the network into two sub-networks. In subsequent steps, each node holding a copy of the message forwards it to a node halfway across in its partition that has not yet received the message.
As with other unicast-based broadcast algorithms, the number of generated and injected messages in the RD algorithm is equal to the number of nodes in a region. Another disadvantage is that the generated messages might pass by other nodes that did not receive the message yet on their way to their destinations; however, the message is not delivered to these other nodes until later. The RD algorithm is deadlock free and uses dimension-order routing to send messages.

2.3 The Safe Minimum Spanning Tree Algorithm

The safe minimum spanning tree algorithm is a modified version of the MST algorithm that works when the buffer size is greater or equal to message length (i.e., virtual cut through buffering is used), when resources are reserved in order, and when a header flit is not allowed to advance to an output buffer until all required resource buffers are allocated. This algorithm was introduced in [6] and was shown to be deadlock free.

3. Fixed Corner Broadcast Algorithm

A broadcast region is defined by two opposite node corners. Whether the broadcasting source node resides inside or outside the broadcast region, a message is sent first, to a fixed corner, which can be the lower left corner of the region (fixed in direction not in location). Deterministic unicast routing can be used for this purpose. After reaching the lower left corner, the message will be broadcasted to the region by replicating the message and sending it to all other routers in the tree according to the dimension-order rule. A router in dimension i is allowed to send a message to all routers in dimensions $\geq i$, until the message reaches all destinations. Turns, such as turning back in opposite direction and changing from a higher dimension to a lower one, may occur and they are only allowed to occur at a router where broadcasting starts (in our case is the lower left corner of the region). These turns may cause deadlock if they are not restricted to a fixed corner, by fixing the corner all turning messages are turning in the same direction (restricted turns) and this will prevent deadlock. That is why a fixed corner is chosen to start broadcasting. By choosing the fixed corner to be the lower left, the restricted turns, which are allowed in this algorithm at the fixed corner and for a 2D Mesh direct network are:

Turning back to opposite direction:
-Y to +Y and -X to +X

Turning from a higher to a lower dimension:
-Y to +X and +Y to +X

3.1 Deadlock Formation

Deadlock may occur if any of the following conditions is violated:

a. All broadcast messages should be routed in a fixed direction along each dimension *when they broadcast to a region*, for example +X and +Y for 2D mesh networks. This restriction is necessary to prevent deadlock formation caused by *turning back* or by *turning from a higher dimension to a lower dimension* at a corner router where broadcasting starts. Since all broadcast messages are routed in the same direction then deadlock is prevented. Turning back and changing to a lower dimension, which are not allowed neither in DOR unicasting nor in MST broadcasting, are only allowed when broadcast messages reach the corner where to start broadcasting to a region. Violating this condition causes deadlock as shown below.

Allowing messages to broadcast from different region corners causes deadlock. Messages will then broadcast in opposite directions along the same dimension. Turning back occurs depending on the location of the corner node in correspondence with the source node. This turning back causes deadlock when it is in a different direction along the same dimension, as shown in Figure 2. By fixing the corner where to start broadcasting, possible turns will be in the same direction along each dimension and messages can follow each other without causing any deadlock.

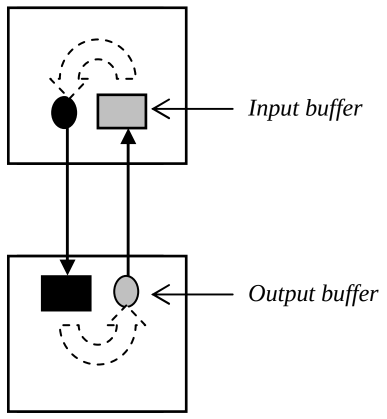


Figure 2: Deadlock caused by turning back in different directions along same dimension

Changing dimension from a higher to a lower one also causes deadlock. However, when broadcasting from a fixed corner, such as the lower left, changing to a lower dimension is restricted to the positive direction only. This restriction prevents deadlock formation.

b. Output buffer size should be greater or equal to message length. Wormhole routing in MST is not deadlock free. If one branch of a MST tree becomes blocked, then all other branches will become blocked as well. To solve this problem, we may use virtual cut-through buffering and make the size of output buffers big enough to store the largest packet. Storing a blocked message should be at the output buffer rather than at the input buffer. Otherwise, deadlock may occur, as shown in Figure 3.

At router A, the header of message 1 (H1) is blocked at +Y output buffer, while allocating and reaching buffers in routes B and D. H1 is waiting for H3 at +Y output buffer of router A. H3 is waiting for H2 at +X output buffer of router C. H2 is waiting for +Y output buffer of router D, which is allocated to message 1.
The problem is that the tail of message 1, which is still at the +X input buffer of router A, cannot reach the +Y output buffer of router D and free the allocated buffers on its way. To advance the tail of message 1 to the +X output buffer of router A, we should be able to advance it as well to its +Y output buffer. If the output buffer is small then the tail of message 1 is stuck at the +X input buffer of router A, which causes a deadlock. Observe that large input buffers do not solve the problem. For example, the input X+ buffer of router D is holding completely message 2, but this did not solve the problem. Therefore, we should insure that output buffers are large enough to hold complete packets, while input buffers can be minimized.

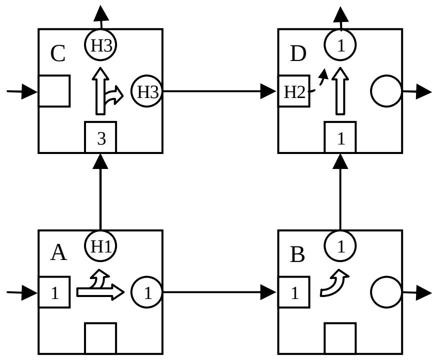


Figure 3: Deadlock caused by small output buffer size

c. Output buffer resources should be reserved in order. When more than one output buffer resource is required then buffer resources can be reserved gradually and should be reserved according to a given order. Ejection should be the last buffer to allocate. Violation of this condition causes deadlock as shown in Figure 4. Message 1 and 2 would like to allocate the same two output buffers in the +X and +Y directions. Message 1 has allocated the +X output buffer and is waiting for the +Y buffer, while message 2 has allocated the +Y buffer and is waiting for the +X buffer. If output buffers are allocated in order, such that the +X buffer is always allocated before the +Y buffer then this deadlock situation would not have happened.

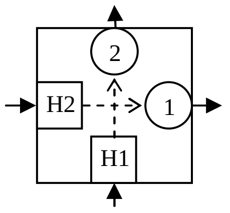


Figure 4: Deadlock caused by unordered reservation

4. Nearest Corner MST Broadcast Algorithm

Nearest Corner is a tree based algorithm, in which the source node may reside inside or outside the broadcast region. The broadcast message is first unicast to *nearest corner* of the broadcast region. Then it will be sent to other routers in the region according to the minimum spanning tree algorithm. Since the broadcasting corner is not fixed, the turns are not restricted. Therefore, deadlock may occur. Instead of fixing the corner, virtual channel buffers are used to avoid deadlock. Two disjoint virtual channel buffer classes are sufficient to avoid deadlock in direct mesh networks with arbitrary dimension. One deterministic buffer class and a second adaptive buffer class when turns are violated. Buffer allocation is done when all required output buffer resources are available. Reserving buffers in a specific order caused deadlocks.

4.1 Deadlock Formation

Deadlock may occur if any of the following conditions is violated:

***a. Two disjoint virtual channel classes are needed*:** One is used for deterministic routing, and the other is used for unrestricted turns. When a message starts using the non-deterministic virtual channel class, it will continue using it until it is ejected. Otherwise, deadlocks may occur.

b. Output buffer size should be greater or equal to message length: This situation is exactly the same as discussed in the fixed corner algorithm.

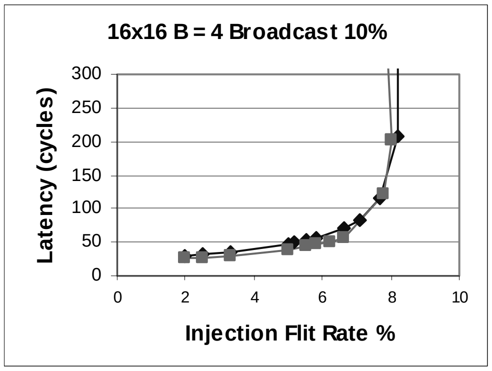
c. Buffer resources should not be reserved: Reserving buffer resources in a specific order resulted in deadlock formation. Therefore, we do not allocate output buffers unless all required buffer resources are available. Otherwise, no buffer resource is reserved.

5. Network Simulation

To measure the performance of broadcast algorithms, a number of mesh networks have been simulated varying few parameters in every run. The simulator is a C++ program that simulates mesh networks at the flit level. A flit transfers over the channel is assumed to take place in one cycle. The simulator can be configured to support different parameters and can generate various statistics.
Latency is measured starting from the generation time of a message until the tail is ejected at the destination node. Traffic is measured as the percentage of channel utilization. A channel is utilized during a clock cycle if it is used to transfer a flit successfully. The injection rate of a node is the percentage of cycles used to inject a flit successfully into the network. The ejection rate is the percentage of cycles used to eject a flit successfully from the network. The average traffic, injection, ejection rates are taken over all channels and nodes in the network.

5.1 Network Performance using Fixed Corner and Nearest Corner Algorithms

Two medium sized networks are simulated, a 2D 16x16 mesh and 4D 4x4x4x4 mesh. Simulation of both networks repeated with different number of virtual channel buffers, 1, 4 and 8 for the fixed corner algorithm and 4 and 8 for the nearest corner algorithm. Experiments were conducted for networks with a low probability, 10%, of broadcast messages and with a high probability, 90%, of broadcast messages.
The performance of 2D and 4D meshes is shown in Figures 5 through 8. The results show that the nearest corner algorithm is performing better as the average rate of injection and ejection is higher while network traffic is lower for almost the same number of routed messages. The nearest corner algorithm results in lower latencies and higher throughput saturation point.



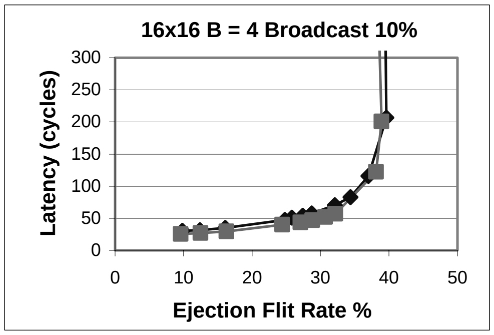
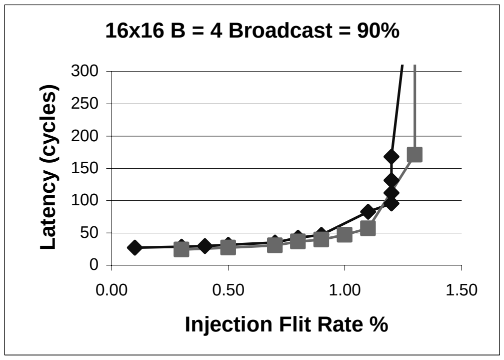


Fig 5: 2D Mesh 16 x 16, VCB = 4 Broadcast = 10%



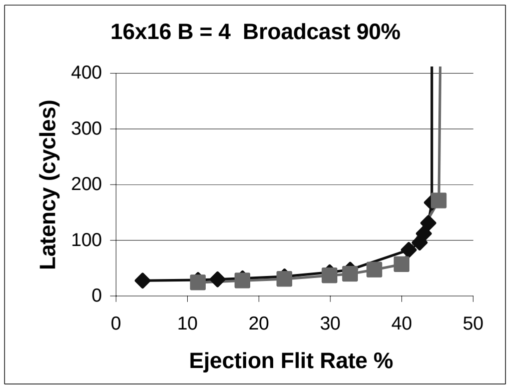
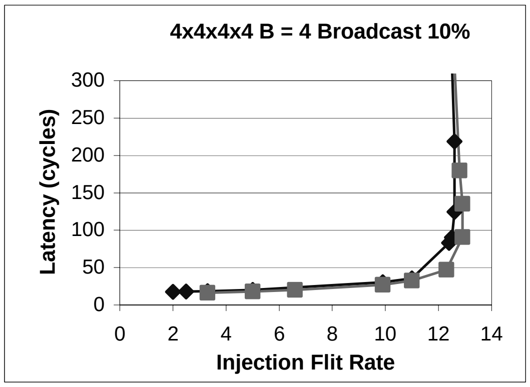


Fig 6: 16x16 2D Mesh, VCB = 4, Broadcast = 90%



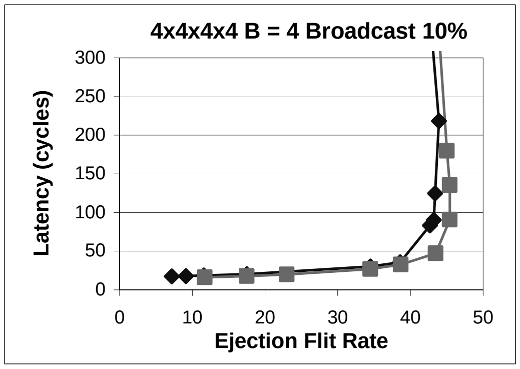
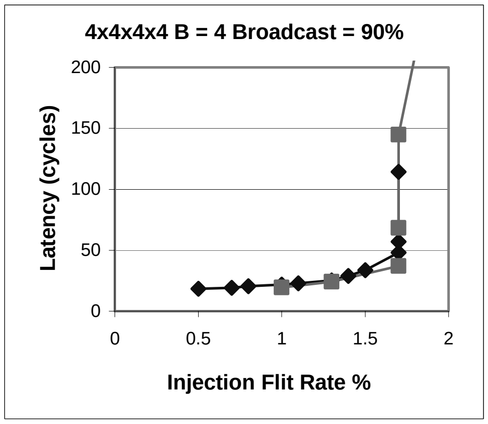


Fig 7: 4x4x4x4 Mesh VCB = 4 Broadcast = 10%



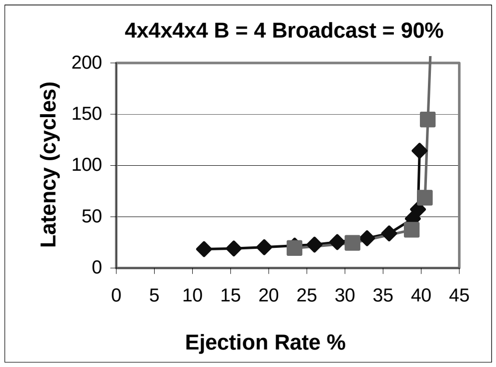
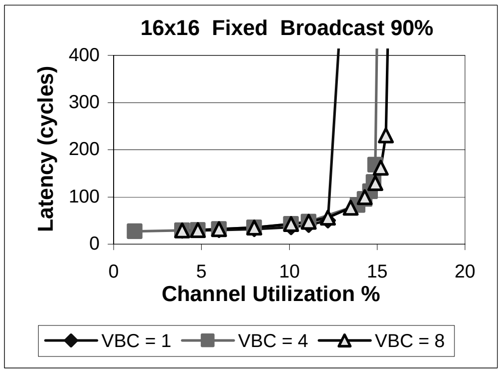
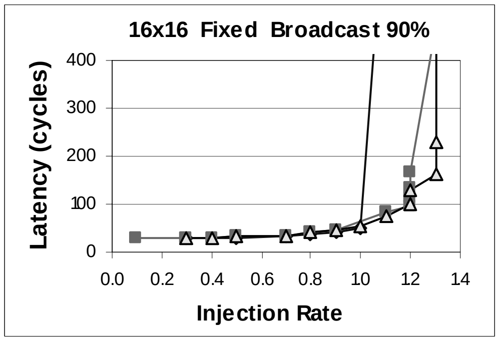


Fig 8: 4x4x4x4 Mesh, VCB = 4 Broadcast = 10%

The performance of applying both algorithms on a 2D mesh network using different number of virtual channel buffers are shown in figures 9 and 10. Increasing the number of virtual channels from 1 to 4 improved the network performance and saturation point significantly. However, increasing this number from 4 to 8 had little effect on network performance.





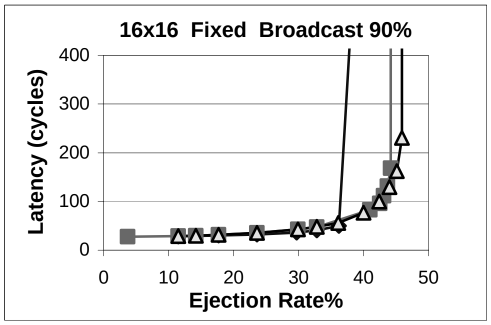
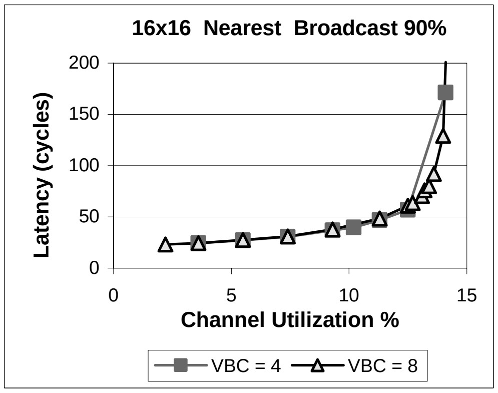
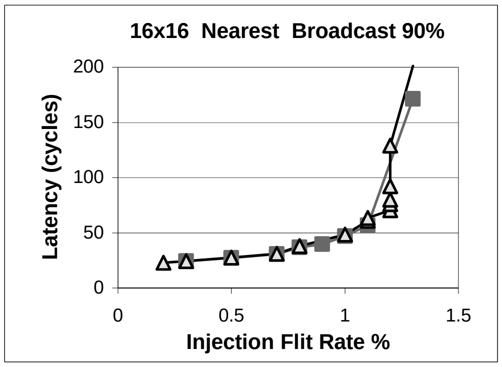


Fig 9: 16x16 Mesh, Fixed-corner, Broadcast = 90%





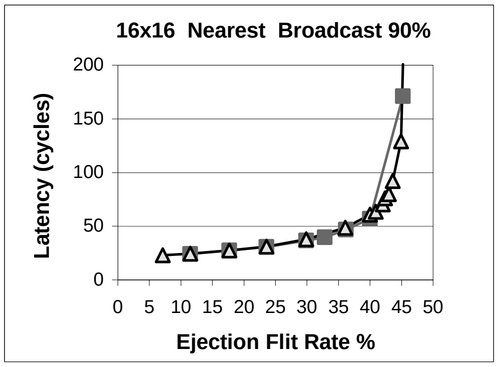


Fig 10: 16x16 Mesh, Nearest-corner, Broadcast = 90%

6. Discussion and Conclusions

From the above simulation results, we can conclude that fixed-corner and nearest-corner region broadcast algorithms perform equally well. The nearest corner algorithm performs slightly better than the fixed corner algorithm. However, this slight improved performance comes at an additional cost. The nearest corner algorithm requires two separate channel classes to avoid deadlock. However, the fixed corner algorithm is very simple. It does not require any buffer classes and is deadlock free, which means that it can be implemented efficiently in router chips. Region broadcasting is not restricted only to source nodes inside the region. Source nodes can be also outside the broadcast region. The low channel utilization, which reached only 15% for high broadcasting rate (90%) indicates the importance of implementing broadcasting in the hardware. Software-based broadcasting algorithms, which use unicast messages to implement broadcasting, require a high channel utilization, which affects the performance of unicast messages. On the other hand, hardware-based broadcast algorithms make better use of hardware resources.

References

[1] Y. J Tsai and P.K. Mckinley, An Extended Dominating Node Approach to Broadcast and Global Combine in Multiport Wormhole-Routed Mesh Networks, *IEEE Transactions on Parallel and Distributed Systems,* vol. 8, No. 1, pp. 41-57, January 1997.

[2] Y. J Tsai and P.K. Mckinley, A Broadcast Algorithm for All-Port Wormhole-Routed Torus Networks, *IEEE Transactions on Parallel and Distributed Systems,* Vol. 7, No. 8, pp. 876-885, August 1996.

[3] J. Duato, S. Yalamanchili and L. Ni, Interconnection Networks: An Engineering Approach, *IEEE Computer Society Press,* 1997.

[4] X. Lin and L. M. Ni, Deadlock-Free Multicast Wormhole Routing in Multicomputer Networks, *Proceedings of the 18th International Conference on computer Architecture,* pp. 116-126, May 1991.

[5] X. Lin, P. K. Mckinley, and L. M. Ni, Performance Evaluation of Multicast Wormhole Routing in 2D-Mesh Multicomputers, *Proceedings of the 1991 International Conference on Parallel Processing,* vol. 1, pp. 435-442, August 1991.

[6] M. F. Mudawwar and R. Mameesh, Region Broadcasting in *k*-ary *m*-way Networks, *Proceedings of the ISCA 13th International Conference on Parallel and Distributed Computing Systems,* August 8-10, 2000, Las Vegas, Nevada, pages 268-274

[7] R. Mameesh, Region Broadcasting in Multiway Channel Networks, *Master Thesis, Computer Science Department, The Am,erican University in Cairo,* January 2000.

[8] M. Barnett, D. G. Payne, and R. van de Geijn, Optimal Broadcasting in Mesh-Connected architectures, *Tech. Rep, TR-91-38, Department of Computer Science, The University of Texas at Austin,* Dec. 1991.

[9] P. K. Mckinley, Y. J Tsai, and D. F. Robinson, A Survey of Collective Communication in Wormhole-Routed Massively Parallel Computers, *Technical Report MSU-CPS-94-35,* June 1994.

[10] S. J. Johnsson and C. T. Ho, Optimum Broadcasting and personalized Communication in Hypercubes, *IEEE Transactions on Computers,* vol. C-38, pp. 1249-1268, Sept. 1989.