# Multidestination Message Passing Mechanism Conforming to Base Wormhole Routing Scheme[1]

**Dhabaleswar K. Panda, Sanjay Singal and Pradeep Prabhakaran**

**Dept. of Computer and Information Science**
**The Ohio State University, Columbus, OH 43210-1277**
**Tel: (614)-292-5199, Fax: (614)-292-2911**
**E-mail: {panda,singal,prabhaka}@cis.ohio-state.edu**

**Abstract:** This paper proposes a novel concept of *multidestination worm* mechanism which allows a message to be propagated in a wormhole network *conforming* to the underlying *base* routing scheme (ecube, planar, turn, or fully adaptive). Using this model, any source has potential to deliver a message to multiple destinations in any *valid path* in the system conforming to the base routing scheme while encountering only a *single* communication start-up. The flexibility of sending unicast messages exists under this model as a subset operation. Two schemes are developed and evaluated under this model to perform fast *multicasting* on 2D/3D meshes/tori. Not only do these schemes demonstrate superiority over Umesh (unicast-based multicast) [25] and Hamiltonian-Path-based [24] schemes, they indicate a very *unique* and *interesting* result that the cost of multicast operations can be *reduced* or kept *near-constant* in e-cube systems as the degree of multicast (number of destinations/src) *increases.* The proposed schemes can also take advantage of routing-adaptivity and irregularity in system topology to further reduce the cost of multicast operations. These results are the first ones in the wormhole-routing literature to propose multicasting schemes with such reduced overhead and provision for taking advantage of adaptivity. This multicast model lays a new foundation to build wormhole-routed distributed-memory systems with fast collective communication operations and distributed-shared-memory systems with fast cache-coherency support.

[1]This research is supported in part by the National Science Foundation Grant # MIP-9309627.

# Contents

# 1 Introduction

The wormhole-routing switching technique is becoming the trend in building future parallel systems due to its inherent advantages like low-latency communication and reduced communication hardware overhead [13, 27]. Intel Paragon, Cray T3D, Ncube, J-Machine, and Stanford DASH are representative systems falling into this category. Such systems are being used for supporting either distributed-memory, shared-memory, or distributed-shared memory programming paradigms. In order to support these paradigms, these systems need fast implementation of collective communication operations (*broadcast, multicast, global combine*, and *barrier synchronization*) [26].

Traditionally, the wormhole-routed systems have supported only *point-to-point* (unicast) message passing mechanism. This mechanism allows a message to have a single destination only. Collective communication and synchronization operations get implemented as costly operations on these systems with *multiple phases* of unicast message exchange. Some examples are broadcasting algorithms by Van de Geijn[4], multicasting algorithms using Umesh operations by McKinley [25], complete exchange algorithms by Scott [33] and Bokhari [6], barrier synchronization algorithms for hypercubes by McKinley [34]. All these algorithms use several *phases* of message exchange to implement the required operation. Since each phase of the algorithm encounters a communication start-up cost, the overhead of the communication algorithm is proportional to the number of phases used in the algorithm especially in the presence of high communication start-up cost. Hence, the basic principle for designing these algorithms has been to use minimum number of phases.

To reduce the number of phases to implement dense communication operations, Ni has proposed a Hamiltonian Path-based scheme [24]. It has been demonstrated that fast multicasting and broadcasting can be achieved in wormhole-routed systems using this approach. This scheme uses a modified router organization to support propagation of multidestination messages (a message with multiple destinations) and allows a message to be consumed at a destination node while being forwarded to the next node concurrently. The message propagation is restricted to a Hamiltonian Path in the network to prevent deadlock. Such an approach is very promising and demonstrates that collective communication operations can be implemented efficiently on wormhole-routed systems. The Hamiltonian path-based scheme also allows both unicast and multidestination messages to co-exist in a system. All messages conform to the Hamiltonian Path (with or without shortcuts) and hence, the routing exhibits limited adaptivity. Though this approach tries to reduce the number of phases, the path length becomes dominant for broadcasting or multicasting with large number of destinations, especially when the system size is large.

Though Hamiltonian path-based routing demonstrates significant potential, its development has been in a direction which is different from the conventional developmental track of wormhole-routed systems. In the past, the wormhole-routed systems started with e-cube routing [13]. Recently in the literature, various deterministic and adaptive routing schemes have been proposed [1, 7, 11, 15, 17, 31]. Though today's commercial and experimental systems (Intel Paragon [19], Cray T3D [12], Ncube-3 [16], Stanford DASH [21]) use only e-cube routing, a natural extension for these machines in near future will be to go for some adaptive routing scheme. In order to implement Hamiltonian-path-based routing, these machines need to go through a complete set of router modification. Hence, it is less likely that a system in near future will be able to take advantage of the potential of Hamiltonian-path-based routing.

If the wormhole-routed systems in near-future go for adaptive routing, a natural question arises whether these systems, besides delivering higher system throughput due to adaptivity, can also implement fast collective communication operations by taking advantage of adaptivity. The collective communication algorithms proposed in the literature so far have been geared towards e-cube routing. Very little work has been done to take advantage of adaptivity to perform these operations faster. In [14], Duato has presented a theory of deadlock-free adaptive multicast routing. A single-path-based adaptive wormhole routing scheme is proposed in [22] to accommodate both unicast and multicast communication in hypercubes. However, no work has been done to develop communication mechanisms for k-ary n-cube wormhole networks which can blend well with the available adaptivity in a system to facilitate efficient implementation of collective communication operations.

In this paper, we take such an approach in defining a new framework of *Base-Routing-Conformed-Path* (BRCP) model. This model is tightly-coupled with multidestination message-passing mechanism while providing flexibility for unicast messages. Unlike the Hamiltonian-path-based scheme, the BRCP model allows a multidestination message to be propagated through any *path* in the network as long as it is a *valid* path *conforming* to the base-routing scheme. For example on a 2D mesh with e-cube (row-column) routing, a valid path can be any row, column, or row-column. If the network supports planar-adaptive routing, a message can also be propagated along a diagonal path. Similar flexibility exists for networks supporting fully-adaptive routing and turn model.

To demonstrate the potential of this new mechanism, we develop and present two schemes for broadcast/multicast. Broadcasting is an important collective communication operation [26]. When this operation takes place over a subset of processes or the processes are not contiguously allocated in a system, it reduces to multicast operation from the system's perspective. These schemes perform far better compared to the Umesh [25] and the Hamiltonian-path-based [24] schemes even for systems with e-cube routing. As the number of destinations for a multicast *increases*, we show that the overhead of multicast can in fact be *reduced*. With increase in adaptivity in the network, the schemes perform even better. We also show that the model can take advantage of *irregularity* in topology and *spatial positions* of the participating nodes to perform multicasting faster. It is shown that this model can be very easily implemented in the existing routers by incorporating minor additional logic to the router and hence, can be incorporated into near-future wormhole-routed systems to obtain significant performance gain with minimal increase in cost.

The paper is organized as follows. The multidestination mechanism and the BRCP model are introduced in section 2 together with their intrinsic benefits and the associated deadlock-freedom properties. A hierarchical leader-based scheme is presented in section 3 to perform broadcasting/multicasting on deterministic and adaptive-routed systems. A multiphase greedy scheme is presented in section 4. Simulation experiments and results are presented in section 5. Finally, we conclude the paper with future research directions.

## 2 Multidestination Mechanism and Base-Routing-Conformed-Path (BRCP) Model

In this section, first we discuss wormhole message passing mechanism with multiple destinations. Then we show how multidestination messages can be effectively routed in a wormhole network using available paths conforming to the base routing. Intrinsic benefits of this model together with deadlock-freedom properties and architectural supports are discussed.

### 2.1 Multidestination Wormhole Mechanism

The concept of wormhole message passing mechanism with multiple destinations was first introduced by Lin and Ni in [23]. They used it together with a Hamiltonian-Path-based routing scheme to perform multicast operations. We describe this mechanism briefly in this section. In single destination wormhole message-passing, every message consists of a body and a header with the destination number. For a multidestination message, the header consists of multiple destinations and can span multiple flits. The *sender* node creates the list of destinations as an ordered list, depending on their intended order of traversal, and incorporate it into the header. Different ways to encode such multidestination addresses have been proposed recently [10]. In this paper, we emphasize on all-destination encoding due to its simplicity for implementation.

Once the worm is injected into the network, it is routed based on the address in the leading header flit corresponding to the first destination. As soon as the worm reaches the router of the first destination node, the flit containing this address is removed by the router. Now the worm is routed to the node whose address is contained in the next header flit (second destination in the ordered list). While the flits are being forwarded by the router of the first destination node to its adjacent router, they are also copied (consumed) flit-by-flit to the system buffer by the router-processor interface of the first destination. This process is carried out in each intermediate destination node of the ordered list. When the message reaches the last destination, it is not routed any further and gets completely consumed by the last destination node.

Fig. 1 illustrates an example of this mechanism with 3 neighboring destinations. A typical multidestination worm format with all destination encoding is shown in Fig. 1a. Figure 1b shows a snapshot of the system when flit A of the message has reached the router of node d3. It is assumed that the header flits have been appropriately consumed at the respective nodes prior to this step. Fig. 1c shows the modified system state when the worm moves ahead by one flit. Each flit is copied to the system buffer of the associated node and also propagated ahead to the next node. It is to be noted that such a mechanism is quite powerful and can deliver a message to multiple destinations much faster than using multiple unicast messages. This multidestination scheme is also quite general in the sense that unicast messages can always be implemented under this scheme as a subset operation with only one destination.

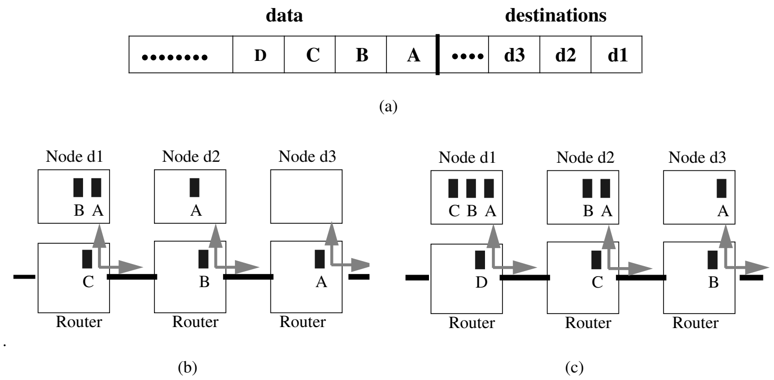


Figure 1: Multidestination wormhole mechanism with a multicast/broadcast worm: (a) a typical format of multidestination worm with all destination encoding, (b) snapshot of the system when flit A of the worm has reached the router of node d3 and (c) snapshot one clock cycle later.

## 2.2 Paths Conforming to Base Routing

The list of destinations in a multidestination worm is ordered by the sender node. This *ordering* depends on two parameters: 1) the *intended traversal* of the message to a set of destinations and 2) the *feasibility* of the intended path to be traversed by the worm in the network. Obviously, the first parameter depends on the application and the second one is determined by the routing constraint of the underlying network. Consider a multidestination worm with $n$ destinations from a source $s$ with an ordered destination list $\{d_1, d_2, \ldots, d_{n-1}, d_n\}$ (the first and the last destinations are $d_1$ and $d_n$, respectively). The validity of this worm depends on the underlying routing scheme. Assume the network supports a deadlock-free routing scheme $\mathcal{R}$ which can be e-cube [13], planar-adaptive [11], turn-model [17], fully-adaptive [15], or any other routing scheme. The Base-Routing-Conformed-Path (BRCP) model is defined as follows:

**Definition 1** *A multidestination worm with an ordered destination list* $\{d_1, d_2, \ldots, d_{n-1}, d_n\}$ *in a network supporting routing scheme* $\mathcal{R}$ *conforms to this base routing iff the destination set* $\{d_1, d_2, \ldots, d_{n-1}\}$ *can be covered as intermediate nodes on one of the possible paths from* $s$ *to* $d_n$ *under the routing constraint* $\mathcal{R}$.

Figure 2 shows examples of multidestination worms under BRCP model. For example, in an ecube system (assuming messages are routed first along row and then along column), a multidestination worm can cover a set of destinations in row/column/row-column order. It is to be noted that a set of destinations ordered in a column-row manner will be an *invalid* path under BRCP model for e-cube systems. Similarly, in a planar adaptive system, a multidestination worm can cover a set of destinations along any diagonal in addition to the flexibility supported by the ecube system. Such additional paths are shown as bold lines in the figure. If the underlying routing scheme supports west first turn model, it can provide further flexibility in covering a lot of destinations using a single worm. For this turn model example, a non-minimal west first routing scheme is assumed. If the base routing scheme supports non-minimal routing then the traversal of multidestination worms can also conform to non-minimal paths. Hence, this model is quite general and can be used by any routing scheme. If the base routing scheme supports non-minimal routing then the traversal of multidestination worms can also conform to non-minimal paths [30]. A similar but restricted method of using only row-path and column-path in e-cube meshes has also been independently proposed recently by Boppana, Chalasani, and Raghavendra [8].

### 2.3 Intrinsic Benefits

The significant benefit of BRCP model comes from the fact that a message can be delivered to multiple destinations with the same overhead as that of sending it to a single destination, if the destinations can be grouped into a single worm under BRCP model. Even the simplest e-cube routing scheme can take advantage of this model by grouping destinations on row, column, or row-column. As the adaptivity of the base routing scheme increases, more and more destinations can be covered by a single multidestination worm. As the number of destinations

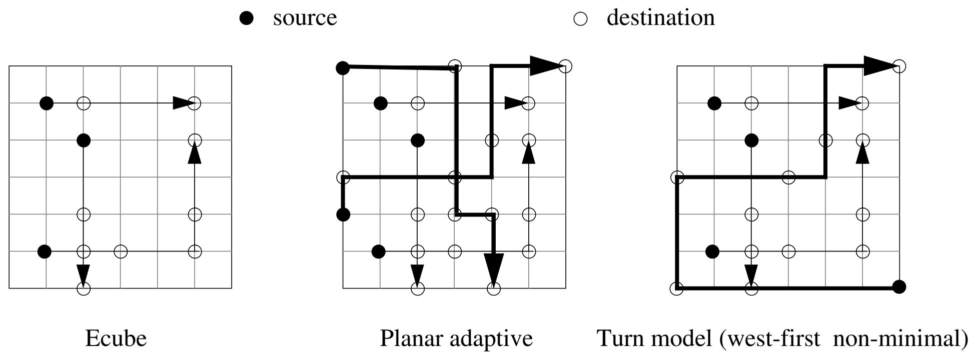


Figure 2: Examples of multidestination worms under the BRCP model conforming to different base routing schemes in a 2-D mesh. In addition to ecube paths, added flexibility for new paths under planar and west first routing schemes are shown in bold lines.

per source *increases* in operations like multicast and broadcast, *less* number of multidestination worms are needed to cover them compared to the unicast-based schemes. If the set of destinations demonstrate some spatial organization (row, column, diagonal, or block) then a suitable scheme based on the underlying routing constraint can be developed to cover them using as few multidestination worms as possible. Similarly, depending on the technological parameters (the ratio of communication start-up time to link-propagation time), a programmer/compiler can optimize on the grouping to implement the required communication operation with minimal time.

Hence, this model opens up an entirely new framework together with its challenges for developing optimal algorithms to implement dense communication like collective communication operations [26] with minimal time. In this paper, we emphasize on this framework by developing and evaluating algorithms for broadcast and multicast which is an important collective communication operation. Before going ahead with developing communication algorithms using the BRCP model, we discuss and present results with respect to deadlock-freedom and the necessary architectural supports to implement this model on $k$-ary $n$-cube systems.

# 3 Deadlock-Freedom and Architectural Supports

In this section, we discuss possible deadlock issues arising out of multidestination wormhole-routing. We evaluate channel dependencies and show that there is possibility of deadlock due to consumption channel dependencies rather than the communication channels. We derive the necessary number of consumption channels to be supported in a system for achieving deadlock-freedom.

### 3.1 Communication Channel Dependency

The BRCP model assumes that the base routing $\mathcal{R}$ is free from deadlock under unicast message passing. A multidestination worm gets routed in a *piece-wise* manner from the source as well as from one of its destination to the next destination. Between neighboring destinations, the worm can take deterministic/adaptive path conforming to the underlying routing scheme. Based on definition 1, the intermediate destinations are chosen in such a way that they can be covered on one of the possible paths from the source to the final destination conforming to the underlying routing scheme. This indicates that a valid multidestination worm does not impose any added communication channel dependency (physical or virtual channel depending on the underlying routing scheme) in the network. Similarly, a multidestination worm with a single destination is equivalent to the traditional unicast worm. Hence, the incorporation of multidestination worms into a network, supporting a given routing scheme, do not create any cyclic dependency of communication channels. This leads to:

**Lemma 1** *Valid multidestination worms satisfying definition 1 can coexist in a network together with unicast worms without leading to deadlock.*

### 3.2 Consumption Channel Dependency

Besides communication channel dependency, it can be noted that a multidestination worm requires the usage of router-processor consumption channel at all of its destinations during its propagation. Hence, added dependency across multidestination worms may arise if each node in the system has limited number of consumption channels. Depending on the types of paths available or used in a system, there might be different deadlock scenarios and requirement of different number of consumption channels. Since e-cube routing paths are always a subset of paths supported by any adaptive routing scheme, we first analyze the consumption channel dependency on e-cube two-dimensional meshes. Later we generalize our result to other topologies and routing schemes.

**Lemma 2** *On an e-cube 2D mesh, only two consumption channels per node are sufficient if either row or column paths are used but not both. Three consumption channels are sufficient if both row and column paths are used. However, supporting all possible paths conforming to e-cube (row, column, and row-column) requires four consumption channels per node.*

**Proof:** First let us consider only row or column paths in an e-cube mesh as shown in Fig. 3a. The worm W1 is trying to cover destination B first and then D (the destination list of each worm shown in the figure is ordered as defined in Fig. 1). Similarly, the worm W2 is trying to cover destinations D first and then B. Consider each node having only one consumption channel. During the propagation of these two multidestination worms, a scenario may arise where the consumption channel of node B is being held by the worm W1 and that of D by the worm W2. Now the worms W1 and W2 need additional consumption channels at nodes D and B, respectively. In the absence of additional consumption channel per node, this situation leads to a deadlock. This can be alleviated by providing two consumption channels per node. Similar solutions have been proposed in [8] for column-path routing. Such argument can also show that the dual-path-based Hamiltonian-routing [24] requires two consumption channels per node.

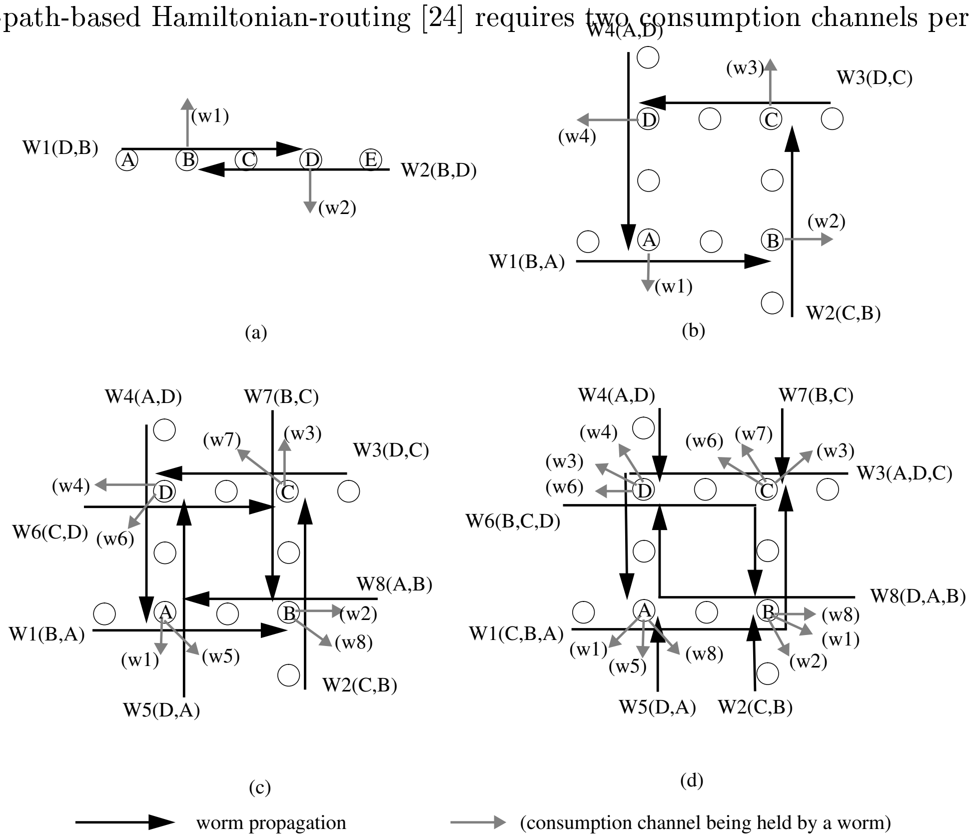


Figure 3: Examples of deadlock due to multidestination worms through consumption channel dependency in 2D mesh with e-cube routing: a) row or column path with one consumption channel/node, (b) row and column paths with one consumption channel/node, (c) row and column paths with two consumption channel/node, and (d) row, column, and row-column paths with three consumption channels per node.

Our proposed BRCP model is more general than the Hamiltonian or column-path routing. Consider both row and column paths being used in an e-cube mesh. With one consumption channel per node, Fig. 3b shows a cyclic dependency between four multidestination worms leading to a deadlock. Such deadlock situation continues to persist, as shown in Fig. 3c, even if two consumption channels per node are available. The problem can be alleviated by allowing three consumption channels per node.

Finally, consider row, column, and row-column paths as defined under the BRCP model for e-cube routing. Figure 3d shows a deadlock situation with four row-column worms(W1–W4) and four column worms (W5–W8). With three consumption channels per node, it can be seen that at node A, the channels have been taken away by worms W1, W5, and W8, respectively before the arrival of worm W3. Similar situation also happens at nodes B, C, and D leading the worms to deadlock. This problem can only be alleviated by having four consumption channels per node. Since the node degree is four for 2D mesh, there can be maximum four messages entering a node at any given time. By using four consumption channels per node, it can be simply seen that a message arriving at a node is always guaranteed a free consumption channel. ■

The above results, derived for 2D mesh supporting e-cube routing, can easily be extended to $k$-ary $n$-cube topology. If the base routing scheme requires $v$ virtual channels per physical channel then there can be maximum $2nv$ worms trying to enter a node at any time. This leads to:

**Theorem 1** *A $k$-ary $n$-cube system with a base routing $\mathcal{R}$ requires $2nv$ consumption channels per node to support deadlock-free multidestination communication under the BRCP model where the base routing $\mathcal{R}$ uses $v$ virtual channels per physical channel to support deadlock-free unicast communication.*

### 3.3 Architectural Supports

The above theorem indicates that only 4 consumption channels per node are required for a 2D e-cube mesh and 8 consumption channels for 2D fully-adaptive system [15]. Similarly, a 3D system will require 6 and 12 consumption channels for e-cube and fully-adaptive routing, respectively. Since these consumption channels are needed to provide deadlock-freedom instead of increasing network throughput [2], they can be configured as *virtual* consumption channels instead of *physical* consumption channels. By restricting multidestination worm movement along some dimension(s) of a system, the requirement of virtual consumption channels can be reduced further as shown in Lemma 2.

Each virtual consumption channel is nothing but a DMA channel. Since the current technology supports multiple (2-4) DMA channels on a single chip, the above requirement translates down to incorporation of 1-2 additional DMA chips at each processor-router interface. Hence, it is a very low cost solution to incorporate these additional channels in a system. It is to be noted that systems like Intel Paragon already have provision to use up to two physical consumption channels for each router while matching the aggregate bandwidth of two consumption channels with that of the processor-memory bus. As the technology is moving towards 64-bit processors with 64/128 bits-wide processor-memory bus and 16/32 bit-wide communication channels, it is not impossible to provide $4-8$ physical consumption channels with more number of virtual consumption channels per node. In addition to providing more consumption channels, for multidestination message passing, the router interface should have logic to concurrently absorb as well as forward flits at destination routers. Current wormhole routers implement absorb logic at final destinations and forward logic at intermediate routers. So very little additional logic is needed to enhance it to support absorb as well as transmit mechanism. Hence, the BRCP Model is a feasible solution for current and near-future generation wormhole-routed systems.

# 4 Multicasting

Wormhole-routed systems are increasingly being used to support distributed memory, shared memory or distributed shared memory programming paradigms. To implement these paradigms, the systems need fast implementation of collective communication and synchronization operations. Multicasting is a very important collective communication pattern and plays an important role in providing program and data distribution in distributed memory systems and cache coherency support in distributed shared memory systems. In this section, we address how efficiently to implement multicasting under the BRCP model in wormhole systems. We provide two schemes to accomplish fast multicasting in such systems.

## 4.1 Hierarchical Leader-based Scheme (HL)

In this section we describe a *hierarchical leader based scheme* for multicast. The scheme is formally described below and specific examples are considered in the subsequent sections.

### 4.1.1 The basic HL scheme

Consider a multicast pattern from a source $s$ with a destination set $D$. Let $L_0$ denote the set $D \cup \{s\}$. The *HL* scheme operates by partitioning this set $L_0$ into disjoint subsets. A node in each subset is designated as the *leader* node for that set such that the leader node can forward a message to the members of its set using a single multidestination worm under the BRCP model. The leaders obtained thus are termed *level* 1 leaders. Let us denote this set of level 1 leaders by $L_1$.

This set $L_1$ is further partitioned into disjoint subsets and leaders are again designated for each subset obtained. Let us call this new set of leaders ( level 2 leaders ) $L_2$. This process of hierarchical grouping is continued as long as it is profitable ( we shall define what we mean by this shortly ) to do so, say for $k$ steps. Eventually, after $m$ grouping steps, we have $m$ sets of level $i$ leaders, $1 \leq i \leq m$. Note that $L_m \subset L_{m-1} \subset \ldots L_1 \subset L_0$.

Once this partitioning has been accomplished, the actual multicast takes place in the following manner : The source performs a unicast-based multicast (Umesh algorithm [25]) to the set $L_m$. (In case the source happens to be the sole level $m$ leader, then this phase is eliminated.) As soon as each leader gets the message, it sends out a series of *multidestination worms* to cover the other destination nodes in its set. Note that a level $i$ leader is also a level $i-1$ leader ($1 \leq i \leq m$ ). Thus, each level $j$ leader ( $1 \leq j \leq m$ ) is actually responsible for sending out $j$ multidestination worms, one for each level in which it is a leader. A single multidestination worm is sent out by the leader at each level to cover all the nodes on its set. The multicast is accomplished as soon as each destination in the set $D$ has received the message. Assuming the grouping has already been done, Fig. 4 gives the pseudo-SPMD-code executed by each node participating in the multicast. It is assumed that the maximum leader level for each node is stored in a local variable L at each node.

```
/* Pseudo code for each participating processor */

multicast()
/*L is the leader level for each node*/
{
If ( my_id == source )
        participate in umesh multicast with level m leaders;
else {
receive message;
if ( L == m) /* Level m leader */
        participate in umesh multicast;
for ( i =L-1; i >=0; i--)
        send out multidestination worm i ; }
}
```

Figure 4: Pseudo-SPMD-code executed by each node participating in the multicast.

The partitioning of the destination set into disjoint subsets can be accomplished in a variety of ways. Here is where the actual power of the HL scheme comes into play. The scheme can take into consideration the *base routing scheme* of the system being used in order to derive the best grouping. For example, if the routing scheme used is *ecube* routing, then the partitioning is done such that the nodes in each set formed lie on a *valid* ecube path. If the system supports some form of adaptive routing then advantage can also be taken of this asset by using a different partitioning scheme. These issues are discussed in greater detail in the subsequent sections.

The grouping/partitioning can be looked upon as a function which takes as input a set of nodes, performs the grouping and returns the set of leader nodes. This function is different for different grouping schemes. Let us define $F_e : L_i \rightarrow L_{i+1}$ to be a function which takes in a destination set $L_i$ as input, performs dimension order ecube grouping, and returns the set of leaders $L_{i+1}$. It should be noted that $L_j \in 2^{L_0}$, ( $0 \leq j \leq m$), where $2^{L_0}$ stands for the power set of $L_0$.

An issue that needs to be addressed is what level of grouping will lead to the best multicast performance or in other words: when is it *profitable* to go for another level of grouping beyond the current one? Note that each level of grouping introduces another *phase* in the multicast. Since each phase is associated with a communication startup time overhead ( $t_s$ ), the goal of the grouping should be to reduce the number of levels as far as possible. At the same time, since the first phase of the multicast is a Umesh based multicast to the highest level leaders, an effort needs to be made to reduce the size of the set $L_m$. This tends to favor larger number of grouping levels. It can be seen that if $\log_2(|L_i|) > \log_2(|L_{i+1}|) + 1$, then it is profitable to go in for another level of grouping. This leads to the following observation:

**Observation 1** *A multicast operation can be achieved in* $(m+1)$ *phases using* $\lceil \log_2(|L_m| + 1) \rceil + m$ *communication start-ups to the destination set* $D$ *where* $m$ *is the number of grouping levels found out by the grouping function. If the source happens to be the sole level* $m$ *leader, then the multicast can take place in* $m$ *phases using* $m$ *communication start-ups.*

### 4.1.2 E-cube Routed 2D Meshes

In this section we consider a system which uses *ecube* routing as the base routing scheme. Consider the mesh illustrated in Fig. 5. We use single dimension grouping along dimension 0 in the first step to obtain the level 1 leaders as shown in the figure. In the next grouping step, the level 1 leaders are grouped along dimension 1 to get the two level 2 leaders shown in the figure. We thus have 6 level 1 leaders and 2 level 2 leaders in this example. ( Remember that each level 2 leader is also a level 1 leader. ) The multicast proceeds as follows : The source uses the Umesh algorithm to send out the message to the two level 2 leaders. In the next phase, these two level 2 leaders send out multidestination worms along dimension 1 to cover the level 1 leaders. In the last phase of the multicast, all the level 1 leaders (including the level 2 leaders ) send out multidestination worms along dimension 0 to cover the remaining destinations. Thus, the multicast terminates after 3 phases and incurs a cost of 4 communication startups.

It should be noted that for the ecube dimension order grouping discussed above, the mul-

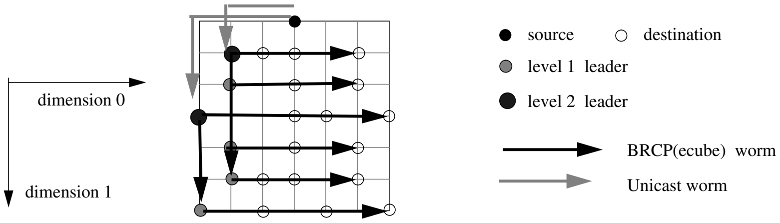


Figure 5: An example multicast in a 2-D mesh using ecube dimension order grouping in hierarchical leader-based scheme.

ticast can be accomplished in at most 3 phases. It does not make sense to use more than 2 ecube dimension order grouping steps, because once all the dimensions have been covered, no benefit can be achieved by invoking the grouping further. Assuming that two stages of grouping are used , the number of communication start-ups needed for the entire multicast is $\lceil \log_2 (|L_2| + 1) \rceil + 2$. It should be noted that for a $k \times k$ mesh, $1 \leq |L_2| \leq k$. Using the notation developed earlier, $L_1 = F_e(L_0), L_2 = F_e(L_1) = F_e(F_e(L_0))$. Table 1 shows the average values for $L_1$ and $L_2$ obtained using the dimension order scheme discussed in this section for different number of destinations. The average is taken over randomly generated destination sets for each destination set size in a $32 \times 32$ mesh. It can be seen from the table that as the number of destinations *increases*, the grouping becomes more and more effective and the value of $L_2$ actually *decreases*. This illustrates a very interesting property of this scheme: the multicast cost can be *decreased* as the destination set size *increases*.

Table 1: Average values for $|L_1|$ and $|L_2|$ for a $32 \times 32$ mesh using dimension order ecube grouping. The rows represent different grouping levels whereas the columns show the number of destinations.

| no. of dests | 64 | 128 | 192 | 256 | 320 | 384 | 448 | 512 | 576 | 640 | 704 | 768 | 832 | 896 | 960 | 1024 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $|L_1|$ | 28.25 | 31.66 | 31.96 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| $|L_2|$ | 18.07 | 14.92 | 12.00 | 9.47 | 8.17 | 7.00 | 6.05 | 5.35 | 4.78 | 4.21 | 3.44 | 3.00 | 2.76 | 2.37 | 1.83 | 1.00 |
| startups for HL | 7 | 6 | 6 | 6 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 3 | 3 |
| startups for umesh | 7 | 8 | 8 | 9 | 9 | 9 | 9 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

### 4.1.3 Taking Advantage of Adaptivity

If the system uses some kind of adaptive routing scheme (eg. planar, Turn model-based, fully), then the grouping can be designed to take of advantage of this additional routing flexibility. The set of valid planar adaptive paths includes the set of valid ecube paths as a subset. Thus, the adaptive grouping can be expected to give at least as good if not better results than simple ecube dimension order grouping. Using the notation developed earlier, this essentially means that $|F_p(L)| \leq |F_e(L)|$, where $F_p$ stands for the grouping function using planar adaptive routing. Similarly, if we consider a grouping function using fully adaptive routing, say $F_f$, then $|F_f(L)| \leq |F_e(L)|$. Further, since the set of fully adaptive paths includes the set of planar adaptive paths as a subset, we also have $|F_f(L)| \leq |F_p(L)|$.

Figure 6 shows the same example used earlier for a system with planar adaptive routing. As can be seen, the set of level 1 leaders is 1 less than the corresponding number obtained using ecube dimension order grouping. However, the number of level 2 leaders is the same as the corresponding number obtained using ecube dimension order routing. This is not surprising, though, since the sets that the grouping functions operate on at the second step are different.

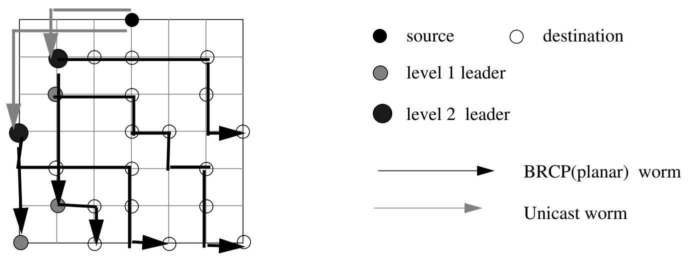


Figure 6: Multicast in a 2-D mesh using planar adaptive grouping.

It should be noted that as opposed to ecube grouping, it *does* make sense to use more than 2 levels of grouping here for a 2-dimensional system. It is a different matter, however, whether these additional grouping steps will lead to any benefits over the groupings already obtained. Whether or not it is profitable to go for more levels of grouping can be verified using the criterion developed earlier. Extra groupings should be done only as long as it results in some benefit in the multicast pattern completion time. In this paper, we use a partial-order graph approach [30] in determining diagonal planar-adaptive paths for a given set of destinations.

## 4.2 Multiphase Greedy Scheme(MG)

In this section, we develop a multiphase greedy scheme. This scheme attempts to minimize the multicast latency by sending as many multidestination messages as possible each step. This is made possible by making the destinations which have received the message act as secondary sources.

### 4.2.1 The basic scheme

Consider a $k \times k$ mesh. Let $s$ be the source and $D$, the set of destinations. Let $c$ be the column having the largest number of destinations. Let the number of destinations in $c$ be $m$. Assume that in the first phase, the source $s$ sends a worm to one of the destinations in $c$, such that it covers this destination and all the destinations above or below this destination in that column. In the next step, this destination can send the message to the remaining destinations in that column by sending another worm. In the next phase, all these destinations along with the original source can act as sources. Thus, we have effectively at least $m+1$ sources in the beginning of the next phase. Consider selecting $m+1$ columns having the largest number of remaining destinations to receive the message in the second phase. Now each of the $m+1$ source can select one of these columns and send the message to a destination in that column. The destination receiving message in each of these columns can forward the message to other destinations in the column. All destinations receiving message in this second phase can act as sources for the third phase. Such process is continued, adding destinations which have received the message to the source set, until there are no more columns left to be covered.

**Theorem 2** *The number of steps required for the basic multiphase greedy scheme in a $k \times k$ mesh with a destination set $D$ is less than or equal to $2 * \lceil log_2 k \rceil + 2$ if $t \leq 1$ and $2 * \lceil log_t k \rceil + 2$ if $t > 1$ where $kt \leq |D| \leq k(t+1)$ and $t$ is an integer.*

**Proof:** Let $S$ be the set of sources at any stage of the algorithm. Each phase in the algorithm takes one or two steps. In the first step, a source sends a multidestination worm to a destination $d$ in a column, which also covers the destinations above or below that point (depending on whichever direction gives more destinations in that column). It can be seen that this worm still conforms to e-cube routing. If there are any destinations in that column which are not covered in the first step, the destination $d$ acts as a source, and sends a worm which covers the remaining destinations in that column in the second step.

Consider the case when $t \leq 1$. If the number of sources at any phase is less than the number of columns remaining to be covered, $|S|$ increases at least by a factor of two at that phase.We can complete the multicast in one more phase when $|S|$ reaches the number of columns having at least one destination to be covered, which is no more than $k$. Each phase requires at most two steps. Thus, the number of steps required for multicast is at most $2 * \lceil log_2 k \rceil + 2$.

Consider now the case where $t > 1$. The column having the largest number of destinations should have at least $t$ destinations by the pigeon hole principle. Thus, at the end of the first phase we add each of these destinations to the set $S$ and $|S|$ would become at least $t$. In the second phase, if $|S| >$ number of columns remaining to be covered, we select the set of $|S|$ columns having the maximum number of destinations. If we do not have $|S|$ columns after the first phase, we select all the remaining columns. Each of the column selected is assigned a source which sends a worm which covers one of the destinations $d$ in that column together with the destinations above or below $d$ as described in the first phase. Also, in the next step, $d$ sends a worm covering the rest of the destinations in that column, if there are any. Each of the destination which receives the message in this phase becomes a source in the next phase.

The selection of the $|S|$ columns is done by sorting the columns in decreasing order of the number of destinations in that column, grouping them into blocks of size $|S|$ and taking the first block. Since we have at least $t$ sources at the end of the first phase, we have $|S| \geq t$ and the size of a block should be at least $t$ and the number of blocks should be at most $k/t$. Using pigeon hole principle, it can be seen that the number of sources after the second phase $\geq$ the number of destinations in the block having the maximum number of destinations $\geq (kt)/(k/t) = t^2$.

Following the above argument, it can be seen that we have at least $t^n$ sources at the end of the $n^{th}$ phase. Once we have enough number of sources to cover all the remaining columns, we can complete the multicast in two more steps.

Thus, the number of phases required to complete the multicast $\leq \lceil log_t k \rceil + 1$ and the number of steps $\leq 2 * \lceil log_t k \rceil + 2$. ■

### 4.2.2 Reducing the number of steps required

It can be seen that the second step of each phase can be overlapped with the first step of the next phase, with around half of the destinations in the columns receiving the message and acting as sources. Thus, for the case where $t > 1$, at the end of the step1 of phase1, we have at least $\lfloor t/2 \rfloor$ sources. Similarly, at the end of the step 2 of phase1 (step1 of phase2), we have at least $\lfloor t^2/2 \rfloor$ sources. Following the same argument, we have at least $\lfloor t^l/2 \rfloor$ sources at the end of step 1 of phase $l$.

Let $s$ be the number of phases required to acquire as many sources as required to cover the remaining columns, with the above modification. Then, at the beginning of the last phase we should have $\lfloor t^s/2 \rfloor \geq c$ where $c$ is the number of columns to be covered at that point. After this, we need at most two more steps to complete the multicast. Thus, the number of steps required for multicast is $r + 2$ where $r$ is equal to the least $s$ satisfying $\lfloor t^s/2 \rfloor \geq c$.

If $t \leq 1$, for all phases except the last one, the number of columns covered gets at least doubled. The step 2 of all phases except the last one can be overlapped with the step 1 of the

Table 2: Average values for the number of steps for a $32 \times 32$ mesh using umesh and multiphase greedy schemes. The columns show the number of destinations.

| no. of dests | 64 | 128 | 192 | 256 | 320 | 384 | 448 | 512 | 576 | 640 | 704 | 768 | 832 | 896 | 960 | 1024 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MG scheme | 3.95 | 3.05 | 3.00 | 3.00 | 3.25 | 3.63 | 3.78 | 3.85 | 3.95 | 3.85 | 3.90 | 3.90 | 3.80 | 3.60 | 3.30 | 2.90 |
| umesh | 7 | 8 | 8 | 9 | 9 | 9 | 9 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

next phase. The last phase requires at most 2 steps. From the above argument, and from the fact that $c$ cannot be more than $k$, we conclude the following:

**Lemma 3** *The number of steps for multilevel greedy multicast for a destination set $D$ in a $k \times k$ mesh with overlapped steps is less than or equal to $\lceil log_t(2k) \rceil + 2$ if $t > 1$ and $\lceil log_2(k) \rceil + 2$ if $t \leq 1$, where $kt \leq |D| \leq k(t+1)$.*

Another optimization can be done as follows: the multicast gets completed along the rows as soon as we have at least one source in every row which has a destination to be covered. An example for multicast using the above scheme is shown in fig 7 and a comparison of the average number of steps for multicast for the Umesh and the multiphase greedy scheme appears in table 2

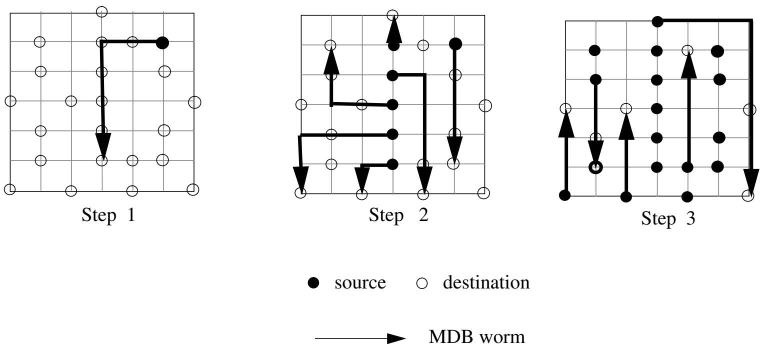


Figure 7: Multicasting using multidestination worms in a 2D mesh under multilevel greedy scheme.

#### 4.2.3 Taking Advantage of Adaptivity

In the ecube routed scheme, there can be contention when two or more sources in the same row tries to send. This can be avoided by using adaptive routing. Further improvement, utilizing adaptivity might be to send worms covering more number of destinations each step conforming to the BRCP model, so that the number of steps required is further reduced. Here, adaptivity provides more flexibility in grouping the destinations, compared to the ecube scheme.

## 4.3 Effect of changes in topology

In the previous section we saw algorithms for multicasting in 2-D meshes. The same algorithms can be modified to work for multidimensional systems, irregular meshes and meshes with wrap-around connections(tori). In this section we see how the schemes discussed above can take advantage of these changes in system topology to accomplish multicast faster.

### 4.3.1 Multidimensional Meshes

The HL scheme developed for 2-dimensional meshes in the previous sections can be easily extended to multidimensional meshes. For a $n$-dimensional system, it is now possible to have upto $n$ ecube dimension order groupings. Assuming $m$ grouping steps are used, the number of communication start-ups incurred in the actual multicast is easily seen to be : $\lceil \log_2 (|L_m| + 1) \rceil + m$.

The MG scheme can be extended for higher dimensions as follows. Consider a $n-$dimensional mesh. The 'columns' in this case are those along the $n^{th}$ dimension. The maximum number of columns in this case would be $k^{n-1}$ where $k$ is the width of the mesh. Thus, the number of steps is bounded by $\lceil \log_t k^{n-1} \rceil + 2$ if $t > 1$ and $\lceil \log_2 k^{n-1} \rceil + 2$ if $t \leq 1$
where $kt \leq$the number of destinations $\leq k(t+1)$, by an analysis similar to that for $2-D$ meshes.

### 4.3.2 Systems with Irregular Topologies

The HL ecube dimension order grouping scheme can take advantage of irregularly shaped meshes. Consider a rectangular 2-dimensional mesh with topology $k_1 \times k_2$, $(k_1 < k_2)$. If grouping is done along the larger dimension first and then along the smaller one, at the end of the first grouping, we have at most $k_1$ leaders. At the end of the next step, we see that we can have at most $k_1$ level 2 leaders. On the other hand if grouping is done in the reverse order we can land up with at most $k_2$ level 2 leaders. It is thus better (on the average) to group along the larger dimension first. Generalizing this, we can see that in order to get the least number of leaders, grouping should be done starting from the largest dimension downwards for a $n$-dimensional irregular mesh.

The MG scheme performs better for rectangular meshes compared to a square mesh of equal size if the larger dimension is along the columns. In this case, the number of destinations covered in each step is likely to be higher and the number of columns to be covered is likely

Table 3: Average values for the number of steps for a $32 \times 32$ mesh and a $32 \times 32$ torus using multiphase greedy scheme. The columns show the number of destinations.

| no. of dests | 64 | 128 | 192 | 256 | 320 | 384 | 448 | 512 | 576 | 640 | 704 | 768 | 832 | 896 | 960 | 1024 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mesh | 3.95 | 3.05 | 3.00 | 3.00 | 3.25 | 3.63 | 3.78 | 3.85 | 3.95 | 3.85 | 3.90 | 3.90 | 3.80 | 3.60 | 3.30 | 2.90 |
| Torus | 3.08 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 2.65 | 2.2 | 2.00 | 2.00 | 2.00 |

to be lower, compared to the square mesh. Both of these tend to reduce the number of steps required for multicast.

#### 4.3.3 Wrap-around Tori

The HL scheme can take advantage of wrap-around connections available in tori to further improve the multicast pattern completion time. For meshes, in ecube dimension order grouping the leaders were chosen to be the leftmost node in the row or the topmost node in the column. This was done so that only a single worm is required to cover each row/column. It is easily seen that this may not always result in the best grouping possible in terms of the number of level $m$ leaders obtained. If the given system has wrap-around connections, then some optimizations can be made. Since in a torus, the system is completely symmetric, the leaders can be chosen anywhere in the row/column. It may be better to choose the leaders in row grouping such that more of these level 1 leaders can be grouped together during level 2 (or column) grouping. Similar criteria can also be chosen for adaptive grouping, the basic idea being to reduce the number of nodes in the final set, say $L_m$, and also the number of phases(levels).

For the MG scheme, in step 1 of each phase of the algorithm, each source can cover a whole column using one worm, and hence requiring only one startup. Also, any worm sent along a row can cover the entire row. Thus, if the algorithm reaches a stage when there exists a set of sources covering all the rows, the multicast can be completed in one more step. Thus, in the worst case, the number of steps required would be $\lceil \log_t k \rceil + 1$ if $t > 1$ and $\lceil \log_2 k \rceil + 1$ if $t \leq 1$, for a $k \times k$ torus where $kt \leq$ number of destinations $\leq k(t+1)$ and $t$ is an integer. It can be seen that this is a substantial improvement over the case with no wrap around. Table 3 shows the average number of steps required for multicast for a $32 \times 32$ mesh and a $32 \times 32$ torus.

## 5 Broadcasting

The *one-to-all broadcast* pattern [9, 3, 20]is a specialized multicast pattern where the message has to be sent to all the nodes in the system. It is clear that the schemes discussed above can be applied for broadcast too, but since this pattern is an important one in actual practice, we discuss how the above described schemes accomplish broadcast in greater detail in this section.

### 5.1 Hierarchical Leader-based Scheme

Using ecube dimension order grouping, we can see that in a 2-D $k \times k$ mesh, there will be exactly $k$ level 1 leaders. Similarly, after level 2 grouping, we always land up with at most 1 level 2 leader. Broadcast is achieved as follows: if the source is not the level 2 leader, in the first phase, it sends out the message to the level 2 leader. This node now sends out a multidestination worm to cover the level 1 leaders, which in turn send out multidestination worms to cover the remaining destinations. Thus, the broadcast can be completed incurring only 3 start-ups. If the source happens to be the level 2 leader, the broadcast can be achieved in at most 2 start-ups. In the first step, the source sends out a multidestination worm to cover the level 1 leaders. In step 2, the level 1 leaders send out multidestination worms to cover the remaining destinations. One-to-all broadcast on a 2-D $7 \times 7$ mesh is illustrated in figure 8. It should be noted that the same applies for irregular 2-D meshes too. However, for 2-D tori, the broadcast can always be accomplished in 2 steps. We thus have the following:

**Observation 2** *One-to-all broadcasting operation in a* 2*-D mesh from any arbitrary source node can be implemented in at most* 3 *phases using at most* 3 *communication startups using ecube dimension order hierarchical grouping. For wrap-around tori, broadcast can be achieved in at most* 2 *communication phases/startups.*

This observation can be generalized for $n$-dimensional meshes as follows:

**Observation 3** *One-to-all broadcasting operation in a $n$-D mesh from any arbitrary source node can be implemented in at most $n+1$ communication phases/startups using ecube dimension order hierarchical grouping. For wrap-around tori, only $n$ communication phases/startups are required.*

### 5.2 Multiphase Greedy Scheme (MG)

One to all broadcast takes two or three steps using multiphase greedy scheme, depending on the position of the source. If the source is on the upper or lower row, it can send one worm to the

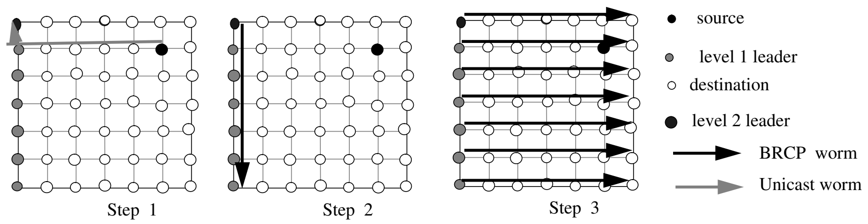


Figure 8: Broadcast in a 2-D mesh using ecube dimension order grouping under hierarchical-leader-based scheme.

leftmost or rightmost column covering that entire column, and in another step, the destinations in that column can cover the entire mesh by sending a worm along their respective rows. If the source not in the upper or lower row, broadcast takes three steps as shown in fig 9.

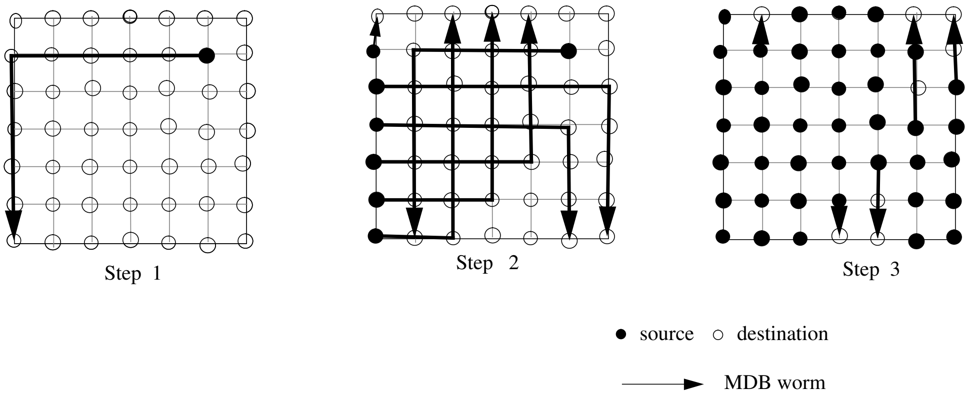


Figure 9: Broadcasting using multidestination worms in a 2D mesh under Multiphase greedy scheme.

# 6 Simulation Experiments and Results

To verify the effectiveness of our new BRCP model, we simulated single-source multicasting on a CSIM-based [32] simulator test bed. This test-bed implements flit-level simulation and can be configured for different topological and technological parameters. We compared our scheme with Hamiltonian Path-based scheme [24] and Umesh algorithm [25]. For all simulation experiments, we assumed system parameters representing the current trend in technology. The following parameters were used: $t_s$ (communication start-up time) as 1.0 microseconds, $t_p$ (link propagation time) as 0.005 nanoseconds for 200 Mbytes link, and $t_{node}$ (router delay) as 20.0 nanoseconds. For all multicast operations, message size was kept constant at 50 flits. Random sources and destinations were chosen for each multicast operation. Each experiment was carried out 40 times to obtain average results. To obtain a fair comparison, $2n$ consumption channels were assumed at each node, irrespective of the base routing schemes being used. It is to be noted that single-source multicast operations are in fact insensitive to the number of consumption channels because there is no node contention in the system. Below we study the impact of the BRCP model against other schemes for varying topology and routing schemes.

## 6.1 Overall Comparison for 2D Meshes with E-cube Routing

Figure 10 compares the cost of single-source multicast against the cost incurred by the Umesh and the Hamiltonian schemes. For all system configurations, schemes under the BRCP model outperform the Umesh by a factor of 2-3. For bigger system sizes, our schemes do better than the Hamiltonian scheme by a factor of 2-4. For all system configurations, the multiphase greedy scheme (mg) does better than the hierarchical leader-based scheme (hl). This is because the multiphase greedy scheme is able to take advantage of multidestination message passing in each phase where as it is restricted to later phases in the leader-based scheme. Currently, we are investigating on a variation of the leader-based scheme to take advantage of multidestination message passing in its earlier phases too. This will try to reduce the cost of the leader-based scheme equal to or less than the multiphase greedy scheme.

Besides the above improvement factors, the most *interesting* and *unique* aspect of these results is that using the BRCP model, the cost of multicast operation in a system can be *reduced* as the degree of multicast *increases*. It can also be noted that the multiphase greedy scheme allows the cost of multicast to remain *near-constant* over a wide range of destinations as the system size increases. The underlying strength behind these schemes is to use multidestination message passing mechanism which allows us to forward messages to more and more destinations rapidly. The multicasting schemes try to achieve better and better grouping with increase in the number of destinations. For less number of destinations, the sparsity is high and the schemes behave equally to that of the Umesh or the Hamiltonian schemes.

The above results indicate that the BRCP model can be used to develop algorithms for fast implementation of other collective communication patterns (barrier synchronization, parallel prefix, reductions, etc.) in wormhole networks supporting distributed memory systems. Similarly, it can be used for implementing cache-coherency (multicast messages doing invalidation/updation) with constant overhead in wormhole systems supporting distributed-shared memory. In the next step of our research, we are considering these directions.

## 6.2 Effect of Adaptivity

We studied the effect of adaptivity to reduce the cost of multicast under the BRCP model. Meshes with e-cube and planar-adaptive routing were considered for comparison. The hierar-

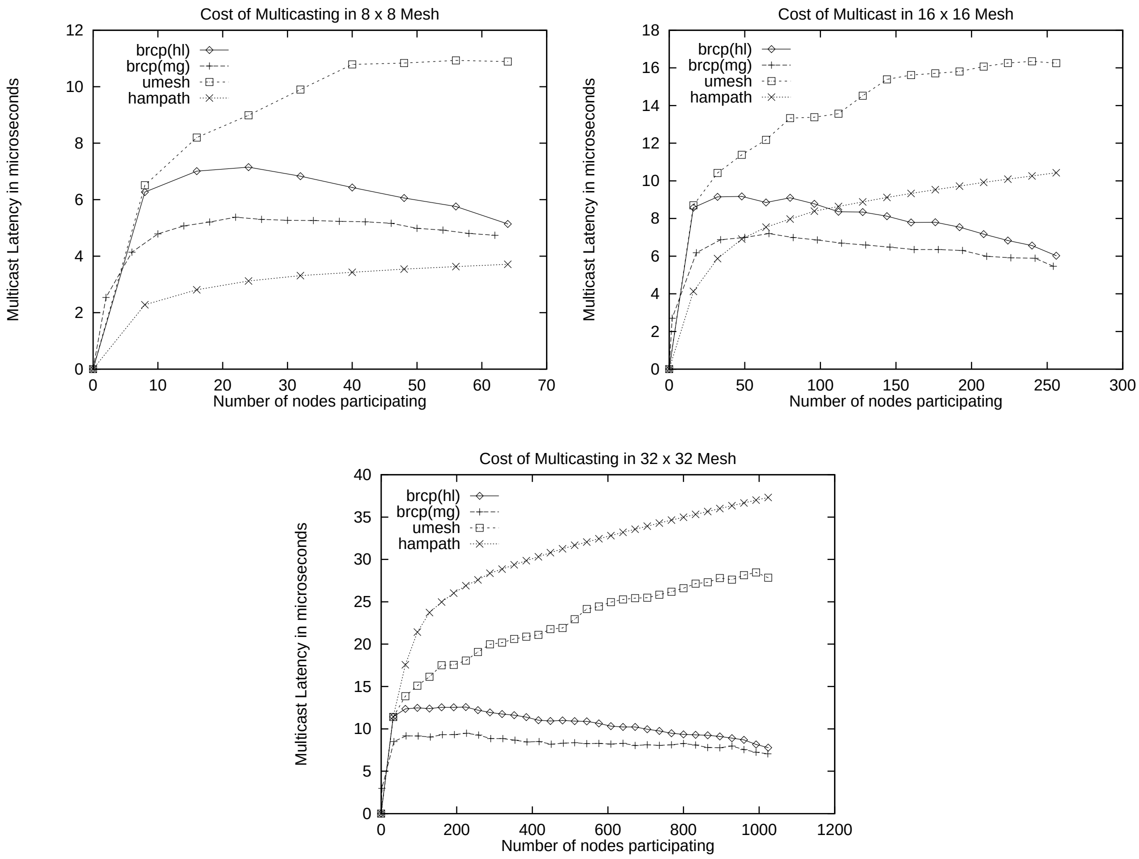


Figure 10: Comparison of the BRCP model-based multicasting schemes (hierarchical leader-based (hl) and multiphase greedy (mg)) with Umesh and Hamiltonian Path-based schemes on various 2-D meshes.

chical leader-based scheme (hl) demonstrates flexibility to invoke different *grouping* sequences as mentioned in section 3.1. For a 2D mesh, we considered the following grouping sequences: $(e, e)$, $(p, p)$, and $(e, e, p)$; where $e$ and $p$ stand for e-cube and planar-adaptive grouping, respectively. It is to be noted that these groupings under the BRCP model try to take advantage of the underlying routing scheme to cover as many destinations as possible using a single worm. The multiphase greedy scheme (mg), as discussed in section 3.2, tries to cover the destinations in the minimum number of phases possible. Depending on the phases and the underlying routing scheme, some worms may encounter contention. So we just simulated the same multiphase greedy algorithm on systems with e-cube or planar-adaptive routing and compared their cost.

Figure 11 shows the effect of adaptivity on both schemes for two different mesh sizes. It is to be noted that $(e, e, p)$ grouping performs better than $(e, e)$ grouping for hierarchical leader-based scheme. Here the increase in adaptivity is being used to cover a set of leaders through paths conforming to planar-adaptive routing, which are not feasible under e-cube routing. An interesting observation can be made here that the $(p, p)$ grouping in fact performs worse than the $(e, e)$ grouping. In our implementation of $p$ grouping we have only considered the flexibility of diagonal paths provided by planar-adaptive routing and have not considered the row/column/row-column paths provided by the e-cube routing. Hence, this does not necessarily reduce the number of leaders. An optimization providing flexibility for all possible paths in $p$ grouping will try to reduce the cost as the *minimum* of the costs exhibited by the $(e, e)$ and $(p, p)$ groupings. So the adaptivity can be used to reduce the cost of multicast when the number of destinations is small. For the multiphase greedy scheme, there is little improvement in multicast cost with increase in adaptivity. The improvement is mostly with less number of destinations. With more number of destinations, the scheme is powerful enough to undergo less number of phases without contention in e-cube system and hence there is not much improvement with adaptivity.

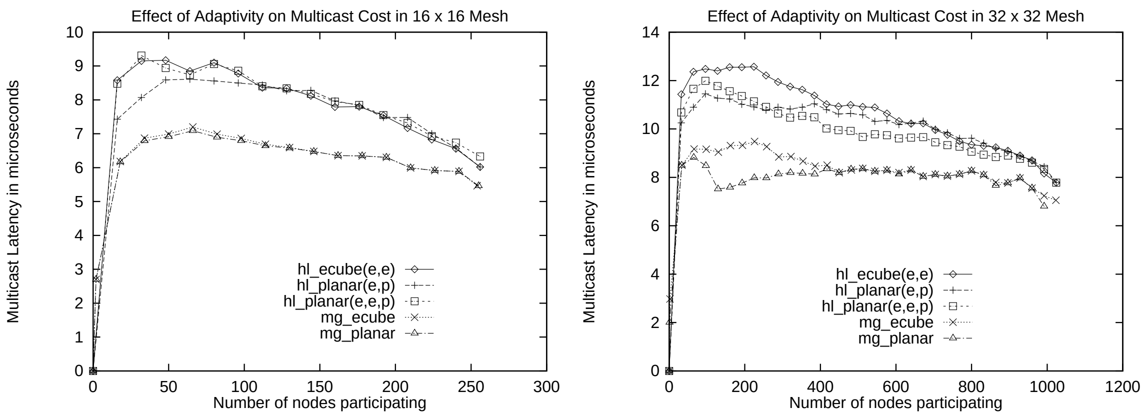


Figure 11: Reducing the cost of multicasting by increase in adaptivity in 2D meshes.

## 6.3 Comparison for 3D Meshes

Figure 12 shows how the hierarchical leader-based scheme performs in 3D meshes compared to other schemes. Here we also see the trend that as the number of destinations increases we can do better multicasting. The hierarchical leader-based scheme is able to perform multicast with a reduced cost by factors of 2-4 and 2-6 compared to the Umesh and the Hamiltonian-path schemes, respectively.

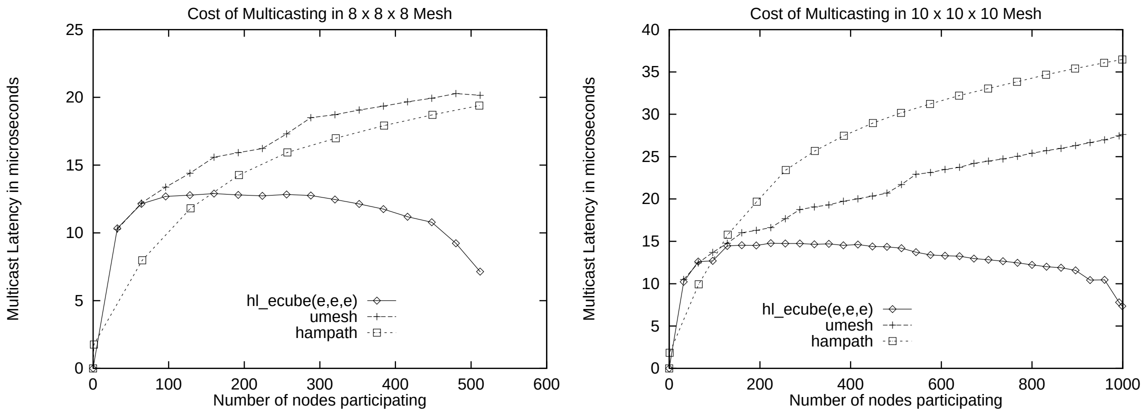


Figure 12: Comparing cost of multicasting in 3D meshes.

## 6.4 Effect of Irregular Topology

Since the BRCP-model schemes use multidestination worms they can take advantage of irregular topologies. Figure 13 shows how these schemes perform with respect to irregularities to perform multicast in 16x16 and 32x8 systems with equal number of processors. It can be seen that both hierarchical-leader (hl) and multiphase greedy (mg) schemes are able to take advantage of adaptivity in performing multicast with reduced cost in 32x8 system than the 16x16 system. The reason behind these is that they can cover the destinations in the longer dimension of the system with a single worm. Such flexibility is totally absent in Umesh scheme. In fact, for irregular systems, the Umesh scheme performs little worse because the path lengths are higher and it increases the propagation cost of some worms. In fact, with more number of destinations, the hierarchical-leader scheme is able to do better grouping to reduce the number of communication start-ups. But the cost increases slightly because of longer path lengths. For systems with higher communication start-up time, these effects will be absent. Hence, the BRCP-model schemes will be able to take advantage of irregularity in topology to perform multicast faster.

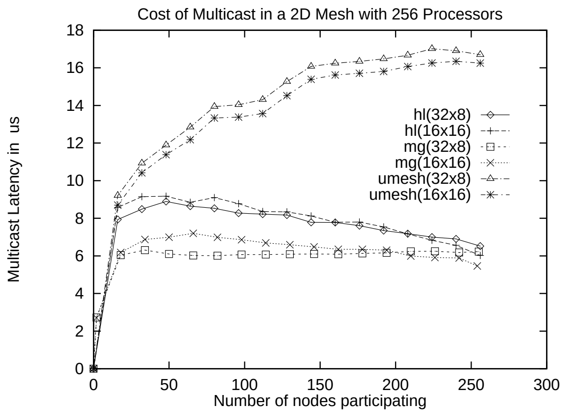


Figure 13: Comparing cost of multicasting in regular and irregular 2D meshes with equal number of processors.

### 6.5 Effect of Wraparound

We studied the impact of wrap-around on reducing the cost of multicast. In the presence of wrap-around, the multidestination worms have more flexibility to cover more destinations using a single worm. Such flexibility can be used to reduce the multicast cost in tori as shown in Fig. 14. With large number of destinations, the multiphase greedy scheme (mg) is able to reduce the cost of multicast by 20-30%. The hierarchical leader-based scheme also has flexibility to take advantage of wrap-around to reduce the cost of multicast, as discussed in section 3.3.3. We are incorporating these flexibilities to our simulator now and will be presenting the results in the final version of the paper.

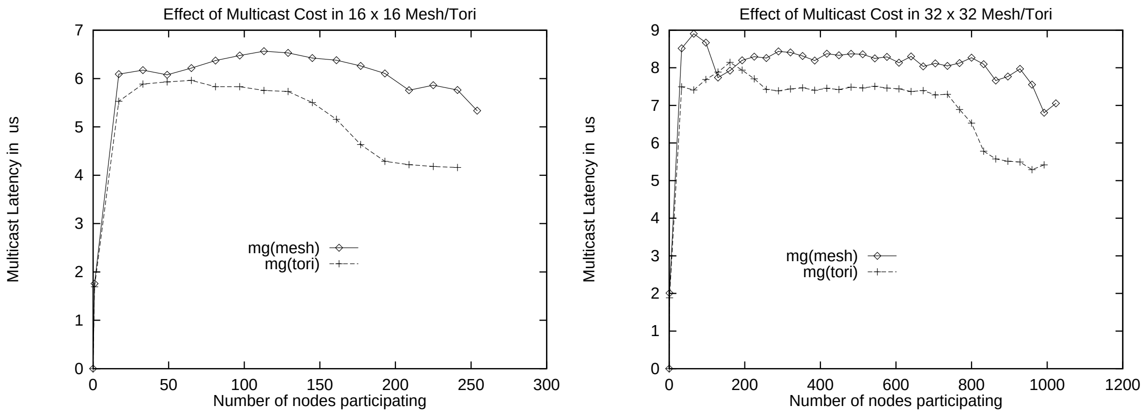


Figure 14: Comparing cost of multicasting in 2D meshes and tori.

### 6.6 Scalability with Increasing System Size

We studied the scalability issues of our multidestination scheme as the system size grows and compared it with other schemes like Umesh and Hamiltonian Path. Figure 15 shows the comparison results for three different system sizes: 8x8, 16x16, and 32x32. Results for both high (90%) and medium (50%) degree of multicasting was derived. It can easily be observed that both hierarchical and multiphase schemes using multidestination mechanisms are much more scalable than Umesh and Hamiltonian path schemes. When the system size grows from 8x8 to 32x32, the cost of implementing multicast to 501.6 using multidestination mechanism compared to 2.3 using Umesh and 9.57 using Hamiltonian path schemes. These numbers clearly shows the scalability potential of the proposed multidestination schemes.

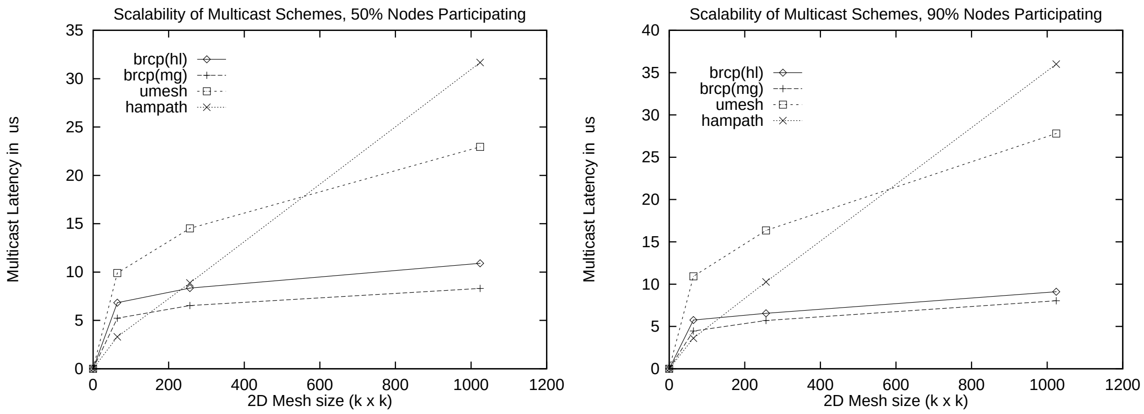


Figure 15: Comparing cost of multicasting with increase in system size.

## 7 Conclusions and Future Research

In this paper we have introduced a new BRCP-model of wormhole routing which uses multidestination worm message passing mechanism conforming to the base routing available in a system. The power of this model is illustrated by developing and presenting two schemes (hierarchical leader-based and multiphase greedy) to implement multicast and broadcast operations in e-cube meshes 2-6 times faster compared to the existing Hamiltonian-path and Umesh schemes in the literature. It has been shown that the requirement of more number of consumption channels per node to provide deadlock-freedom under this model can easily be supported by the current and near-future technology. The power of this model to take advantage of adaptivity, irregularity in system topology, and spatial distribution of destination nodes to reduce the cost of multicast is demonstrated through examples and simulation results.

The presented model provides the following *unique* and *interesting* results: 1) the cost of multicast can be *reduced* or kept *near-constant* as the degree of multicast *increases* and 2) in addition to increasing the throughput of a system by increase in *adaptivity*, the latency of multicast can also be *reduced* by taking advantage of adaptivity. Such results are the first ones in the literature. These results provide us the following strong guideline in developing future high-performance systems using wormhole-routing: *future systems should provide multidestination message passing by default to implement dense communication and synchronization operations faster and the unicast message passing can always be implemented as a subset operation of the multidestination message-passing.*

In this paper, we have studied the impact of the BRCP model on developing and evaluating schemes for single-source multicast pattern. We are extending our work to other collective communication patterns like multiple-source multicast, barrier-synchronization, complete-exchange, parallel-prefix, and reductions, etc. A similar multidestination approach is being taken in defining new rendezvous [18] and multirendezvous [28] synchronization primitives to implement barriers with minimal cost in k-ary n-cube wormhole networks. Since multicast is a basic mechanism for implementing cache-coherency in distributed-shared-memory systems, it will be interesting to see how the BRCP model with reduced or near-constant cost for multicasting can be effectively used to provide fast cache-coherency on wormhole networks. Since the BRCP model has potential to implement dense communications faster, it will also be interesting to see how it can help in building scalable wormhole systems with clustered configurations [5]. From the perspectives of distributed-memory systems, we are also investigating the impact of the multidestination mechanism to reduce program execution time of programs using multicast communication [29]

**Acknowledgements:** The author would like to thank members of his research group for providing comments, criticisms, and suggestions to this work. Special thanks go to Prof. Rajendra Boppana of Univ. of Texas, San Antonio and Prof. Suresh Chalasani of Univ. of Wisconsin for discussion on the deadloack-freedom properties of the BRCP model.

## References

[1] V. S. Adve and M. K. Vernon. Performance Analysis of Mesh Interconnection Networks with Deterministic Routing. *IEEE Transactions on Parallel and Distributed Systems*, 5(3):225–246, 1994.

[2] S. Balakrishnan and D. K. Panda. Impact of Multiple Consumption Channels on Wormhole Routed k-ary n-cube Networks. In *Proceedings of the International Parallel Processing Symposium*, pages 163–167, April 1993.

[3] A. Bar-Noy and S. Kipnis. Multiple Message Broadcasting in the Postal Model. In *International Parallel Processing Symposium*, pages 463–470, 1993.

[4] M. Barnett, D. G. Payne, and R. Van de Geijn. Optimal Broadcasting in Mesh-Connected Architectures. Technical Report TR91-38, Dept. of Computer Science, Universityof Texas at Austin, Dec 1991.

[5] D. Basak and D. K. Panda. Scalable Architectures with k-ary n-cube cluster-c Organization. In *Symposium on Parallel and Distributed Processing*, pages 780–787, 1993.

[6] S. H. Bokhari and H. Berryman. Complete Exchange on a Circuit-Switched Mesh. In *Proceedings of the Scalable High Performance Computing Conference*, pages 300–306, 1992.

[7] R. V. Boppana and S. Chalasani. A Comparison of Adaptive Wormhole Routing Algorithms. In *International Symposium on Computer Architecture*, pages 351–360, 1993.

[8] R. V. Boppana, S. Chalasani, and C. S. Raghavendra. On Multicast Wormhole Routing in Multicomputer Networks. In *Symposium on Parallel and Distributed Processing*, 1994. to appear.

[9] J. Bruck, R. Cypher, and C.-T. Ho. Multiple Message Broadcasting with Generalized Fibonacci Trees. In *Symposium on Parallel and Distributed Processing*, pages 424–430, 1992.

[10] C.-M. Chiang and L. M. Ni. Multi-Address Encoding for Multicast. In *Proceedings of the Parallel Computer Routing and Communication Workshop, Lecture Notes in Computer Science, Springer-Verlag*, 1994. to appear.

[11] A. A. Chien and J. H. Kim. Planar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors. In *Proceedings of the International Symposium on Computer Architecture*, pages 268–277, 1992.

[12] Cray Research, Inc. *Cray T3D System Architecture Overview*, 1993.

[13] W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, pages 547–553, May 1987.

[14] J. Duato. A New Theory of Deadlock-Free Adaptive Multicast Routing in Wormhole Networks. In *Symposium on Parallel and Distributed Processing*, pages 64–71, 1993.

[15] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1320–1331, 1993.

[16] B. Duzett and R. Buck. An Overview of the Ncube-3 Supercomputer. In *Proceedings of the Frontiers of Massively Parallel Computation*, pages 458–464, 1992.

[17] C. J. Glass and L. Ni. The Turn Model for Adaptive Routing. In *Proceedings of the International Symposium on Computer Architecture*, pages 278–287, 1992.

[18] S. K. S. Gupta and D. K. Panda. Barrier Synchronization in Distributed-Memory Multiprocessors using Rendezvous Primitives. In *Proceedings of the International Parallel Processing Symposium*, pages 501–506, April 1993.

[19] Intel Corporation. *Paragon XP/S Product Overview*, 1991.

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

[21] D. Lenoski et al. The Stanford DASH Multiprocessor. *IEEE Computer*, pages 63–79, March 1992.

[22] X. Lin, A.-H. Esfahanian, P. K. McKinley, and A. Burago. Adaptive Wormhole Routing in Hypercube Multicomputer. In *Symposium on Parallel and Distributed Processing*, pages 72–79, 1993.

[23] X. Lin, P. K. McKinley, 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.

[24] X. Lin, P. K. McKinley, and L. M. Ni. Performance Evaluation of Multicast Wormhole Routing in 2D-Mesh Multicomputers. In *Proceedings of the International Conference on Parallel Processing*, pages I:435–442, 1991.

[25] P. K. McKinley, H. Xu, A.-H. Esfahanian, and L. M. Ni. Unicast-based Multicast Communication in Wormhole-routed Direct Networks. In *Proceedings of the International Conference on Parallel Processing*, pages II:10–19, 1992.

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

[27] L. Ni and P. K. McKinley. A Survey of Wormhole Routing Techniques in Direct Networks. *IEEE Computer*, pages 62–76, Feb. 1993.

[28] D. K. Panda. Optimal Phase Barrier Synchronization in k-ary n-cube Wormhole-routed Systems using Multirendezvous Primitives. In *Workshop on Fine-Grain Massively Parallel Coordination*, pages 24–26, May 1993.

[29] D. K. Panda and V. A. Dixit-radiya. Message-Ordering for Wormhole-Routed Multiport Systems with Link Contention and Routing Adaptivity. In *Scalable High Performance Computing Conference*, pages 191–198, 1994.

[30] D. K. Panda and P. Prabhakaran. Multicasting using Multidestination-Worms Conforming to Base Wormhole Routing Scheme. Technical Report OSU-CISRC-11/93-TR37, Dept. of Computer and Information Science, The Ohio State University, 1993.

[31] G. D. Pifarre, L. Gravano, S. A. Felperin, and J. Sanz. Fully Adaptive Minimal Deadlock-Free Routing in Hypercubes, Meshes, and Other Networks: Algorithms and Simulations. *IEEE Transactions on Parallel and Distributed Systems*, 5(3):247–263, 1994.

[32] H.D. Schwetman. Using CSIM to Model Complex Systems. In *Proceedings of the 1988 Winter Simulation Conference*, pages 246–253, 1988.

[33] D. Scott. Efficient All-to-all Communication Patterns in Hypercube and Mesh Topologies. In *Proceedings of the Symposium on Parallel and Distributed Processing*, pages 398–403, 1991.

[34] H. Xu, P. K. McKinley, and L. Ni. Efficient Implementation of Barrier Synchronization in Wormhole-routed Hypercube Multicomputers. *Journal of Parallel and Distributed Computing*, 16:172–184, 1992.