All-to-All Broadcast on Switch-Based Clusters of Workstations

Matt Jacunski, P. Sadayappan and D. K. Panda
Department of Computer and Information Science
The Ohio State University, Columbus, OH 43210
Phone: (614) 292-0053, Fax: (614) 292-2911
Email: {jacunski,saday,panda}@cis.ohio-state.edu

Abstract

This paper presents efficient all-to-all broadcast algorithms for arbitrary irregular networks with switch-based wormhole interconnection and unicast message passing. First, all-to-all broadcast is considered within a single switch cluster. Both combining and non-combining algorithms are compared via analytical modeling and simulation. The characteristics of optimal all-to-all broadcast operation are considered and applied to development of multi-switch algorithms. The single switch algorithms are considered on two switch clusters and a near-optimal algorithm is developed which schedules use of interconnecting links where the potential for link contention exists. Finally, the Link Scheduling concept is extended to handle arbitrary irregular networks. Operation of this algorithm is simulated on a 128-node irregular network, and shows a 27.1% improvement in performance compared to other algorithms.

1 Introduction

The availability of switch-based commodity gigabit networking technologies such as Myrinet [2] and the high performance of personal computers (and workstations) makes parallel computing over clusters a promising alternative to the use of custom designed parallel computers. An important issue to be addressed is the development of efficient implementations of portable parallel programming models such as the Message Passing Interface (MPI) [13] for switch-based clusters of workstations.

MPI defines primitives for point-to-point communication as well as various collective communication operations between processes. While considerable research has addressed the development of efficient algorithms for collective communication operations over regular networks such as meshes and hypercubes [1, 3, 4, 5, 8, 10, 11, 12, 14], much less work has been done for optimizing collective communication over irregular switch-based clusters [9, 15]. In this paper we address the collective communication operation of all-to-all broadcast (called All_Gather in MPI) for switch-based clusters with arbitrary topology.

The paper is organized as follows. In Section 2, all-to-all broadcast is considered within a single-switch cluster. Both combining and non-combining algorithms are compared via analytical modeling and simulation. Section 3 treats the two-switch case and develops a near-optimal algorithm. The general case of multiple switches with arbitrary topology is considered in Section 4. Conclusions are provided in Section 5.

2 All-to-all broadcast on a single switch

All-to-all broadcast among nodes connected to a single switch may be performed without link contention by a variety of methods. Three such algorithms are presented below. An extended version of this paper [7] provides an analytical comparison of these algorithms. Also, the complete irregular switch-based cut-through network model used for this paper and the associated issues related to deadlock-free and adaptive routing for such networks can be found in [7].

2.1 Logical Ring (LR) algorithm

This algorithm organizes all participating nodes in a logical ring by rank order. In the first step of the required $(P-1)$ steps, each node sends its message data to the subsequent node in the ring and receives a message from the preceding node in the ring. In all remaining steps, the message just received is forwarded to the following node and another message is received from the preceding node. This algorithm has the characteristic that all communication from any single node's perspective is with only two other nodes.

2.2 Simultaneous Broadcast (SB) algorithm

The SB algorithm performs P simultaneous broadcasts, one originating from each node. Each broadcast is performed by sending the node's local message data to each of the $(P-1)$ other nodes participating in the all-to-all operation. Each node will receive $(P-1)$ messages.

Because each node sends only its local data, sends may be performed asynchronously to receiving. Transmission may overlap startup costs for all steps after the first. Each node begins by sending messages consecutively to the destinations $(rank + step)$, checking after each send for the arrival of an incoming message. This continues until the

network can accept no more messages or an incoming message is detected. At this point, receives and sends begin alternating until all messages have been sent. The node then continues to receive until all required messages have been received.

2.3 Combining algorithm

The combining algorithm consists of $\lceil log_2 P \rceil$ steps during which every node sends a message and receives a message. The message to be sent consists of the local message data combined with message data received in previous steps. The size of the message sent and the message received doubles at each step (with the possible exception of the last step). The sum of the sizes of all messages sent or received by each node is $(P-1)m$. Beginning at $step = 0$, the address of the destination at each step is $(rank + 2^{step}) \bmod P$. The address of the source node at each step is $(rank - 2^{step}) \bmod P$.

3 All-to-all broadcast on 2-switch networks

In this section, we discuss the execution of all-to-all broadcast on networks consisting of two switches. The purpose of this is not that this is a common network topology. We discuss two switch networks to illustrate the issues affecting algorithm performance introduced with reduced connectivity (and therefore potential link contention) without the complexities of the general irregular networks explored in the next section.

Executing the all-to-all broadcast on a network consisting of two switches with a single interconnecting link introduces the potential for link contention. Communication patterns which approach optimal performance will have the following properties: (a) degree of pipelining of communication components are significant, (b) transmission latencies are minimal, and (c) node idleness is minimized. The combining and SB algorithms operate as described in the previous section. There is slight modification to the ring algorithm and we introduce a new algorithm which is a modification of the SB algorithm.

3.1 Switch-Ordered Ring (SO-R) algorithm

The operation of the ring algorithm remains essentially the same as for the single switch case except for ordering of the nodes by switch to eliminate potential link contention. For either switch, only one local node sends to a node on the other switch and only one remote node receives from a local node. This communication pattern encounters no link contention and therefore is expected to perform about as well as the ring algorithm on a single switch. The only additional cost encountered is the additional switching time for the messages being sent between switches.

3.2 2-switch Link Scheduling (LS2) algorithm

We introduce a new algorithm, the 2-switch Link Scheduling (LS2) algorithm, which attempts to attain near-optimal performance as defined above. This algorithm is a special case of the more general Link Scheduling (LS) algorithm formally described in the next section. The basic idea is that use of the interconnecting link is scheduled among nodes in a way that permits every node to remain busy with useful work. The algorithm is described below for the two possible cases: the balanced case where there are the same number of nodes on each of the two switches, and the unbalanced case where there are unequal numbers of nodes on each switch.

In general, the algorithm executes in two phases. During the first phase, nodes take turns using the interconnecting link to transfer their message data to a node on the other switch. During the steps when it is not a node's turn to use the interconnecting link, it participates in a local all-to-all broadcast of the local message data. The second phase consists of performing another local all-to-all broadcast, this time distributing the remote data received during the first phase.

3.2.1 Balanced node distribution

Consider 2 switches with N nodes on each switch ($P = 2N$). These nodes are assigned an index, 1 to N, within the switch to which they belong. The algorithm completes in two phases. During the first phase, the N nodes on each switch perform an N-step modified SB all-to-all broadcast within each switch. A standard SB all-to-all operation takes $N-1$ steps. The modification is that at step number i, the node with index i does not participate in the local all-to-all but rather sends its local message across the interconnecting link to the node at index $(N-i)$ on the other switch (a remote node). Likewise, at step i, the node with index $(N-i)$ receives from the remote node with index i. In this way, at the end of the first phase the following has been accomplished: (a) a local switch all-to-all broadcast operation has completed, (b) all local switch message data has been transferred to remote nodes, and (c) all remote message data has been received by local nodes. Since each local node now holds exactly one remote message which must be distributed within the switch, the second phase is simply a local (within-switch) SB all-to-all broadcast operation, distributing the data received in the first phase. The second phase completes in $N-1$ steps.

3.2.2 Unbalanced node distribution

Consider 2 switches, one with N nodes attached and the other with N^* nodes attached where $N > N^*$ and $N + N^* = P$. The LS2 algorithm for unbalanced node distributions also has two phases. The first phase on the switch with fewer nodes, however, will have multiple stages.

For the switch with N nodes, the first phase consists of an N-step modified SB all-to-all broadcast. Each of the N steps involves one of the local nodes sending to a node in the other switch (a remote node). The target addresses of

these inter-switch sends is $(N^* - step) mod(N^*)$. A local node is involved in receiving from a remote node during only the first N^* steps of the first phase. After the N^* step of the first phase, all N^* remote messages have been received so the node scheduled to receive from outside the switch has nothing to receive and therefore continues with the next step. At the completion of the first phase, (a) a local switch all-to-all broadcast operation has completed, (b) all local message data has been transferred to some remote node, and (c) all remote message data has been received by N^* of the local nodes. In the second phase, each of the N^* nodes which received an inter-switch message performs a broadcast to all other local nodes. The second phase still requires $N-1$ steps (because there are $N-1$ destinations for each broadcast) but each node needs to receive a maximum of only N^* messages. Therefore, this second phase will require less time to complete than that for the balanced scenario.

For the switch with N^* nodes, the first phase consists of $\lceil \frac{N}{N^*} \rceil$ stages, each with N^* steps. During the first of these stages, the following occurs: (a) messages are sent by local nodes to remote nodes in a scheduled fashion as previously described, (b) the first N^* remote messages are received, one per local node, and (c) a modified SB all-to-all broadcast is executed, distributing the local messages locally. During each of the following stages of the first phase, modified SB all-to-all broadcasts are performed, distributing messages received from remote nodes in the previous stage while another N^* remote messages are received. This continues until a total of N messages have been received. The second phase consists of the distribution of the remote messages received in the final stage of the first phase.

3.2.3 Analysis and performance of LS2 algorithm

In the balanced case where the number of nodes on each of two switches is the same, the LS2 algorithm is near-optimal. That is, (a) communication components are overlapped for all possible steps except one, (b) all nodes remain busy with useful work during the entire execution of the algorithm, and (c) there is no unnecessary communication. The only point at which communication overlap does not occur is at the transition from the first phase to the second phase. Whichever node receives from the remote switch in the last step of the first phase must wait until receipt of that message is complete before beginning the second phase.

The performance degrades when the numbers of nodes on the two switches are unequal. The LS2 algorithm does not make optimal use of the interconnecting link when the node distribution is not balanced. Once there are no more messages to be sent from one switch, the link becomes idle in that direction. It is possible to improve the algorithm to make use of this link during the entire execution of the algorithm with improved results.

We consider two scenarios to illustrate the performance issues and possible solutions for all-to-all broadcast communication on 2 switches. Each consists of two 32-port switches, each with a single bidirectional interconnecting link. The balanced scenario divides the nodes equally between the switches, 16 on each. The unbalanced divides the nodes asymmetrically: 21 nodes on one switch and 11 on the other.

Algorithm	1-switch (reference)	Balanced 2-switch	Unbalanced 2-switch
Combining	139.69	249.16	593.14
SB	137.79	1079.47	1082.24
RO-Ring	205.78	1204.24	1088.22
SO-Ring	205.78	206.84	206.84
LS2	-	141.64	184.24

Table 1. Completion time in microseconds for balanced 2-switch network (16 nodes per switch) and unbalanced 2-switch network (11 nodes on one switch, 21 nodes on the other) for 256 flit messages and 5 microsecond startup time.

Table 1 shows the performance results for the two scenarios considered here. For these simulations, t_s (communication startup time) = 5 microsecond and the message length is 256 flits. Results for the single switch case with these parameters are given for reference. For these experiments, the following parameters were used: t_s (communication startup time) varied with experiments, t_{phy} (link propagation time = 16.0 nanoseconds, t_{route} (routing delay at switch = 500 nanoseconds, t_{sw} (switching time across the router crossbar for a flit) = 16.0 nanoseconds, t_{inj} (time to inject a flit into the network) = 16.0 nanoseconds, and t_{cons} (time to consume a flit from the network) = 16.0 nanoseconds.

4 All-to-all broadcast on irregular networks of arbitrary size

For irregular networks of arbitrary size and complexity, the potential for link contention increases. The rank ordering ring (RO-R) algorithm, the default implementation supplied by MPICH [6], becomes less attractive as the potential for link contention increases. Two algorithms which perform well for arbitrary irregular networks are discussed below.

4.1 Switch-Ordered Ring (SO-R) algorithm

For an irregular network, an efficient ordering of nodes for the ring algorithm is found by performing an in-order traversal of the BFS tree that was generated to determine network routing characteristics. Although this ordering does not eliminate potential link contention completely, the potential is greatly reduced.

The lists of nodes on each switch are concatenated in the order found traversing the BFS tree to determine the node ordering for the SO-R algorithm. This ordering has a simple implementation and provides good performance for arbitrary irregular networks. Performance of the SO-R algorithm for a 10-switch, 128 node irregular topology is shown in Table 2.

4.2 Link Scheduling (LS) algorithm

Here we extend the concepts of the LS2 algorithm presented in the previous section to handle arbitrary irregular networks. This algorithm permits an enhancement to the LS2 algorithm which improves performance.

The LS2 algorithm requires logically organizing the switches in the irregular network into a ring. Essentially, the LS algorithm involves nodes on each switch performing three tasks: (a) sending currently held data blocks to nodes on another switch (the remote destination switch), (b) distributing these currently held data blocks among the nodes on the local switch, and (c) receiving data blocks from another switch (the source switch). Since nodes on the local switch communicate only with nodes on two other switches, data blocks must be forwarded around the network until all non-local data blocks have been received at each switch. There is a steady stream of data blocks needed by nodes on the local switch flowing into the switch and a steady stream of data blocks required at the remote destination switch flowing out of the local switch.

The LS algorithm consists of a setup phase plus two operational phases. The setup phase entails determining the switch ordering by traversing the routing BFS tree. Each node finds its remote source and remote destination switch from this ordering. The logical organization of the switches may be thought of as a ring.

In the first phase, there are one or more stages. During each stage, a local modified SB all-to-all operation occurs while sending remote messages to nodes on the remote destination switch and receiving remote messages from nodes on the the remote source switch. The data distributed by the modified local all-to-all broadcast is the local data in the first stage and the data received in the previous stage thereafter. Figure 1 shows the communication pattern for a particular switch in a 15-node irregular network with 4 local nodes, 5 nodes on the remote source switch, and 3 nodes on the remote destination switch. Phase 1 completes in a number of steps which is a multiple of the number of local nodes and when all required remote messages have been sent and received. To maintain a flow of only the required data blocks into and out of the local switch, the sequence of remote receiving and remote sending reverses for every stage. The first node to receive a remote message in one phase must be the first to send it to the remote destination in the subsequent phase.

The second phase consists of any nodes which received a remote message in the final stage of the first phase performing a local broadcast of that message data.

4.3 Simulation experiments and results

We simulate execution of the three algorithms presented in the single switch section as well as the SO-R and LS algorithms on an irregular 128-node system. We again use a 5 microsecond startup time and 256 flit message length. The 128-node irregular network consists of 10 switches, each with 7 to 21 nodes attached. There are 1 to 5 external links to a switch with no more than 2 links between any 2 switches. The results of these simulations are shown in Table 2. The LS algorithm completes in 27.1% less time than the SO-R algorithm. This improvement reflects the increased communication component pipelining possible with the LS algorithms. Another significant result is the 94% improvement over the rank ordered ring (RO-R) algorithm. This result highlights the degradation in performance possible when the algorithm depends on the mapping of logical rank rather than adjusting based on knowledge of the underlying network topology.

Algorithm	128-node irregular topology
Combining	20542.14
SB	28537.30
RO-Ring	22632.85
SO-Ring	1868.07
LS	1361.36

Table 2. Completion time in microseconds for 128-node irregular network with 256 flit messages and 5 microsecond startup.

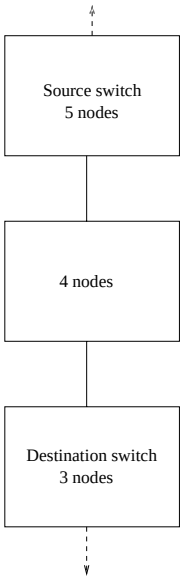
5 Conclusions

In this paper, we have shown efficient ways of implementing all-to-all broadcast on the emerging clusters of workstations based on commodity switch-based networks. We also characterized the performance of standard implementations of this operation, suggested simple enhancements which provide significant benefit, and introduced new algorithms which approach optimal performance in some cases.

For single switch clusters, the LR, SB, and Combining algorithms were considered. It was shown that selection of the best performing algorithm is a function of both network characteristics as well as the message length and number of participating nodes in the all-to-all broadcast operation. These algorithms were compared both analytically and through simulation.

An enhancement to the ring algorithm was proposed which gave significant performance improvements for 2-switch clusters. A 2-switch Link Scheduling algorithm was proposed which was link-contention free, node-contention free, and enabled pipelining of communication components. Execution of this algorithm was simulated for both balanced and unbalanced 2-switch networks with associated performance improvements over other algorithms.

Finally, two algorithms were proposed for arbitrary irregular networks. The Switch-Ordered Ring algorithms provides an efficient ordering for the logical ring of nodes, and the general Link Scheduling algorithm was introduced. The LS algorithm uses the same switch ordering as the SO-R algorithm but enables pipelining of communication components with associated performance improvements.



		receive pattern				send pattern			
step	PHASE 1	PE 1	PE 2	PE 3	PE 4	PE 1	PE 2	PE 3	PE 4
1	stage 1	2	3	4	S1	D3	1	2	3
2		3	4	S2	1	4	D2	1	2
3		4	S3	1	2	3	4	D1	1
4		S4	1	2	3	2	3	4	D1
5	stage 2	S5	1	2	3	2	3	4	D2
6		4	S5	1	2	3	4	D3	1
7		3	4	S4	1	4	D3	1	2
8		2	3	4	S3	D2	1	2	3
9	stage 3	2	3	4	S2	D1	1	2	3
10		3	4	S1	1	4	D1	1	2
11		4	S1	1	2	3	4	D2	1
12		-	1	2	3	2	3	4	D3
13	PHASE 2	4	-	2	3	-	3	4	1
14		3	4	-	2	-	4	1	2
15		2	3	4	-	-	1	2	3

Figure 1. LS algorithm communication patterns for a switch with 4 local nodes. In this example, there are 5 nodes on the source switch and 3 nodes on the destination switch. Assume a total of 15 nodes participating in the all-to-all broadcast so this switch must send a total of 12 messages to the destination switch and receive a total of 11 messages from the source switch. Source and destination nodes are identified by their respective indices in the switch. Messages originating from the remote source switch are identified by Sx and messages destined for the remote destination switch are identified as Dx where x is the node index.

References

[1] M. Barnett, S. Gupta, D. G. Payne, L. Shuler, R. van de Geijn, and J. Watts. Interprocessor Collective Communication Library (Intercom). In *Scalable High Performance Computing Conference*, pages 357–364, 1994.

[2] N. J. Boden, D. Cohen, et al. Myrinet: A Gigabit-per-Second Local Area Network. *IEEE Micro*, pages 29–35, Feb 1995.

[3] R. V. Boppana, S. Chalasani, and C. S. Raghavendra. On Multicast Wormhole Routing in Multicomputer Networks. In *Symposium on Parallel and Distributed Processing*, pages 722–729, 1994.

[4] L. De Coster, N. Dewulf, and C.-T. Ho. Efficient Multipacket Multicast Algorithms on Meshes with Wormhole and Dimension-Ordered Routing. In *International Conference on Parallel Processing*, pages III:137–141, Aug 1995.

[5] J. Duato. A Theory of Deadlock-Free Adaptive Multicast Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, pages 976–987, September 1995.

[6] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A High-Performance, Portable Implementation of the MPI, Message Passing Interface Standard. Technical report, Argonne National Laboratory and Mississippi State University.

[7] M. Jacunski, P. Sadayappan, and D. K. Panda. All-to-All Broadcast on Switch-Based Clusters of Workstations. Technical Report OSU-CISRC-10/98-TR44, The Ohio State University, October 1998.

[8] S. L. Johnsson and C.-T. Ho. Optimum Broadcasting and Personalized Communication in Hypercubes. *IEEE Transactions on Computers*, pages 1249–1268, September 1989.

[9] R. Kesavan, K. Bondalapati, and D. K. Panda. Multicast on Irregular Switch-based Networks with Wormhole Routing. In *Proceedings of the International Symposium on High Performance Computer Architecture (HPCA-3)*, pages 48–57, February 1997.

[10] R. Kesavan and D. K. Panda. Minimizing Node Contention in Multiple Multicast on Wormhole k-ary n-cube Networks. In *Proceedings of the International Conference on Parallel Processing*, pages I:188–195, Chicago, IL, Aug 1996.

[11] X. Lin and L. M. Ni. Deadlock-free Multicast Wormhole Routing in Multicomputer Networks. In *Proceedings of the International Symposium on Computer Architecture*, pages 116–124, 1991.

[12] P. K. McKinley, H. Xu, A.-H. Esfahanian, and L. M. Ni. Unicast-based Multicast Communication in Wormhole-routed Networks. *IEEE Transactions on Parallel and Distributed Systems*, 5(12):1252–1265, Dec 1994.

[13] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard*, Mar 1994.

[14] D. K. Panda, S. Singal, and R. Kesavan. Multidestination Message Passing in Wormhole k-ary n-cube Networks with Base Routing Conformed Paths. Technical Report OSU-CISRC-12/95-TR54, The Ohio State University, December 1995. *IEEE Transactions on Parallel and Distributed Systems*. In Press.

[15] J. Y. L. Park, H. A. Choi, N. Nupairoj, and L. M. Ni. Construction of Optimal Multicast Trees Based on the Parameterized Communication Model. In *Proceedings of the International Conference on Parallel Processing*, Chicago, IL, Aug 1996.