An Efficient Implementation of Tree-Based Multicast Routing for Distributed Shared-Memory Multiprocessors *

M.P. Malumbres and José Duato
Departamento DISCA
Universidad Politécnica de Valencia
Camino de Vera s/n, 46071 Valencia, Spain
jduato,mperez@gap.upv.es

Josep Torrellas
CSRD
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801
torrella@csrd.uiuc.edu

Abstract

This paper presents an efficient routing and flow control mechanism to implement multidestination message passing in wormhole networks.It is targeted to situations where the size of message data is very small, like in invalidation and update messages in distributed shared-memory multiprocessors (DSMs) with hardware cache coherence. The mechanism is a variation of tree-based multicast with pruning to avoid deadlocks. The new scheme does not require that the destination addresses in a given multicast message be ordered, thereby avoiding any ordering overhead. It allows messages to use any deadlock-free routing function and only requires one startup for each multicast message. The new scheme has been evaluated on several k-ary n-cube networks under synthetic loads. The results show that the proposed scheme is faster than other multicast mechanisms when the multicast traffic is composed of short messages.

1. Introduction

The performance of scalable multiprocessors is often determined by how effectively they support processor communication. Multicast communications routinely appear in parallel programs. Typical examples include explicit distribution of data to several nodes or invalidation and update messages in distributed shared-memory multiprocessors [2] (DSMs). Similarly, many-to-one messages are also common. Examples include barrier synchronization and global reductions. It appears, therefore, that optimizing the multicast operation would improve the performance of scalable multiprocessors.

Efficient support for multicast has been the subject of much previous research. Deadlock-freedom was studied for multicast communications in multicomputer networks using wormhole switching in [3, 6]. Multicast messages are propagated following a few paths that visit all destinations without suffering ramifications. This type of multicast is called path-based multicast. Routing algorithms like dual-path and multi-path were presented using 2D-meshes.

New partially and fully adaptive path-based multicast wormhole routing algorithms called PM, FM and LD were defined for 2D-meshes [4]. However, the design of deadlock-free adaptive multicast algorithms is complex. For this reason, new methodologies for designing deadlock-free adaptive multicast algorithms were proposed in [1, 5].

Recently, the BRCP (Base Routing Conformed Path) model was developed [7]. This is a new path-based message passing mechanism that transports multicast and broadcast messages and is deadlock-free. This mechanism can use any base routing scheme such as e-cube, planar-adaptive, turn-model or fully adaptive. Multicast and broadcast messages are carried toward their destinations in several sequential steps using two protocols: Hierarchical Leader-based (HL) and Multiphase Greedy (MG).

Path-based multicast has several inefficiencies, especially when messages are short. The main problem of the path-based scheme is that each multicast message needs a preparation phase to order the destinations. Usually, it involves a split-and-sort function with a software cost of $O(n* logn)$, where n is the number of destinations, increasing considerably the total latency of a multicast message. If the preparation phase is performed at compile-time, another problem of some path-based mechanisms is the number of steps needed to send a multicast message and its influence on the total message latency, taking into account that communication startup time is usually very high.

In this paper, we propose a new tree-based multicast mechanism that overcomes the limitations of the previously proposed mechanisms. First, the new mechanism does not require an initial ordering of the destinations and only needs one startup for each message (like unicast messages). Other features of the proposed scheme are: it can use a minimal path for all the destinations of a multicast message, it is able to use any deadlock-free routing algorithm used by unicast messages and it does not require several delivery channels to guarantee deadlock-freedom.

*This work was supported by the Spanish CICYT under Grant TIC94–0510–C02–01

The rest of this paper is organized as follows: Section 2 describes the new multicast mechanism; Section 3 analyses deadlock avoidance; Section 4 evaluates the scheme and compares it to other schemes; and Section 5 presents conclusions and future work.

2. Tree-Based Multicast with Pruning.

Tree-based multicast has traditionally been considered a good mechanism for broadcast and multicast in store-and-forward networks. However, with the arrival of wormhole switching, it became very prone to deadlock. As a consequence, other multidestination routing mechanisms like path-based multicast have been studied. However, we have reconsidered tree-based multicast, in order to accomodate it to wormhole switching and overcome some inefficiencies of path-based when DSM networks are considered.

The operation of the new tree-based multicast mechanism is similar to the traditional one in some respects. In a multicast message, each address flit is routed at all intermediate nodes. These nodes decide the best path to follow. They can open a new path if the paths reserved by address flits already processed are not good. Therefore, a multicast message will be able to expand as many branches as needed in its advance toward the destinations.

One of the differences of our scheme is the way in which we organize the information in a message. Figure 1 shows the format of a multicast message.

d1	data flits	d2	d3	● ● ●	dn

Figure 1. Message format for tree-based multicast with pruning.

For our scheme to work correctly, data flits must be stored in an auxiliary buffer at each intermediate node, even if it is not a destination. So, we need to add a new auxiliary buffer to store a copy of the data flits. This buffer is associated with every input channel of every node. When the first address flit of a message arrives at an intermediate node, the following data flits are copied to the corresponding auxiliary buffer. It stores the data until the tail of the message leaves the node. Thus, when a destination address flit, d_i where $i > 1$, is routed at a node n_k, two things may happen:

- If d_i opens a new path at node n_k, that is, it does not follow any path previously established by destination addresses d_1,...,d_{i-1} in the same message, then node n_k must inject the data flits after transmitting d_i. Thus, d_i establishes a new branch in the multicast tree.
- If d_i decides to use a path previously established by d_j (with $j < i$) in the same message, then it crosses node n_k following that path. Data flits are not injected after transmitting d_i, because they were sent after transmitting the destination address flit d_j.

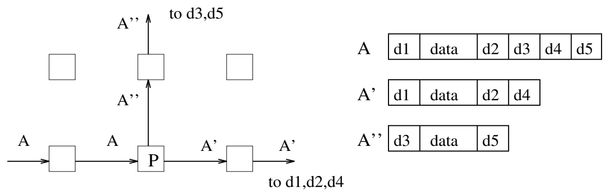

Figure 2. A multicast branching example: The original multicast worm, A, is divided into two worms, A' and A", at node P.

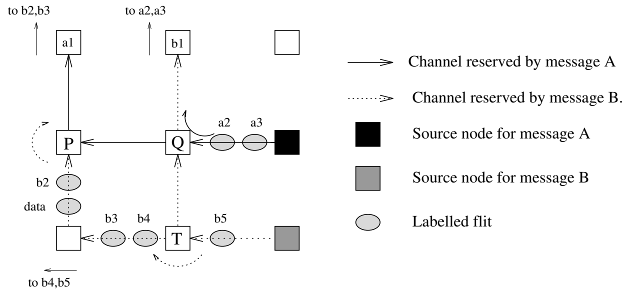

Figure 3. A deadlock between two multicast messages A and B.

It is important to note that the message format remains the same during the advance of the multicast worm. For each new branch, a "new" multicast message is expanded on-the-fly with the same format as the original one. Figure 2 shows an example of multicast worm propagation.

3. Deadlock Recovery in Tree-Based Multicast with Pruning.

We propose solving deadlocks and contention by controlling multicast ramifications through a pruning mechanism. When one of the branches of a multicast message is blocked at a given node, a pruning of all the other branches of the message is performed at that node. As flow control stops flits in previous nodes, pruning is also performed at those nodes. Then, the pruned branches can freely advance and release channels that could block other messages.

For example, figure 3 shows a deadlock on 2D-mesh using XY routing where two multicast worms block each other. Message A has three destinations: a_1, a_2 and a_3. Message B has five destinations: $b_1, b_2, ..., b_5$. The first address flit of A, a_1, crossed Q and P and reached its destination. The first address flit of B, b_1, also reached its destination crossing nodes T and Q. The deadlock state is reached when the destination address flits of each message, a_2 and b_2, can not advance because the other message is using the requested channel. The requested channels are indicated by arrows in figure 3.

To recover form deadlock, the pruning mechanism is used at nodes Q and T. When node Q routes a_2 and finds that there is no free output channel for it, a pruning of all the other branches of message A is performed at this node. In this case, the branch opened by a_1 is pruned, so that it can freely advance toward its destination. When node T routes b_5 and finds that there is no space in the selected output channel then the branch destined for b_1 will be pruned. This pruning is redundant but shows that nodes performing a pruning do not need to synchronize.

The described pruning mechanism is able to recover from deadlocks produced as a consequence of using tree-based multicast, assuming that the routing function for unicast messages is deadlock-free.

4. Evaluation.

We have developed a flit-level simulator of interconnection networks that supports unicast routing, path-based multicast routing and tree-based multicast routing with pruning. In our experiments, we run several simulations to analyze the behavior of tree-based multicast routing against unicast routing and path-based multicast routing algorithms like Dual-Path [6] and PM [4].

4.1. Simulation Parameters and Router Design.

All multicast messages have one data flit, a typical data size for invalidation messages in distributed shared-memory multiprocessors. The number of destinations of each multicast message varies between 4 and 25. A uniform distribution is used to construct the destination set of each multicast message. Deterministic routing algorithms are used for tree-based multicast and unicast in all the simulations. The routing algorithms for Dual-Path and PM have been described in [6, 4]. In the multicast experiments of section 4.2 traffic consists of multicast messages only. In section 4.3 we present simulations with a traffic pattern composed of unicast and multicast messages.

In all simulations, we have assumed that each physical channel has a bandwidth of one flit per clock cycle. Furthermore, both the switch and the routing circuit require one clock cycle to process a flit. In section 4.2, each router has four injection and delivery channels. Multiple delivery channels are required to avoid deadlock in path-based multicast algorithms [7]. However, in section 4.3 each router has only one injection channel and one delivery channel. Each physical channel has queues with capacity for two flits at each end and one auxiliary queue at the input side with capacity for one flit.

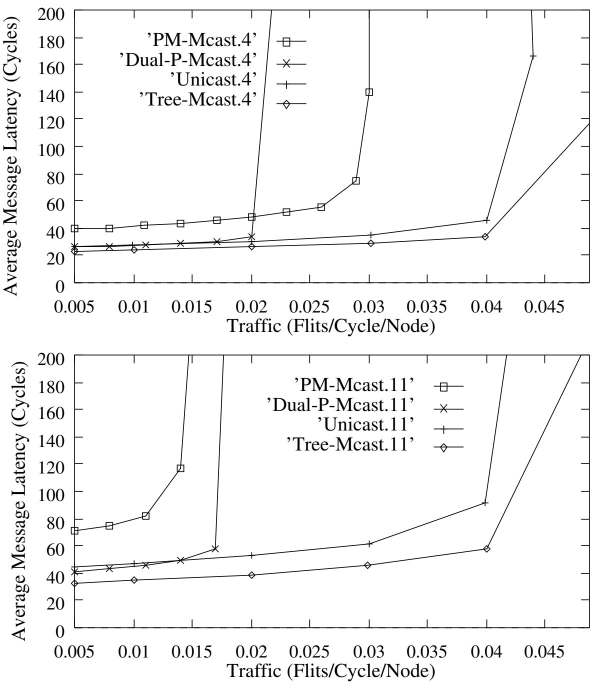

Figure 4. Comparative evaluation of different multicast mechanisms for an 8x8 2D-mesh.

4.2. Comparative Evaluation of Different Multicast Mechanisms.

In this section, we compare different multicast schemes on an 8x8 2D-mesh topology, using the simulation parameters presented above. We compare our tree-based mechanism to two path-based schemes, namely the Dual-Path and the PM routing algorithm. In addition, we include the unicast mechanism as a reference.

We note that the startup time and the time needed to generate multicast messages in path-based multicast schemes, have not been considered when computing the network latency. If we included such time, the Dual-Path and PM schemes would increase their latency by the amount of cycles required to perform the message preparation phase of each message. Also, Dual-Path will usually need two startups for each message and unicast will need n startups, being n the number of message destinations. Therefore, the results presented in this section for the Dual-Path, PM and unicast algorithms are optimistic.

Figure 4 shows the average message latency for the different multicast mechanisms using traffic composed of multicast messages with 4 and 11 destinations. The measurements are performed for different traffic loads. Each curve has a label that indicates the associated multicast mechanism and the number of destinations of each message.

The PM algorithm (PM-Mcast in figure 4) has a higher latency than the other algorithms for traffic conditions be-

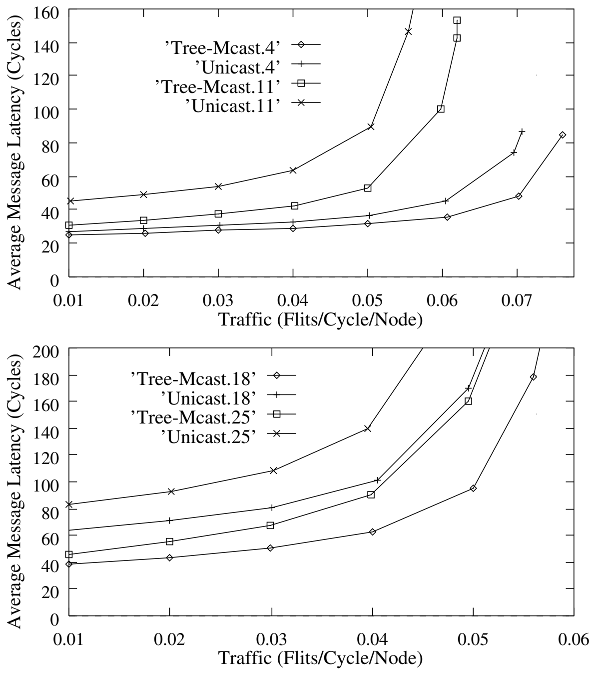

Figure 5. Tree-based multicast versus unicast in an 8x8 2D-mesh with mixed traffic.

low saturation. The Dual-Path algorithm achieves better results than the PM algorithm when the number of destinations grows. However, it reaches the saturation point very quickly if we compare it to unicast (Unicast in the figures) and the tree-based mechanism (Tree-Mcast in the figures).

Finally, we can see that tree-based multicast routing performs much better than path-based multicast routing for all traffic loads and number of destinations, even without considering the message preparation latency. Therefore, we now focus on comparing our scheme to unicast routing.

4.3. Tree-Based Multicast Evaluation with Mixed Traffic.

In order to obtain a more realistic view of the behavior of tree-based multicast, we analyzed the performance using mixed traffic consisting of unicast and multicast messages. However, we did not consider the startup latency. Note that this latency is constant for tree-based multicast but increases linearly with the number of destinations for unicast.

In figure 5, we show the average message latency of tree-based multicast and unicast routing. The traffic pattern consists of a 40% of unicast messages with 8 data flits per message and a 60% of multicast messages with one data flit. This pattern may be representative of the traffic in a distributed shared-memory multiprocessor where updates and invalidations produce multicast messages and cache misses are served by unicast messages, each one containing a cache line (8 data flits).

From figure 5, we can see that, under this load, tree-based multicast still behaves much better than unicast routing. So, we expect the new multicast scheme to have a good behavior under real traffic.

5. Conclusions and Future Work.

This paper has presented a fast multicast flow control mechanism for wormhole networks. The advantages of the new scheme are that multicast messages do not need a preprocessing step that orders the destinations, only require one start-up, reach the destinations following minimal paths if the base routing algorithm is minimal, work for any topology, and can use the routing algorithm of unicast messages. We call the new scheme tree-based multicast with branch pruning. The new scheme is deadlock-free and is particularly efficient for short messages, like those used to transfer invalidations and updates in DSMs.

We have presented a preliminary evaluation of the new scheme with simulations of synthetic multicast loads. Multicast messages routed with the new scheme have significantly lower latency than if they are routed with state-of-the-art path-based schemes. Furthermore, for high traffic, the network has substantially higher throughput. We also show that tree-based multicast with branch pruning is better than unicast, even when startup latency is not considered.

We plan to use SPLASH2 multiprocessor applications with multicast invalidation and update messages and a detailed simulation model of a DSM multiprocessor to evaluate the impact of the proposed scheme on overall execution time. We also plan to study efficient many-to-one communication schemes in order to develop a better support for acknowledgment messages in DSMs.

References

[1] J. Duato, A new theory of deadlock-free adaptive routing in wormhole networks, *IEEE Trans. Parallel Distributed Syst.*, vol. 4, no. 12, pp. 1320–1331, Dec. 1993.

[2] D.Lenoski, J.Laudon et al, The Stanford DASH multiprocessor, *IEEE Computer*, 25(3):63-79, March 1992.

[3] X. Lin and L.M. Ni, Deadlock-free multicast wormhole routing in multicomputer networks, in *Proc. 18th Annu. Int. Symp. Comput. Architecture*, May 1991.

[4] X. Lin, P.K. McKinley and A.H Esfahanian. Adaptive multicast wormhole routing in 2D-mesh multicomputers, in *Proc. Parallel Architectures Lang. Europe 93*, June 1993.

[5] X. Lin, P.K. McKinley and L.M. Ni. The message flow model for routing in wormhole-routed networks, in *Proc. 1993 Int. Conf. Parallel Processing*, Aug. 1993.

[6] X. Lin, P.K. McKinley and L.M. Ni, Performance evaluation of multicast wormhole routing, in *Proc. Int. Conf. Parallel Processing*, I:435-442, 1991.

[7] D.K.Panda, S.Singal and P.Prabhakaran. Multidestination message passing mechanism conforming to base wormhole routing scheme, in *Proc. of the Parallel Computer Routing and Communication Workshop*, pages 131-145, 1994.