On the Design of Scalable Pipelined Broadcasting for Mesh Networks

Ahmed Y. Al-Dubai and Mohamed Ould-Khaoua

Department of Computing Science
University of Glasgow
Glasgow G12 8RZ, UK.
E-mail: {aldubai, mohamed}@dcs.gla.ac.uk

Abstract

Minimising the communication latency and achieving considerable scalability are of paramount importance when designing high performance broadcast algorithms. Many algorithms for wormhole–switched meshes have been widely reported in the literature. However, most of these algorithms handle broadcast in a sequential manner and do not scale well with the network size. As a consequence, many parallel applications cannot be efficiently supported using existing algorithms. Motivated by these observations, this paper presents a new broadcast algorithm for the all-port mesh networks. The unique feature of the proposed algorithm is its capability of handling broadcast in only one message-passing step irrespective of the network size. Results from a comparative analysis and simulation reveal that the proposed algorithm exhibits superior performance characteristics over those of the well-known Recursive Doubling, Extending Dominating Node and Network Partitioning algorithms.

1. Introduction

The success of a large-scale multicomputer is highly dependent on the efficiency of its underlying interconnection network, which is constructed from routers and channels; the routers are responsible for moving data across the channels between the processing nodes. The mesh has been one of the most common networks for multicomputers due to its desirable properties, such as ease of implementation, recursive structure, and ability to exploit communication locality found in many parallel application to reduce message latency. The J-machine, Caltech Mosaic, Intel Touchstone Delta, Symult 2010, and Stanford DASH are examples of practical systems that are based on the mesh topology

Collective communication, such as broadcast, which refers to the delivery of the same message originating from a given source to all network nodes, is important in many real-world parallel applications found in the areas of Science and Engineering [4, 10]. For instance, broadcast communication is often needed in scientific computations to distribute large data arrays over system nodes in order, for example, to perform various data manipulation operations. Furthermore, it is required in control operations such as global synchronisation and to signal changes in network conditions, e.g., faults. In the distributed shared-memory paradigm, broadcast communication is often used to support shared data invalidation and updating procedures required for cache coherence protocols [12].

Several broadcast algorithms have been proposed in the literature for the wormhole-switched mesh [2, 3, 4, 11]. These algorithms try to reduce the broadcast latency by reducing the number of message-passing steps, i.e. the number of exchanges, required to perform a broadcast operation. However, most of these algorithms do not scale well with the system size as they suffer from the degrading effects of the start-up latency, the required time to handle a broadcast message at both the source and destination nodes [3], especially when the network size is large. This is because the number of message-passing steps that is required to complete a broadcast operation usually depends on the network size. In this paper, the Coded Path Routing (or CPR for short) which has been proposed in [1] will be used as a new approach for designing efficient broadcast algorithms for the mesh.

A unique feature of the CPR is that a message with a single address can be delivered to an arbitrary number of destinations with single start-up latency only. Specifically, the CPR is used to devise a new broadcast algorithm for the all-port 2-dimensional mesh. Owing to the properties of the CPR, the proposed algorithm requires a minimal number of message-passing steps to implement a broadcast operation, irrespective of the system size. An extensive comparative analysis presented below reveals that the new broadcast algorithm exhibits superior performance characteristics over the well-known Recursive Doubling, Extending Dominating Node and Network Partitioning algorithms proposed in [2], [3] and [15], respectively.

The remainder of this paper is organised as follows. Section 2 outlines the motivation of this study. Section 3 describes briefly the CPR. System model and the new broadcast algorithm for the all-port 2-D lie in Section 4..

Proceedings of the 16th Annual International Symposium on High Performance Computing Systems and Applications (HPCS'02)
0-7695-1626-2/02 $17.00 © 2002 **IEEE**

Section 5 compares the performance of the proposed algorithm to the Recursive Doubling, Extending Dominating Node and Network Partitioning algorithms. Finally, Section 6 concludes this study.

2. The Motivation

Existing algorithms for collective communication, such as broadcast, are founded on either the unicast-based [14] or multidestination-based approach [7]. The main objectives of the two approaches is firstly to reduce the waste of network bandwidth due to additional traffic caused by excessive replication of broadcast messages inside the network, and secondly to reduce the communication delay due to the start-up latency. The start-up latency varies from one system to another, and is usually higher than channel transmission times in terms of current implementation technology [12].

Current practical multicomputers have adopted the unicast approach due to its simplicity. Collective communication in this approach is implemented as a sequence of unicast message exchanges as it uses the same routing provided for normal unicast (one-to-one) messages. The algorithms proposed by Barnett et al [2], Tsai and McKinley [3] and Cang and Wu [11] are examples that use the unicast approach. Unfortunately, these algorithms use several phases of message exchanges, each phase encountering separated start-up latency. As a result, the communication overhead can be significant and detrimental to network performance, especially in the presence of a high start-up latency [9].

Several researchers have suggested the multidestination approach to reduce the degrading effects of the start-up latency, [7, 10]. A message in this approach can carry many addresses, and can be delivered to a group of destinations in a single message-passing step. The source node generates an ordered list of destinations, i.e. depending on the intended order of traversal, and incorporates it into the header flit. Lin and Ni [7] have proposed one of the first algorithms that employ this approach. Their algorithm reduces the number of message-passing steps by using Hamiltonian-path based routing. The authors in [9] have suggested some additional hardware to the router in order to support this scheme; for instance, the router should be capable of delivering an incoming broadcast message to the local host while simultaneously forwarding it to the next router.

Despite the fact that the multidestination approach reduces the number of message-passing steps required, it suffers from several limitations. Firstly, each message-passing step requires a message preparation phase to sort n addresses with a minimum software cost of O($n\times\log n$) [8]. Consequently, this preparation phase may take more time than the actual message transmission time [8]. Secondly, since the list of addresses in the header flits are sorted, the routing may not use a minimal path for all of the sorted destinations. This increases the message journey through the network, and as a result may lead to increased message contention inside the network. Thirdly, due to the presence of many addresses in the header, each address occupies one flit. If each flit in the header is assumed to require one updating cycle, this approach requires (n–1) additional communication cycles that are spent in updating n addresses in the header flits.

Most previous studies [2, 3, 4, 5, 11, 14] have focused on minimising the number of message-passing steps required for collective communication, such as broadcast. However, there has been hardly any study that has considered minimising the effects of the network size on the performance of broadcast algorithms. As a result, most existing algorithms do not scale well with the network size since the number of message-passing steps increases proportionally with the system size. In an effort to address this issue, this study proposes a new routing approach for the development of efficient broadcast algorithms that can maintain good performance levels for various network sizes.

3. The Coded Path Routing (CPR)

This section describes the Coded Path Routing (CPR) approach that can reduce the overhead due to the start-up latency and the effects of the network size on the performance of collective communication. The CPR exploits the main features of wormhole switching, such as few buffer requirements and distance insensitivity, to overcome the limitations of the existing approaches, and to efficiently support collective communications.

In the CPR, the header flit has two bits that form the control field. The two bits indicate to a router which action to take, e.g., pass or receive, upon the reception of a message. Fig. 1 describes the “Control Field” algorithm that the router use to either interpret or modify the control field. In fact, the two bits of the control field have originally been specified in order to enable the CPR to be used in different systems, such as those using one-port or multiple port router models, and also to support different types of collective communication operations, including broadcast and multicast.

However, to illustrate the advantages of the CPR, we will focus our discussion in the present study on the use of the CPR for the development of broadcast algorithms; we plan to extend in the future the application of the CPR to multicast communication

```
Procedure Control Field (message, operation)
  Begin
        receive the second field of the message;
        if (current router is not the addressed router) then
               if (control field =10) then pass the message to the next router
                 else
                      if (control field =01) then receive the  message;
                      else
                          if (control field =11) then
                          {
                             receive the message; pass the message to the next router ;
                          }
         else receive the message;
  End.
```

Figure 1: The "Control Field" algorithm that a router uses in the CPR.

As in the path-based algorithms of [6, 7], which use the multidestination approach, it is assumed that a router in the CPR can simultaneously receive a message and passes a copy to the next router. The CPR has several advantages over the existing routing algorithms used in the unicast and multidestination approaches. Unlike in the unicast-based approach, a message with a single address in the CPR can be delivered to an arbitrary number of nodes with only single start-up latency. This is achieved by adding a simple circuitry to the existing router logic to deal with the two bits in the control field.

The router in the CPR is simpler to implement than that in the multidestination approach as some operations are not required during a message-passing step. These include sorting addresses, deleting current address and making another routing operation according to the order of the destinations in the header flit. Moreover, unlike in the multidestination approach, the message length in the CPR is fixed regardless of the number of destination nodes that receive the message. This minimises the header-processing overhead that forms a significant drawback in the multidestination-based approach. Finally, another benefit of the CPR is related to its higher flexibility in determining a path for all destinations during a given message-passing step. In router-based network, the router is mainly responsible of communication operation by sending messages to or receiving messages from neighbouring nodes. The main advantage of wormhole routers is that their buffer requirements can be small, making routers to be extremely small and fast. In general, each wormhole router consists of address decoders, routing arbitration unit, switch and several channels with their corresponding channel controllers [13]. Fig. 6 illustrates the general router structure in wormhole switched meshes. Once the message header arrives at the router inputs it is fed into the address decoders, which extracts the packet address and generates requests demanding acceptable outputs, based on the underling routing algorithm.

In multidestination-based scheme, the router becomes responsible for carrying out further jobs; such as preparing an ordered list for the addresses, removing every updated address and making many routing steps based on the node whose address will be at the top of the ordered list [7]. Obviously, the multidestination-based router is susceptible to additional routing overhead. In [13], the set-up delay in dimensional order wormhole router is calculated as follows

$$T_{router\ \ delay} = T_{Ad.} + T_{Ar.} + T_{Sw.} \qquad (1)$$

where, $T_{Ad.}$ is the time required for extracting and updating the header by the address decoders, whereas $T_{Ar.}$ and $T_{Sw.}$ refer to the time required by arbitration unit and switching unit, respectively. For N destinations in multidestination-based scheme, the additional overhead required by addressing decoders is calculated approximately by:

$$T_{overhead} = O(N \times \log N) + T_{updating}(N-1) \qquad (2)$$

while the first component represents the preparation time required for sorting the destinations according to the routing algorithm as a software overhead cost [13], the second one refers to the additional overhead for updating the list of destinations.

It is now clear that the main drawback of multidestination-based scheme is that it sacrifices router efficiency for preparing and updating the list of destinations. In contrast, the CPR router does not require additional buffer storage for the head message. Unlike the previous schemes, the CPR requires only two additional bits in the header, which can be included in the same flit that carries the destination address. Specifically, a new logic unit is required in address decoders to make the router capable of dealing with the address and control field of the CPR while updating the header. As soon as the header reaches address decoders, the control field is extracted with the address and a request is generated demanding switch crossbar a proper output channel. As a concurrent action, the message is then dealt according to the value of control field of the CPR. Thus, the CPR router requires only a minor modification to the existing address decoder in the typical unicast-based router compared to that of the multidestination-based scheme. For further detail on the CPR, we refer the reader to [1].

4. Preliminaries

4.1. The System Model

A 2D mesh has $N = N_x \times N_y$ nodes, arranged in the two dimensions X, and Y respectively, with N_x and N_y being the number of nodes in the two dimensions. A node is identified by a two co-ordinate vector $(x, y,)$, $0 \le x \le N_x - 1$, $0 \le y \le N_y - 1$. The mesh topology is asymmetric due to the absence of the wrap-around connections along each dimension. Therefore, nodes may not be connected to the same number of neighbours; those at the corners, edges, and middle of the network have 2, 3 and 4 neighbours respectively. Fig. 2 depicts the structure of target system. In this system, the node consists of a processing element (PE) and router.

The PE contains a processor and some local memory. A node uses four input and four output channels to connect to its neighbouring nodes; two in a dimension, one for each direction. There are also local channels used by the PE to inject/eject messages to/from the network, respectively. Messages generated by the PE are injected into the network through the injection channel.

Messages at the destination node are transferred to the PE through the ejection channel. Similar to the previous studies of [9, 10], this study considers the multiple-port router model where multiple copies of the broadcast message can be injected into the network through different output channels concurrently. Furthermore, multiple broadcast messages can be transferred to the local PE. The input and output channels are connected by a crossbar switch that can simultaneously connect multiple input to multiple output channels given that there is no contention over the output channels.

4.2. A New Broadcast Algorithm

This paper proposes the Parallel Coded Paths algorithm referred to below as PCP for short, for an all-port 2D mesh based on the CPR. The new PCP algorithm exploits the features of the CPR to implement broadcast operations in a single message-passing step, thus considerably reducing the effects of the start-up latency. Fig. 2 describes the proposed algorithm. Examining Fig. 2 shows that the PCP algorithm can fully exploit the all-port feature of the mesh to achieve low communication latency during the propagation of the broadcast message from one router to the next. This has the net effect of greatly reducing the overall time required to complete a broadcast operation.

The PCP algorithm achieves this using the following rules. Firstly, a source node simultaneously sends four copies of the broadcast message across the four directions $-X$, $+X$, $-Y$ and $+Y$ after changing the control field in the header flit from “00” to “11”, as outlined above in the CPR approach. To avoid the contention and deadlock problems, only the source node uses $\pm Y$ directions. Secondly, each message crosses all the intermediate nodes along each dimension, where the last destination node is specified in the header flit.
Then, only the nodes that share the same co-ordinate on the Y dimension with the source node send copies of the broadcast message along the $\pm X$ directions. After that, each node receives the message along the $\pm X$ directions, it forwards copies across the same direction, taking the advantage of the all-port facility. Finally, the broadcast operation is completed when the broadcast message reaches the corners.

Unlike the existing broadcast algorithms, this algorithm can achieve a high degree of parallelism by allowing the periods of transmission to overlap each other, enabling most destination nodes to receive the broadcast message in parallel, thus leading to considerably minimising the overall latency required to implement the broadcast operation.

5. Performance Analysis and Comparison

In this section, we compare the performance of the proposed PCP algorithm to the well-known Recursive Doubling [2], the Extended Dominating Node [3] and the Network Partitioning algorithm [15]. In the rest of this section we will use the short abbreviation PCP, RD, EDN and NP-D to refer to the three algorithms, respectively. Firstly, we will compare these three algorithms in terms of the number of message-passing steps required. We then conduct a timing analysis to estimate the communication latency experienced by a broadcast message. Finally, we present results from simulation experiments to study the dynamic behaviour of the PCP under different traffic conditions. Let us now compare the performance of the proposed PCP algorithm to the PCP, RD, EDN and NP-D in terms of scalability, i.e. the number of message passing steps required to implement broadcast operation in different network sizes.

Definition 1 Given a source node (S_x, S_y), destination node/nodes d such as $d \subseteq N$ in 2D mesh network, sending start-up latency α and receiving start-up latency γ; we say that (S_x, S_y) capable of delivering a broadcast message M to d in a single message-passing step if and only if it requires $(\alpha + \gamma)$ as start-up latency, irrespective of the number of nodes traversed.

Firstly, we will compare these broadcast algorithms in terms of the number of message-passing steps required. We then present results from simulation experiments to study the dynamic behaviour of the PCP under different traffic conditions.

```
Algorithm: The PBR broadcast in All-Port 2-D Mesh with CPR
/* Input: source node $(S_x, S_y)$; M: Message */
/* Output: All nodes receive a copy of M */
Control field :=00;
Let $x^+ > S_x$, $y^+ > S_y$, $x^- < S_x$ and $y^- < S_y$
Let $A^{Sx,\pm y}$ be the nodes which have the $S_x$ in the X co-ordination.
Let $A^{\pm x,Sy}$ be the nodes which have the $S_y$ in the Y co-ordination.
Let $C_{corners}$ include the four corner nodes of the mesh.
Control field:=11;
send M to $A^{\pm x,Sy}$
for all $x^+, x^-, y^+, y^-$ do_in_parallel
{
  receive and pass M;
  if node $A \in A^{Sx,\pm y}$ then
    {
     for all $x^+, x^-$ do_in_parallel
     receive and pass M;
    }
  else
  if node $A \in C_{corners}$ then
   {
    receive M;
    control field:=00;
   }
}
```

Figure. 2: A description of the proposed PBR broadcast algorithm.

The RD was originally proposed by Barnett et al [2]. This algorithm requires $Log_2 N$ steps for broadcasting in 2D mesh. In this algorithm, each node holding a copy of the message is responsible for a partition of a row or column, which will be then divided in half. In each half, a node sends a copy of the message to the node in the other half that occupies the same relative position. This process is implemented recursively until the completion of the broadcast operation. In the absence of contention problem, it can fully take advantage of the pipelining effect of wormhole switching. The EDN was proposed by Tsai and McKinley [3].

The EDN model has been developed to systemically construct collective operations for multiport wormhole-routed networks. In this approach, the network is divided into several levels. For each level, a dominating set is assigned. For instance, a dominating set D of a graph G is a set of vertices in G such that every vertex in G is either in D or is adjacent to at least one vertex in D. The authors in [11] have shown that the number of message-passing steps required to implement broadcasting in a network size of $(2^k \times 2^k)$ is $k+1$.

In [15], the NP-D has been proposed to achieve more parallelism during broadcast operation. The broadcast algorithm works in three stages. First, the main network is divided into sub-networks and the source node divides the message M broadcast into sub-messages h; a leader node is chosen for each sub-network. Second, the leader nodes perform a broadcast operation concurrently. Third, the sub-networks collect sub-messages $M_0, M_1, ..., M_{h-1}$ from the nodes that have received sub-messages in the second stage and combine them into the main message M [15]. However, the NP-D does not minimise the message-passing steps required to perform broadcast operation in that it requires $k+2$ so as to perform a broadcast operation in the $(2^k \times 2^k)$ mesh [15].

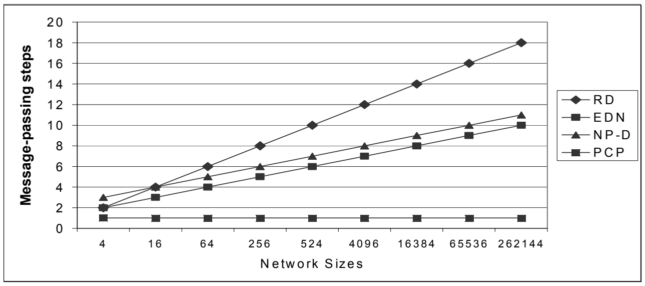

Figure 3: Comparison of the message passing steps

Fig. 3 plots the number of message-passing steps required for a broadcast operation by the four algorithms in the $(2^k \times 2^k)$ mesh. For the sake of illustration, the network size was varied from 64 to 262144 nodes. The results reveal that while the number of steps required by the EDN, RD and NP-D algorithms increases with the system size, that required by the PCP is fixed (only one) regardless of the network size. This section analyses the communication latency, that is the time required for a broadcast message to reach all the network nodes, in the PCP, EDN and RD. We use the resulting expressions for the communication latency to study the effects of important parameters, such as the message length, network size and start-up latency on the performance of the three algorithms.

The pipelining nature of wormhole switching makes latency less sensitive to message distance, especially when message are long. We define the communication latency for a broadcast operation as the time when a broadcast message is injected into the network until the last node in the network receives the message. Although in a multiple port system a source node can send multiple copies of the broadcast messages simultaneously through different channels in succession, the source prepares several copies of the broadcast message in a sequential manner [3].

Definition 2 In the absence of contention in the network, the communication latency, τ, for a message length of L flit can be generally estimated as

$$\tau_{Broadcast} = M\alpha + \beta D + \beta L + C\mu + \gamma \qquad (3)$$

where M: is the number of copies of the broadcast message prepared by the source to be injected into the network, α: the sending latency for each message, β: the time required to transmit a flit on a channel, D: the distance between the source and destination of a message, γ: the receiving latency, μ: the time required to change the control field in the header message and C: the number of message-passing steps required to deliver the message to all network nodes.

Both the sending and receiving latency form the start-up latency, i.e. start-up latency = $\alpha + \gamma$ [2, 3]. Definition 2 will be used as a basis for analysing the execution time of the PCP, RD and EDN.

As in the previous studies [2, 3], we assume that the sending latency time and the receiving latencies are approximately equal, i.e., $\alpha \approx \gamma$. Let us determine first the communication latency of the PBR. Let the time required for changing the value of control field in the CPR equal the sending latency $(\mu = \alpha)$. For a source (S_x, S_y, S_z), a message visits the following number of channels along each of the three dimensions.

$$\begin{aligned} d_X &= \max((N_x - 1 - S_x), S_x) \\ d_Y &= \max((N_y - 1 - S_{y)}), S_y) \end{aligned} \qquad (4)$$

The communication latency, τ_{BCP}, in the PCP, can be approximated by

$$\tau_{PCP} = \beta\ (d_X + d_Y) + \beta L + 4\alpha + 2\mu + \gamma \qquad (5)$$

The RD requires $Log_2 N$ steps to complete a broadcast operation [2]. In a network size of $(2^k \times 2^k)$, the authors in [1] have shown that this algorithm requires $2k$ massage-passing steps. The time required to send a broadcast message to all the network nodes can be written as [3]

$$\tau_{RD} = 2k(\alpha + \gamma + \beta L) + D\beta \qquad (6)$$

The EDN requires k+1 message-passing steps [2]. Based on the rules of this algorithm, a total time of $(3\alpha + \beta L + \gamma + d_1\beta)$ is required for a message sent by the source to reach the four highest-levels of the network with k-1 message-passing steps. Two message passing steps are needed required by the source node to deliver the message to the four highest –level EDNs. Therefore, the EDN broadcast algorithm requires a total time approximated by:

$$\begin{aligned}\tau_{EDN} &= (K-1)(3\alpha + \beta L + \gamma + d_1\beta) \\ &\quad + 3\alpha + 2\beta L + 2\gamma + d_2\beta \end{aligned} \qquad (7)$$

where d_1 and d_2 represent the distances traversed by the message in the $k-1$ message-passing steps and the last two steps, respectively. Based on the rules of the NP-D broadcast algorithm [15], the total time required for $(2^k \times 2^k)$ mesh can be approximated by

$$\begin{aligned}\tau_{NP\text{-}D} &= \\ &(k - d + 2^d + 3 + [(d+1)\log_3 2] + [d\log_5 2)]\alpha \\ &+ (\frac{9}{4} + \frac{(k - d + 2 + [(d+1)\log_3 2)}{2^d})\gamma \\ &+ (3 \times 2^k + 2^{d+1} - 3)\beta L \end{aligned} \qquad (8)$$

Examining equations 3,4,5 and 6, the results show that our CPR broadcast scheme has the lowest α, γ and β factors and the network size insensitivity is shown through these equations as a unique feature for our broadcast algorithm. Having examined the improvement in broadcast achieved by the PCP in the terms of the number of message-passing steps required, in what follows we conduct a performance comparison of the PCP and RD under dynamic situations using simulation experiments. We have decided to exclude the EDN and NP-DA algorithms from the comparison because the RD is much more able to take advantage of the pipelining effect of wormhole routing to avoid channel contention among messages [11]. A simulation program was used to model the broadcast operations of the PCP in the 2D mesh. The program was written in VC++ and built on top the event-driven CSIM 18-package [16]. In the simulation software, processes are used to model the active entities of a system, and can execute in a quasi-parallel fashion, providing a convenient interface for writing modular simulation program. In our case, every message is modelled as a process. For studying broadcast operation, the main program activates a set of CSIM parallel processes that are used to broadcast a message in the network.

In each experiment, different source nodes were chosen randomly using a uniform number generator. The statistics were collected with 95% confidence interval when the system reaches the steady state; when results do not change much with time. It is worth mentioning that the values of the communication latency parameters are consistent with the Cray T3D (channel rate was set at β =0.0033 μs and $\alpha = 0.75\mu s$) [10].

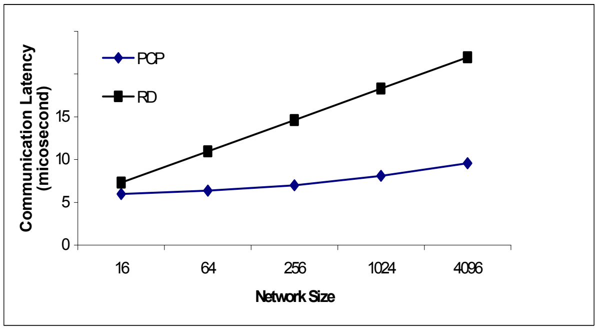

Figure 4: Comparison of broadcast algorithms in 8×8 mesh (T3D parameters

Fig. 4 plots the communication latency versus different network sizes with 100 flits for the message length. The results from the figure show that the advantage of our algorithm is significant as it requires a fixed number of message-passing steps in the all network sizes and it is able to fully utilise the multiple-port facility of the system. In contrast, Fig. 4 confirms the fact that the RD performance is highly dependent on the network size. In Fig. 5, we have examined the performance of the PCP and the RD using two network sizes N $= 4 \times 4$ and 8×8. The message length was varied from 30 to 210 flits and the parameters of the communication latency were set in a way to be consistent with the Cray T3D [1].

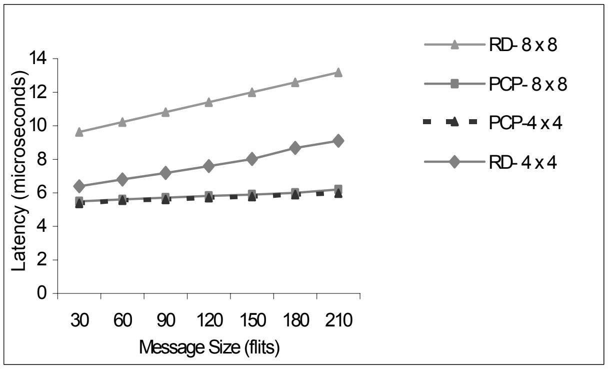

Figure 5: Comparison of the broadcast algorithms in 8×8 and 4×4 mesh (T3D parameters)

Fig. 5 reveals that there is a slight difference in the achieved performance of the PCP in the two network sizes. This is because in the PCP, when the network size increases, only the distance (i.e., the number of nodes) traversed by the message increases while there is no increase in the number of message-passing steps required to complete the broadcast operation. Taking the advantage of wormhole distance insensitivity, the PCP algorithm has also become insensitive to the network size. However, turning to the RD shows the high effect of network size on the algorithm performance. If we increase the network size to 8×8, the advantage of the CPR becomes more noticeable. This is due to the fact that the RD, like most of existing algorithms, is highly sensitive to the message length. It is negatively affected by the increase in the number of the message-passing steps, required to meet the increase in the network size, and the increase in the message length.

6. Conclusion and Future Directions

While the existing broadcast algorithms implement broadcasting sequentially and, therefore, do not scale well with the network size, this paper has suggested an efficient broadcast algorithm, which overcomes the limitations of the existing ones. The proposed algorithm has the main advantage of requiring only one message-passing step irrespective of the network size. Unlike the previously proposed algorithms, our algorithm achieve a high degree of parallelism during the propagation of the broadcast message from one router the next, i.e., most of the network nodes receive the broadcast message in parallel. Moreover, our performance analysis and simulation results have revealed that the proposed algorithm has superior performance characteristics than the existing Recursive Doubling, Extending Dominating Node and Network Partitioning algorithms. The next step in our work is to extend our work towards devising new multicast algorithms and compare their performance with existing well-known algorithms. Another possible line for future research is to support collective communication in other common multicomputer networks, such as hypercubes and tori.

References

[1] A. Y. Al-Dubai, M. Ould-Khaoua, An efficient adaptive broadcast algorithm for the mesh network, *Proc. 8th Int. Conf. Parallel and Distributed Systems (ICPADS'2001), IEEE Computer Society* Press, KyongJu City, Korea, pp. 83-90, June 26-29, 2001.

[2] M. Barnett, G. David, R. A. van de Geijn and J. Watts, Broadcasting on meshes with wormhole routing *IEEE, J. Parallel & Distributed Computing*, vol. 35, pp. 111-122, 1996.

[3] Y.-J Tsai and P.K. McKinley, An extended dominating node approach to broadcast and global combine in multiport wormhole routed mesh networks, *IEEE, J. Parallel & Distributed Computing,* vol. 8, no. 1, pp. 41 – 58, 1997.

[4] J. Watts, Efficient collective communication on multidimensional meshes with wormhole routing. *Tech. Rep. TR-94-19. Dept. Computer Science, Univ. Texas at Austin*, June 1994.

[5] D. Scott, Efficient all-to-all communication patterns in hypercube and mesh topologies, *Proc. Int. Symp. Parallel & Distributed Processing*, pp. 398-403, 1991.

[6] D. F. Robinson, P. K. McKinley and B. C. Cheng, Path based multicast communication in wormhole routed unidirectional torus networks, *J. Parallel & Distributed Computing*, vol. 45, pp.104 - 121, 1997.

[7] X. Lin and L. M. Ni, Deadlock-free multicast wormhole routing multicomputer networks, *Proc. Int. Symp. Computer Architecture*, pp. 116-124, 1991.

[8] M. P. Malumbres and J. Duato, An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors, *J. Systems Architecture*, vol. 46, pp. 1019-1032, 2000.

[9] D. K. Panda, Issues in designing efficient and practical algorithms for collective communication on wormhole routed systems, *Proc. 1995 Workshop Challenges for Parallel Processing*, pp. 8-15, 1995.

[10] R.E. Kessler and J.L. Schwarzmeier, CRAY T3D: A new dimension for Cray research, *Proc. ComCon*, pp. 176-182, 1993.

[11] S. Cang and 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] J. Duato, S. Yalamanchili and L. Ni, Interconnection networks: An engineering approach, *IEEE Computer Society Press*, 1997.

[13] Andrew A. Chien, "A Cost and Speed Model for k-ary n-cube Wormhole Routers," *IEEE Trans. Parallel & Distributed Systems*, vol. 9. no. 2, pp. 150-160, 1998.

[14] P. K. McKinley, H. Xu, A. Esfahanian and L. M. Ni, Unicast-based multicast communication in wormhole-routed direct networks, *IEEE, J. Parallel & Distributed Computing*, vol. 5, no. 12, pp. 1254-1265, 1994.

[15] Y.-C. Tseng, S.-Y. Wang and C.-W. Ho, Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach, *IEEE, J. Parallel & Distributed Computing s*, vol. 10, no. 1, Jan. 1999.

[16] H. D. Schwetman, CSIM: A C-based, process-oriented simulation language, *Tech. Rep*. pp. 80-85, Microelectronics and Computer Technology Corp., 1985.