

On High Performance Multicast Algorithms for Interconnection Networks

Ahmed Al-Dubai¹, Mohamed Ould-Khaoua², and Imed Romdhani³

^{1,3} School of Computing, Napier University, 10 Colinton Road
Edinburgh, EH10 5DT, UK

{a.al-dubai, i.romdhani}@napier.ac.uk

² Department of Computing Science, University of Glasgow
Glasgow G12 8RZ, UK
mohamed@dcs.gla.ac.uk

Abstract. Although multicast communication, in interconnection networks has been a major avenue for a lot of research works found in literature, there are several key issues that should still be improved in order to meet the increasing demand for service quality in such systems. Apparently, most of the related works handle multicast communication within limited operating conditions such as low traffic load, specific network sizes and limited destination nodes. However, this paper investigates the multicast communication under different scenarios. It presents a comparison study of some well known multicast algorithms proposed for wormhole switched interconnection networks. Unlike the previous studies, this paper considers the multicast latency at both network and node levels. Performance evaluation results show that our proposed algorithm can greatly improve the performance of multicast operation.

Keywords: Wormhole Switched Networks, Path-based Multicast, Routing Algorithms.

1 Introduction

Multicast communication, in which a source node sends the same message to an arbitrary number of destination nodes in the network, is one of the most useful collective communication operations [1, 2, 3, 6]. Due to its extensive use, efficient multicast is critical to the overall performance of interconnection networks [1, 2, 3, 4, 14]. For instance, multicast is frequently used by many important applications such as parallel search and parallel graph algorithms [3, 14]. Furthermore, multicast is fundamental to the implementation of higher-level communication operations such as gossip, gather, and barrier synchronisation [1, 2, 4]. Ensuring a scalable implementation of a wide variety of parallel applications necessitates efficient implementation of multicast communication. In general, the literature outlines three main approaches to deal with the multicast problem: unicast-based [1, 3], tree-based [3, 13, 17] and path-based [2, 3, 8, 14, 15]. A number of studies have shown that *path-based* algorithms exhibit superior performance characteristics over their unicast-based and tree-based counterparts [2, 14, 15]. In path-based multicast, when the units (called flits in wormhole switched networks) of a message reach one of the

destination nodes in the multicast group, they are copied to local memory while they continue to flow through the node to reach the other destinations [2, 3, 8]. The message is removed from the network when it reaches the last destination in the multicast group. Although many interconnection networks have been studied [3], and indeed deployed in practice, none has proved clearly superior in all roles, since the communication requirements of different applications vary widely. Nevertheless, *n*-dimensional *wormhole switched* meshes have undoubtedly been the most popular interconnection network used in practice [2, 3, 5, 6, 10, 11] due to their desirable topological properties including ease of implementation, modularity, low diameter, and ability to exploit locality exhibited by many parallel applications [3]. In wormhole switching, a message is divided into elementary units called flits, each of a few bytes for transmission and flow control. The *header* flit (containing routing information) governs the route and the remaining data flits follow it in a pipelined fashion. If a channel transmits the header of a message, it must transmit all the remaining flits of the same message before transmitting flits of another message. When the header is blocked the data flits are blocked in-situ. Meshes are suited to a variety of applications including matrix computation, image processing and problems whose task graphs can be embedded naturally into the topology [3, 6, 10]. Meshes have been used in a number of real parallel machines including the Intel Paragon, MIT J-machine, Cray T3D, T3E, Caltech Mosaic, Intel Touchstone Delta, Stanford DASH [3]. Recently, among commercial multicomputers and research prototypes, Alpha 21364's multiple processors network and IBM Blue Gene uses a 3D mesh. In addition, a mesh has been recently the topology of choice for many high-performance parallel systems and local area networks such as Myrinet-based LANs.

The rest of the paper is organised as follows. Section 2 outlines some related works; Section 3 accommodates the proposed multicast algorithm. Section 4 conducts extensive analysis and simulation experiments and Section 5 summarises this work.

2 Background and Motivation

In general, existing multicast communication algorithms rely on two main strategies. In view of the dominance of the start-up time in the overall multicast latency, algorithms in the first class try to reduce the number of start-ups required to perform multicast, but this has been shown to be inefficient under high traffic loads [8, 10, 14]. For instance, the Dual Path (DP) and Multi Path (MP) algorithms proposed in [10] use this strategy. Briefly, DP uses at most two copies of the multicast message to cover the destination nodes, which are grouped into two disjoint sub-groups. This may decrease the path length for some multicast messages. The MP algorithm attempts to reduce path lengths by using up to four copies (or $2n$ for the *n*-dimensional mesh) of the multicast message. As per the multi-path multicast algorithm, all the destinations of the multicast message are grouped into four disjoint subsets such that all the destinations in a subset are in one of the four quadrants when source is viewed as the origin. Copies of the message are routed using dual-path routing (see [10] for a complete description). Algorithms in the second class, on the other hand, tend to use shorter paths, but messages can then suffer from higher latencies due to the number of start-ups required [15]. Based on this strategy, for

example, the Column Path (CP) algorithm presented in [15] partitions the set of destinations into at most $2k$ subsets (e.g. k is the number of columns in the mesh), such that there are at most two messages directed to each column.

Generally, most existing path-based algorithms incur high multicast latency. This is due to the use of long paths required to cover the groups serially like algorithms under the umbrella of the first multicast approach or those of the second category, in which an excessive number of start-ups is involved. In addition, a common problem associated with most existing multicast algorithms is that they can overload the selected multicast path and hence cause traffic congestion. This is mainly because most existing grouping schemes [8, 10, 15] do not consider the issue of load balancing during a multicast operation. More importantly, existing multicast algorithms have been designed with a consideration paid only to the multicast latency at the network level, resulting in an erratic variation of the message arrival times at the destination nodes. As a consequence, some parallel applications cannot be performed efficiently using these algorithms, especially those applications which are sensitive to variations in the message delivery times at the nodes involved in the multicast operation. Thus, our objective here is to propose a new multicast algorithm that can overcome the limitations of existing algorithms and thus leading to improve the performance of multicast communication in mesh networks. In a previous work [2], a new multicast scheme, the Qualified Group (QG) has been proposed for symmetric meshes. Such a scheme has been studied under restricted operating conditions, such as specific traffic load, fixed network sizes and a limited number of destination nodes [2]. In the context of the issues discussed above, this paper makes two major contributions. Firstly, the QG is generalised here with the aim of handling multicast communication in symmetric, asymmetric and different network sizes. Secondly, unlike many previous works, this study considers the issue of multicast latency at both the network and node levels across different traffic scenarios.

3 The Qualified Groups (QG) Algorithm

In an attempt to avoid the problems of existing multicast algorithms, this section presents the Qualified Group (QG) path-based multicast algorithm. The QG algorithm takes advantage of the partitionable structure of the mesh to divide the destination nodes into several groups of comparable sizes in order to balance the traffic load among these groups, which leads to avoid the congestion problem in the network. The groups, in turn, implement multicast independently in a parallel fashion, which results in reducing the overall communication latency. In general, the proposed algorithm is composed of four phases which are described below. For the sake of the present discussion and for illustration in the diagrams, we will assume that messages are routed inside the network according to dimension order routing [3, 10].

Definition 1. Consider a mesh (V, E) , with node set V and edge set E , a multicast set is a couple (p, \mathcal{D}) , where $p \in V$, $\mathcal{D} = \{p_1, p_2, \dots, p_k\}$ and $p_i \in V, i = 1, \dots, k$. The node p is the source of the multicast message, and the k nodes in \mathcal{D} are the

destinations. To perform a multicast operation, node p disseminates copies of the same message to all the destinations in \mathcal{D} .

We have adopted the dimension order routing due to the fact that this form of routing is simple and deadlock and livelock free, resulting in a faster and more compact router when the algorithm implemented in hardware, [3, 15]. However the QG algorithm can be used along any other underlying routing scheme, including the well-known Turn model and Duato's adaptive algorithms [3]. This is because the grouping scheme, as explained below, in QG can be implemented irrespective of the underlying routing scheme (in the algorithmic level), which is not the case in most existing multicast algorithms in which destination nodes are divided based on the underlying routing used (in the routing level) [8, 10, 15]. It is worth mentioning that such a research line will be investigated further in our future works.

Phase 1. In this phase, a multicast area is defined as the smallest n -dimensional array that includes the source of the multicast message as well as the set of destinations. The purpose of defining this area is to confine a boundary of network resources that need to be employed during the multicast operation.

Definition 2. In the n -dimensional mesh with a multicast set (p, \mathcal{D}) , a multicast area G_{MA} includes the source node $p[d_1, d_2, \dots, d_n]$ and destination nodes $\mathcal{D}[(d_1, d_2, \dots, d_n)]$ such that $\forall d_i \in \{d_1, d_2, \dots, d_n\}$, has two corners, upper corner $u_{d_i} = \max(\mathcal{D}[d_i], p[d_i])$ and lower corner $l_{d_i} = \min(\mathcal{D}[d_i], p[d_i])$ such

$$\text{that } mid_{d_i} = \begin{cases} (l_{d_i} + u_{d_i})/2 & \text{if } (l_{d_i} + u_{d_i}) \text{ is even} \\ ((l_{d_i} + u_{d_i}) - 1)/2 & \text{if } (l_{d_i} + u_{d_i}) \text{ is odd} \end{cases}$$

Phase 2. The multicast area G_{MA} is then divided into groups. The objective behind grouping the destination nodes is to distribute the traffic load over the multicast area in order to avoid traffic congestion, which contributes significantly to the blocking latency. Besides, grouping enables the destination nodes to receive the multicast message in comparable arrival times; i.e., this helps to keep the variance of the arrival times among the destination nodes to a minimum.

Definition 3. In an n -dimensional mesh with a multicast set (p, \mathcal{D}) , a divisor dimension Div_{d_i} for \mathcal{D} satisfies the following condition

$$Div_{d_i} = \min(N_{d_1}, N_{d_2}, \dots, N_{d_n}), N_{d_i} = \left| \mathcal{D}[d_i^{\uparrow}] - \mathcal{D}[d_i^{\downarrow}] \right| :$$

$$\mathcal{D}[d_i^{\uparrow}] = \sum_{mid_{d_i}}^{u_{d_i}} \mathcal{D}[d_i] \text{ and } \mathcal{D}[d_i^{\downarrow}] = \sum_{l_{d_i}}^{mid_{d_i}-1} \mathcal{D}[d_i]$$

Notice that if $N_{d_1} = N_{d_2}$, d_1 is given a higher priority, i.e., a higher priority is given based on the ascending order of the dimensions. For instance, if $N_x = N_y = N_z$, X dimension will be considered as a divisor dimension. The divisor dimension is used as a major axis for the grouping scheme in this phase. The multicast area G_{MA} is then divided into a number of disjoint groups as formulated in the following definition.

Definition 4. Given an n -dimensional mesh with a multicast set (p, \mathcal{D}) and a multicast area G_{MA} , $\forall G_i, G_j : G_i \subseteq G_{MA}$ and $G_j \subseteq G_{MA} \rightarrow G_i \cap G_j = \Phi$.

According to Definition 4, G_{MA} is divided into a number of primary groups as given in equation 1; where g_{pr} refers to the number of primary groups obtained after dividing the destination nodes over the division dimension, such that

$$g_{pr} = \begin{cases} p_t & \text{if } \exists G_i \subseteq G_{MA} : G_i = \Phi \\ 2^n & \text{otherwise} \end{cases} \tag{1}$$

where p_t is an integer, $1 \leq p_t < 2^n$

Phase 3. This phase is responsible for qualifying the groups already obtained in the preceding phase for a final grouping. Having obtained the primary groups, g_{pr} , we recursively find the multicast area for each group, $G_i \subseteq G_{MA}$, as defined in Definition 4, and determine the internal distance $Int(G_i)$ for each group G_i .

$$Int(G_i) = Dist(p_f(G_i), p_n(G_i)) + N_{G_i} \tag{2}$$

Where $Dist$ refers to the Manhattan distance in which the distance between two nodes, for instance the distance between two nodes $(p1_x, p1_y)$ and $(p2_x, p2_y)$ is given by $Dist(p1, p2) = |(p1_x - p2_x) + (p1_y - p2_y)|$. While the first term, $Dist(p_f(G_i), p_n(G_i))$, in the above equation represents the distance between the farthest p_f and the nearest node p_n in a group G_i from/to the source node p , respectively, the second term, N_{G_i} , represents the number of destination nodes that belong to the relevant group $G_i \subseteq G_{MA}$. We then determine the external distance $Ext(G_i)$.

$$Ext(G_i) = Dist(p_n(G_i), p) \tag{3}$$

The minimum weight W_m for a group $G_i, 1 < i \leq g_{pr}$, where g_{pr} refers to the number of primary groups, is then calculated by

$$W_m(G_i) = Ext(G_i) + Int(G_i) \tag{4}$$

Definition 5. Given a multicast area G_{MA} and $G_i \subseteq G_{MA}$, where $1 < i \leq g_{pr}$, the average of the minimum weights W_{av} , for the multicast area G_{MA} , is given by

$$W_{av} = \frac{\sum_{i=1}^{g_{pr}} W_m(G_i)}{g_{pr}} \tag{5}$$

Definition 6. Given a multicast area G_{MA} , $G_i \subseteq G_{MA}$, and W_{av} , the qualification point, $QP(G_i)$, for each group is calculated as follows

$$QP(G_i) = \frac{(W_m(G_i) - W_{av})}{W_{av}} \tag{6}$$

The qualification point for each group is compared to an assumed threshold value TD , which is used to set a limit for the partitioning process.

Definition 7. Given a multicast area G_{MA} and $G_i \subseteq G_{MA}$, we say that G_i is a qualified group if and only if its minimum weight $W_m(G_i) \leq W_{av}$ or if its qualification point $(QP(G_i)) \leq TD$.

For example, given that the threshold value is $TD = 0.5$, each qualified group must hold at least half of the total average weight W_{av} of the groups. Once a group $G_i \subseteq G_{MA}$ does not satisfy the condition formulated in Definition 7, it is treated as an unqualified group. In this case, this unqualified group is divided into two sub-groups based on its division dimension. If the new resulting groups are qualified the partitioning process is terminated. Otherwise, the unqualified group is divided into a number of sub-groups sb , where $2 \leq sb \leq 2^n$. For instance, for any unqualified group $G_i \subseteq G_{MA}$ in the 2D mesh, it can be divided into four groups at maximum, even if the new obtained groups are still larger than those which meet the qualification point. In fact, the partitioning process is terminated at this stage in order to reduce the number of comparisons during the qualifying phase. This helps to keep the algorithm simple and maintains a low preparation time.

Phase 4. For each group resulting from Phase 3, the nodes which have the lowest communication cost, in terms of distance from the source node, are selected as the representative nodes of the qualified groups that can receive the multicast message from the source node. In other words, the nearest node for each qualified group is

ected so that it could be sent the multicast message with a single start-up only. Concurrently, the representative nodes act as “source” nodes by delivering the message to the rest of the destination nodes in their own groups with one additional start-up time only. After qualifying all the groups, the source node sends the message to the representative nodes in the qualified groups. The source node performs this operation with a single start-up latency taking advantage of the multiple-port facility of the system by creating two disjoint paths in this step. Concurrently, every representative node in each group acts as a source node and, in turn, sends the message to the rest of the destinations in its own group.

4 Performance Evaluation

A number of simulation experiments have been conducted to analyse the performance of *QG* against *DP*, *MP* and *CP*. A simulation program has been developed to model the multicast operation in the mesh. The developed model has been added to a larger simulator called MultiSim [6], which has been designed to study the collective communication operations on multicomputers and has been widely used in the literature [2, 10, 12]. The simulation program was written in VC++ and built on top the event-driven CSIM-package [7]. We have used the 2D mesh with four injection channels and four ejection channels. Two unidirectional channels exist between each pair of neighbouring nodes. Each channel has a single queue of messages waiting for transmission. In our simulations, the start-up latency has been set at 33 cycles, the channel transmission time at 1 cycle and the threshold *TD* at 0.5. The network cycle time in the simulator is defined as the transmission time of a single flit across a channel. The preparation time (which consists of dividing the destination nodes into appropriate subsets and creating multiple copies of the message as needed, depending on the underlying algorithm) of the *DP*, *MP*, *CP* and *QG* algorithms are set at 2, 2, 4 and 16 cycles, respectively. The preparation time was deliberately set higher in the *QG* algorithm to reflect the fact that our algorithm requires a longer time to divide the destinations into qualified groups. All simulations were executed using 95% confidence intervals (when confidence interval was smaller than 5% of the mean). The technique used to calculate confidence intervals is called batch means analysis. In batch means method, a long run is divided into a set of fixed size batches, computing a separate sample mean for each batch, and using these batches to compute the grand mean and the confidence interval. In our simulations, the grand means are obtained along with several values, including confidence interval and relative errors which are not shown in the figures. Like existing studies [1, 2, 3, 10, 15, 13], only the grand mean is shown in our figures.

4.1 Latency at the Node Level

This section presents the coefficient of variation of the multicast latency as a new performance metric in order to reflect the degree of parallelism achieved by the multicast algorithms. A set of simulation experiments have been conducted where the message inter-arrival times between two messages generated at a source node is set at 250 cycles. The message length is fixed at 64 flits and the number of destination

nodes is varied from 20, 30, 40... to 60 nodes. The coefficient of variation (CV) is defined as SD / M_{nl} , where SD refers to the standard deviation of the multicast latency (which is also the message arrival times among the destination nodes) and M_{nl} is the mean multicast latency. The coefficient of variation of QG has been compared against that of DP , MP and CP . Table 1 contains performance results for the 16×16 mesh, which have been obtained by averaging values obtained from at least 40 experiments in each case. The $QG_{IMPR} \%$ in Table 1 refers to the percentage improvement obtained by QG over its DP , MP and CP competitors.

As shown in Table 1, QG achieves a significant improvement over DP , MP and CP . This is due firstly to the efficient grouping scheme adopted by QG which divides the destinations into groups of comparable sizes. Secondly, and more importantly, unlike in DP , MP and CP , the destination nodes for each qualified group in QG (except those selected in the first message-passing step) receive the multicast message in the second message-passing step, in parallel. This has the net effect of minimising the variance of the arrival times at the node level. In contrast, DP , MP and CP perform multicast with either longer paths as in DP and MP or in an excessive number of message-passing steps, as in CP .

Table 1. The coefficient of variation of the multicast latency in the DP , MP and CP algorithms with the improvement obtained by QG ($QG_{IMPR} \%$) in the 16×16 mesh

	#Destinations=20		# Destinations=40		# Destinations=60	
	CV	($QG_{IMPR} \%$)	CV	($QG_{IMPR} \%$)	CV	($QG_{IMPR} \%$)
DP	0.386	46.19	0.416	54.83	0.476	76.27
MP	0.326	23.48	0.365	35.69	0.420	55.56
CP	0.467	76.74	0.489	81.56	0.504	86.49
QG	CV= 0.2640		CV= 0.2695		CV= 0.27004	

4.2 Latency in the Presence of Multicast and Unicast Traffic

In some real parallel applications, a message may have to compete for network resources with other multicast messages or even with other unicast messages. To examine performance in such situation, results for the mean multicast latency have been gathered in the 10×10 mesh in the presence of both multicast (10%) and unicast (90%) traffic (similar studies are outlined in [8, 10, 15]). The message size is set at 64 flits and the number of destinations in a given multicast operation has been set to 10 and 20 nodes, respectively. The simulation results are provided in Figs. 1 and 2. Fig. 1 reports results for 10 destinations while Fig. 2 shows results for 20 destinations. Under light traffic, QG , DP and MP have comparable performance behaviour, with MP having a slightly lower latency. On the other hand, CP has a higher time.

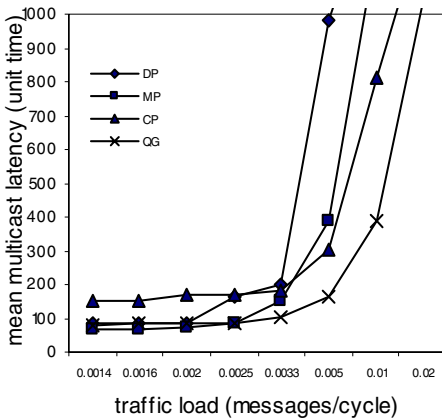


Fig. 1. Mean multicast latency in the 10×10 mesh. Message length is 64 flits, number of 10 destinations = 10 nodes, traffic consists of multicast (10%) and unicast (90%).

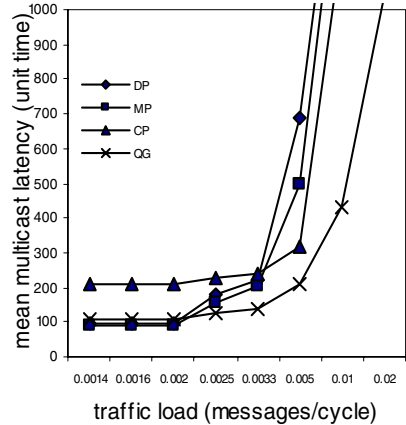


Fig. 2. Mean multicast latency in the 10×10 mesh. Message length is 64 flits, number of destination = 20 nodes, traffic consists of multicast (10%) and unicast (90%).

This is mainly due to the dominating effect of the start-up latency in such a situation. However, under heavy traffic, an opposite behaviour is noticed in that *QG* performs the best in terms of both latency and throughput, followed by *CP*. More importantly, we can observe from Fig. 2 that as the number of destinations increases the performance advantage of *QG* becomes more noticeable over that of *CP*. This is mainly because *QG* alleviates significantly the congestion problem at the source node. In contrast, the source node in *CP* suffers from a higher load and as more destinations are involved in the multicast operation, the more severe this limitation becomes.

5 Conclusions and Future Directions

In this study, the *QG* multicast algorithm has been evaluated under different scenarios and conditions. Results from extensive simulations under different conditions have revealed that the *QG* algorithm exhibits superior performance over well-known algorithms, such as dual-path, multiple-path, and column-path algorithms. Unlike existing multicast algorithms, the *QG* algorithm can maintain a lower variance of message arrival times at the node level. Consequently, most of the destination nodes receive the multicast message in comparable arrival times. It would be interesting to further investigate the interaction between the important parameters that affect the performance of the *QG* algorithm, notably the grouping scheme, network size, threshold value, multicast group size, and traffic load, with the aim of proposing an analytical model that could predict, for example, the multicast latency given a particular grouping scheme, network size, multicast group size, and traffic load.

References

- [1] Nen-Chung Wang, Cheng-Pang Yen and Chih-Ping Chu, Multicast communication in wormhole-routed symmetric networks with hamiltonian cycle model, *Journal of Systems Architecture*, vol.51, Issue 3 , March 2005, pp.165-183, 2005.
- [2] A. Al-Dubai, M. Ould-Khaoua and L. Mackenzie, An efficient path-based multicast algorithm for mesh networks, *Proc. the 17th Int. Parallel and Distributed Processing Symposium (IEEE/ACM-IPDPS)*, Nice, France, 22 -26 April, pp. 283-290, 2003.
- [3] J. Duato, C. Yalamanchili, L. Ni, *Interconnection networks: an engineering approach*, Elsevier Science, 2003.
- [4] Abderezak Touzene, Optimal all-ports collective communication algorithms for the k -ary n -cube interconnection networks , *Journal of Systems Architecture*, vol.50, Issue 4, pp. 169-236, 2004.
- [5] Yuh-Shyan Chen, Chao-Yu Chiang and Che-Yi Chen, Multi-node broadcasting in all-ported 3-D wormhole-routed torus using an aggregation-then-distribution strategy, *Journal of Systems Architecture*, vol.50, Issue 9 , pp. 575-589, 2004.
- [6] P. K. McKinley, C. Trefftz, MultiSim: A simulation tool for the study of large-scale multiprocessors, *Proceedings of the Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS' 1993)*, pp. 57-62. 1993.
- [7] H. D. Schwetman, CSIM: A C-based, process-oriented simulation language, Tech. Rep. pp. 80-85, *Microelectronics and Computer Technology Corp.*, 1985.
- [8] E. Fleury, P. Fraigniaud, Strategies for path-based multicasting in wormhole-routed meshes, *J. Parallel & Distributed Computing*, vol. 60, pp. 26-62, 1998.
- [9] Y.-C. Tseng, S.-Y. Wang, C.-W. Ho, Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach, *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 1, pp. 44-61, 1999.
- [10] X. Lin, P. McKinley, L.M. Ni, Deadlock-free multicast wormhole routing in 2D-mesh multicomputers, *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 8, pp. 793-804.1994.
- [11] S. Cang, J. Wu, Time-step optimal broadcasting in 3-D meshes with minimal total communication distance, *J. Parallel & Distributed Computing*, vol. 60, pp. 966-997, 2000.
- [12] D. F. Robinson, P. K. McKinley, C. Cheng, Path based multicast communication in wormhole routed unidirectional torus networks, *Journal of Parallel Distributed Computing*, vol. 45, 104 - 121, 1997.
- [13] M. P. Malumbres, J. Duato, An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors, *J. Systems Architecture*, vol. 46, 1019-1032, 2000.
- [14] P. Mohapatra, V. Varavithya, A hardware multicast routing algorithm for two dimensional meshes, *the Eighth IEEE Symposium on Parallel and Distributed Processing*, pp. 198-205, News Orleans, October 1996.
- [15] R. V. Boppana, S. Chalasani, C.S Raghavendra, Resource deadlock and performance of wormhole multicast routing algorithms, *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 6, 535-549, 1998.
- [16] S. Wang, Y. Tseng, C. Shiu , J. Sheu, Balancing traffic load for multi-node multicast in a wormhole 2D torus/mesh, *the Computer Journal*, vol. 44, no. 5, pp. 354-367, 2001.
- [17] D. R. Kumar, W. A. Najjar, P. K. Srimani, *A new adaptive hardware tree-based multicast routing in k-ary n-cubes*, *IEEE Computer*, vol. 50, no. 7, pp. 647-659, 2001.