Balancing Traffic Load for Multi-Node Multicast in a Wormhole 2D Torus/Mesh

San-Yuan Wang*, Yu-Chee Tseng*, Ching-Sung Shiu, and Jang-Ping Sheu*

Department of Computer Science and Information Engineering
National Central University, Chung-Li 32054, Taiwan
*E-mail: {sywang, yctseng, sheujp }@csie.ncu.edu.tw

Abstract

This paper considers the multi-node multicast problem in a wormhole-routed 2D torus/mesh, where an arbitrary number of source nodes each intending to multicast a message to an arbitrary set of destinations. To resolve the contention and the congestion problems, we propose to partition the network into subnetworks to distribute, and thus balance, the traffic load among all network links. Several ways to partition the network are explored. Simulation results show significant improvement over existing results for torus and mesh networks [2, 3, 5].

1. Introduction

In a multicomputer network, processors often need to communicate with each other for various reasons, such as data exchange and event synchronization. Efficient communication is critical for high-performance computing. This is especially true for those *collective communication patterns*, such as *broadcast* and *multicast*, which involve more than one source and/or destination.

This paper considers the *multi-node multicast* problem in a 2D torus/mesh with wormhole, dimension-ordered, and one-port routing capability[1]. There are an arbitrary number of source nodes each intending to send a multicast message to an arbitrary set of destination nodes. We approach this problem by using multiple unicasts to implement multicast. The challenge is that there may exist serious contention when the source set or destination set is large or when there exists hot-spot effect (i.e., sources and/or destinations concentrate in some particular area). To resolve the contention problem, we apply two schemes: *network partitioning* and *load balancing*. We first partition the network into a number of "subnetworks" and then evenly distribute these multicasts, by re-routing them, to these subnetworks, with the expectation of balancing the traffic load among all network links.

Our work is not to propose a completely brand-new scheme, in the sense that after a torus/mesh is partitioned, the obtained subnetworks are each a "dilated" network still maintaining a similar torus/mesh topology. Thus, it is possible to apply the best available multicast schemes on these subnetworks. The details are in Section 2, where several ways to partition the torus/mesh are proposed. It is worth noting that the network-partitioning idea was originally proposed by the same authors in [7] and [8] for single-node broadcast and single-node multicast, respectively. The contribution of this paper is in extending its applicability to multi-node multicast, demonstrating its capability to balance load, and exploring more ways to partition a torus/mesh. Through extensive simulations, we justify that our network-partitioning approach can achieve better load balance and reduce multicast latency[2, 3, 5].

2. Preliminaries

2.1. Network Model

A wormhole-routed multi-computer network consists of a number of computers (nodes) each with a separate *router* to handle its communication tasks [4]. From the connectivity between routers, we can define the topology of a wormhole-routed network as a graph $G = (V, C)$, where V is the node set and C specifies the channel connectivity. We assume the *one-port model*, where a node can send, and simultaneously receive, one message at a time.

A message is partitioned into a number of *flits* to be sent in the network. The *header* flit governs the routing, while the remaining flits simply follow the header in a pipelined fashion. In the contention-free case, the communication latency for sending a message of L bytes is commonly modeled by $T_s + LT_c$ [4], where T_s is the *startup time* (for initializing the communication) and T_c is the *transmission time* per byte. Also, we consider networks that are connected as torus or mesh. Due to the space limitation, we omit the presentation about meshes.

2.2. Subnetworks of a Wormhole Network

Definition 1 Given a wormhole network $G = (V, C)$, a *subnetwork* $G' = (V', C')$ of G is one such that $V' \subseteq V$ and $C' \subseteq C$.

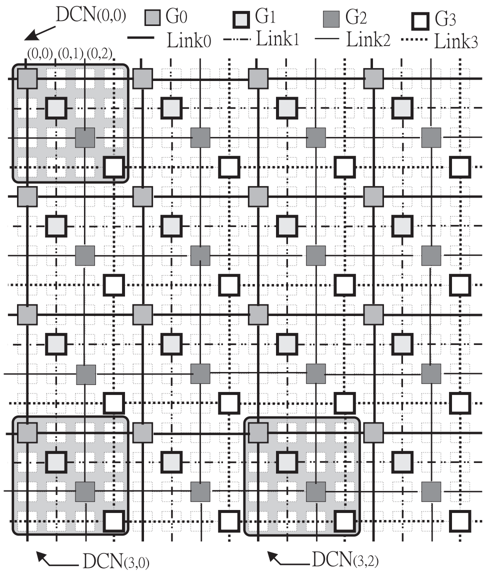


Figure 1. Four dilated-4 subnetworks, each as an undirected 4×4 torus, in a 16×16 torus.

For instance, Fig. 1 shows four subnetworks, G_i, $i = 0..3$, in a 16×16 torus. There are some subtleties in the above definition that need of special attention:

- A subnetwork is not necessarily a "graph" in standard graph theory. Specifically, suppose channel $(x, y) \in C'$. Then the vertices x and y are not necessarily in the vertex set V'. For instance, in Fig. 1, the subnetwork G_0 contains links $(p_{0,0}, p_{0,1})$ and $(p_{0,1}, p_{0,2})$. However, only node $p_{0,0}$ is in G_0's node set.

- The previous point in fact carries special meanings for wormhole routing. For instance, each G_i in Fig. 1 can be considered as a 4×4 torus, with each link "dilated" by four links. However, the dilated torus can work almost like an ordinary torus, since communication in wormhole routing is known to be quite distance-insensitive.

- A subnetwork, though capable of using all links in its link set, should be constrained in its capability in initiating/retrieving packets into/from the subnetwork subject to its node set. For instance, in Fig. 1, nodes $p_{0,1}$ and $p_{0,2}$ of G_0 are neither allowed to initiate a new worm into, nor allowed to retrieve a pass-by worm from, the subnetwork. They can only passively relay worms it receives according to the routing function.

Our approach in this paper is to use multiple subnetworks in a torus to balance the communication load in different parts of the torus, thus eliminating congestion and hot-spot effects. This is of importance particular for massive communication problems such as multi-node multicast. This leads to an important issue of making each subnetwork less dependent of other subnetworks, as formulated in the following definition.

Definition 2 Given two subnetworks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, G_1 and G_2 are said to be *node-contention-free* if $V_1 \cap V_2 = \emptyset$, and *link-contention-free* if $E_1 \cap E_2 = \emptyset$.

Definition 3 Given a set of subnetworks $G_1, G_2, \ldots, G_k$, the *level of node contention* (resp., *level of link contention*) among these subnetworks is defined to be the maximum number of times that a node (resp., link) appears in these subnetworks, among all nodes (resp., links) in the network.

2.3. A General Model for Multi-Node Multicasts

A multi-node multicast instance can be denoted by a set of 3-tuple $\{(s_i, M_i, D_i), i = 1..m\}$. There are m source nodes $s_1, s_2, \ldots, s_m$. Each $s_i, i = 1..m$, intends to multicast a message M_i to a set D_i of destinations.

Next, we derive a general approach to multi-node multicast based on the concept of subnetworks. Given any network G, we construct from G two kinds of subnetworks: *data-distributing networks (DDNs)* and *data-collecting networks (DCNs)*. Suppose we have α DDNs, $DDN_0, DDN_1, \ldots, DDN_{\alpha-1}$, and β DCNs, $DCN_0, DCN_1, \ldots, DCN_{\beta-1}$. We require the following properties in our model:

P1: $DDN_0, DDN_1, \ldots, DDN_{\alpha-1}$ together incur on each node about the same level of node contention, and similarly on each link about the same level of link contention.

P2: $DCN_0, DCN_1, \ldots, DCN_{\beta-1}$ are disjoint and they together contain all nodes of G.

P3: DDN_i and DCN_j intersect by at least one node, for all $0 \le i < \alpha$ and $0 \le j < \beta$.

Now given a problem instance $\{(s_i, M_i, D_i), i = 1..m\}$, a general approach is derived as follows.

Phase 1: Each multicast $(s_i, M_i, D_i), i = 1..m$, selects a target data distribution network, say, DDN_a to distribute its message. The selection should be done with *load balance* in mind. Then s_i chooses a node $r_i \in DDN_a$ as a representative of s_i in DDN_a and sends M_i to r_i.

Phase 2: From node r_i, perform a multicast (r_i, M_i, D'_i) on DDN_a, where the destination set D'_i is obtained from D_i by the following transformation. For each $DCN_b, b = 0..\beta - 1$, if DCN_b contains one or more destination nodes in D_i, then select any node $d \in DDN_a \cap DCN_b$ (by **P3**) as the representative of the recipients of message M_i in DCN_b. Then we join d into D'_i.

Phase 3: In each $DCN_b, b = 0..\beta - 1$, after the representative node d receives M_i, it performs another multicast $(d, M_i, D_i \cap DCN_b)$ on the subnetwork DCN_b.

The following two properties are not a necessity, but would offer regularity in designing phases 2 and 3.

P4: $DDN_0, DDN_1, \ldots, DDN_{\alpha-1}$ are isomorphic.

P5: $DCN_0, DCN_1, \ldots, DCN_{\beta-1}$ are isomorphic.

In the next section, we will discuss how to define the DDNs and DCNs in tori and meshes that satisfy our needs.

3. Subnetworks of a 2D Torus

3.1. DDN's and DCN's in a 2D Torus

A 2D torus $T_{s\times t}$ consists of $s \times t$ nodes each denoted as $p_{x,y}$, where $0 \le x < s$ and $0 \le y < t$. Node $p_{x,y}$ has a link connected to each of $p_{(x\pm1) \bmod s, y}$ and $p_{x,(y\pm1) \bmod t}$.

Definition 4 Given a torus $T_{s\times t}$ and any integer h that divides both s and t, define h subnetworks $G_i = (V_i, E_i), i = 0..h-1$, such that:

$$
\begin{aligned}
V_i &= \{p_{x,y} | x = ah + i,\ y = bh + i, \\
&\quad \text{for all } a = 0..\frac{s}{h} - 1 \text{ and } b = 0..\frac{t}{h} - 1\} \\
C_i &= \{\text{all channels at rows } ah + i \\
&\quad \text{and at columns } bh + i\}.
\end{aligned}
$$

Intuitively, G_0 contains all nodes at the intersection of rows ah and columns bh, and G_i is obtained from G_0 by shifting G_0's nodes by i positions on both indices. In our terminology, each subnetwork is a "dilated-h" torus of size $(s/h) \times (t/h)$. Fig. 1 shows an example, with four subnetworks (each as a dilated-4 4×4 torus) in a 16×16 torus.

Lemma 1 *The subnetworks $G_i, i = 0..h-1$, defined in Definition 4 are free from both node and link contention.*

Observe that in Definition 4, all links in the original torus have been used, so it is impossible to add more subnetworks without increasing link contention. However, there are still some nodes (e.g., nodes $p_{1,0}$ and $p_{0,1}$) that are not included in any subnetwork.

Definition 5 Given a torus $T_{s\times t}$ and any integer h that divides both s and t, define h^2 subnetwork $G_{i,j} = (V_{i,j}, E_{i,j}), i, j = 0..h-1$, such that:

$$
\begin{aligned}
V_{i,j} &= \{p_{x,y} | x = ah + i,\ y = bh + j, \text{ for all} \\
&\quad a = 0..\frac{s}{h} - 1 \text{ and } b = 0..\frac{t}{h} - 1\} \\
C_{i,j} &= \{\text{all channels at rows } ah + i \\
&\quad \text{and at columns } bh + j\}.
\end{aligned}
$$

Lemma 2 *The h^2 subnetworks $G_{i,j}, i, j = 0..h-1$, defined in Definition 5 are free from node contention, but have link contention of h.*

In the above definition, every node and every link have been used by some subnetwork(s), so it is impossible to add more subnetworks without increasing node and link contentions. However, we have only considered subnetworks with *undirected* links. With duplex capability, an undirected link can be regarded as two *directed* links in opposite directions. If we allow such separation, further improvement is possible. Let's call a direct link a *positive* link if it goes from a lower index to a higher one, and a *negative* link otherwise. The following is an extension of Definition 4.

Definition 6 Given a torus $T_{s\times t}$ and any integer h that divides both s and t, define h subnetworks $G_i^+ = (V_i^+, E_i^+), i = 0..h-1$, such that (refer to Definition 4):

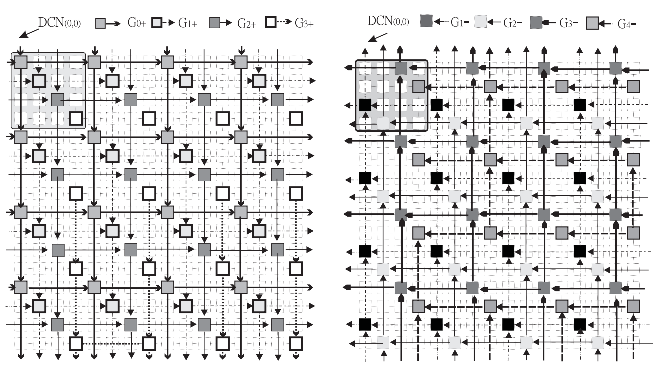
$$
\begin{aligned}
V_i^+ &= V_i \\
C_i^+ &= \{\text{all positive links in } C_i\},
\end{aligned}
$$

and h subnetworks $G_i^- = (V_i^-, E_i^-), i = 0..h-1$, such that:

$$
\begin{aligned}
V_i^- &= \{p_{x,y} | x = ah + i,\ y = bh + i + \delta, \text{ for all} \\
&\quad a = 0..\frac{s}{h} - 1 \text{ and } b = 0..\frac{t}{h} - 1\} \\
C_i^- &= \{\text{all negative links at rows } ah + i \\
&\quad \text{and at columns } bh + i + \delta\},
\end{aligned}
$$

where δ is any constant satisfying $1 \le \delta \le h - 1$.

Intuitively, G_i^+ is the same as G_i except that G_i^+ contains only positive links. Subnetwork G_i^- is obtained from G_i^+ by shifting each of the latter's nodes along the second dimension by δ positions and using only negative links. This is to resolve the node contention. For instance, Fig. 2 illustrates this definition in a 16×16 torus with $h = 4$ and $\delta = 2$ (for clarity, the eight subnetworks are drawn separately according to their link directions).



(a) (b)

Figure 2. Eight dilated-4 subnetworks, each as a directed 4×4 torus, in a 16×16 torus.

Lemma 3 *The $2h$ subnetworks G_i^+ and $G_i^-, i = 0..h-1$, defined in Definition 6 are are free from both node and link contention.*

Table 1. Comparison on levels of node and link contention incurred by different definitions of subnetworks in a torus.

type	subnet.	no of subnet.	links	node cont.	link cont.
I	$G_i, i=0..h-1$	h	undirected	no	no
II	$G_{i,j}, i,j=0..h-1$	h^2	undirected	no	h
III	$G_i^+, G_i^-, i=0..h-1$	$2h$	directed	no	no
IV	$G_{i,j}^*, i,j=0..h-1$	h^2	directed	no	$h/2$

The following is an extension of Definition 5.

Definition 7 Given a torus $T_{s\times t}$ and any integer h that divides both s and t, define h^2 subnetworks $G_{i,j}^* = (V_{i,j}^*, E_{i,j}^*), i,j=0..h-1$, such that (refer to Definition 5):

$$V_{i,j}^* = V_{i,j}$$
$$C_{i,j}^* = \begin{cases} \{\text{all positive links of } C_{i,j}\} & \text{if } i+j \text{ is even} \\ \{\text{all negative links of } C_{i,j}\} & \text{if } i+j \text{ is odd} \end{cases}$$

Lemma 4 *The h^2 subnetworks $G_{i,j}^*, i,j=0..h-1$, defined in Definition 7 are free from node contention, but have a link contention of $h/2$.*

In Table 1, we summarize the above definitions on the levels of node and link contention incurred by different subnetworks.

Definition 8 Given a torus $T_{s\times t}$ and any integer h that divides both s and t, define st/h^2 data collecting networks $DCN_{a,b} = (V_{a,b}, C_{a,b}), a=0..s/h-1, b=0..t/h-1$, such that

$$V_{a,b} = \{p_{x,y} | x = a\times h+i, y=b\times h+j \text{ for all } i,j=0..h-1\}$$
$$C_{a,b} = \{\text{all (undirected) links induced by } V_{a,b}\}.$$

For instance, when $h=4$, Fig. 1 illustrates the 16 DCNs (each as a 4×4 block) in a 16×16 torus. The same DCN definition will be used on all earlier four DDN definitions. Finally, it is not hard to see that these definitions satisfy properties **P1-P5**.

4. Multi-Node Multicast in a 2D Torus

Given a multi-node multicast instance $\{(s_i, M_i, D_i), i=1..m\}$, next we show in more details how to apply the multi-node multicast model in Section 2.1 using the DDNs and DCNs defined above. Throughout this section, let $DDN_0, DDN_1, \ldots, DDN_{\alpha-1}$ be h DDNs obtained from Definition 4, 5, 6, or 7, and $DCN_0, DCN_1, \ldots, DCN_{\beta-1}$ be k DCNs obtained from Definition 8.

4.1. Phase 1: Balancing Traffic among DDNs

In this phase, each multicast $(s_i, M_i, D_i), i=1..m$, should be distributed to one of the DDNs. There are two concerns to distribute the load. First, each DDN should receive about the same number of multicasts. Second, in each DDN, each node should be responsible for about the same number of multicasts. If the multicast pattern is given in advance, these are not hard to achieve.

A more distributed approach is to have each s_i randomly choose a DDN as its target subnetwork. This approach is more appropriate if multicasts arrive in an unpredictable or asynchronous manner or in a *stochastic* model, such as that assumed in [6]. In particular, if subnetworks of types II and IV are used (where each node must belong to some subnetwork), it is possible to skip this phase by letting s_i serve as its own representative node. Load balance is achieved automatically if multicasts arrive stochastically randomly.

4.2. Phase 2: Multicasting in DDNs

In this phase, each multicast (s_i, M_i, D_i) is translated into a (r_i, M_i, D_i') to be performed in a DDN. Since each DDN is still a torus under our definition (except that there is some link dilation), this is still a multicast on a conceptually smaller torus (due to the distance-insensitive characteristic of wormhole routing). Also, it should be commented that the way that D_i is translated to D_i' will incur a concentration effect and thus there is a high probability that $|D_i'| < |D_i|$. So, the multicast is on a smaller network with a smaller destination set. Statistically, we can say that $|D_i'| \approx |D_i|/\alpha$.

Overall, each DDN will still need to perform a multi-node multicast. With the dimension-ordered routing constraint, one possibility is to use the U-torus scheme [5] for each multicast.

4.3. Phase 3: Multicasting in DCNs

In this phase, each multicast (r_i, M_i, D_i') will incur a multicast $(d, M_i, D_i \cap DCN_c)$ on each $DCN_c, c=0..\beta-1$. Since DCN_c is a mesh and dimension-ordered routing is required, one possibility is to apply the *U-mesh* scheme [3].

4.4. Simulation and Performance Comparison

We have developed a simulator to study the performance issue. We mainly compared our scheme against the U-torus scheme [5] under various situations. The parameters used in our simulations are listed below.

- The torus size is 16×16.
- Startup time $T_s = 30$ or $300\mu sec$; transmission time per flit $T_c = 1\mu sec$.
- Dilation $h = 2$ or 4 (refer to Table 1).
- The problem instance is $\{(s_i, M_i, D_i), i=1..m\}$ with $|M_i| = 32 \sim 1024$ flits, and $m = |D_i| = 16 \sim 240$ nodes.
- A hot-spot factor of $p = 25\%, 50\%, 80\%,$ or 100% is used. Specifically, when generating D_i, we first choose $p|D_i|$ destination nodes which are common to all destination sets $D_i, i=1..m$. Then the rest $(1-p)|D_i|$

destination nodes are chosen randomly from the network. A larger p thus indicates higher contention on destination nodes.

Below, we show our simulation results from several prospects. Based on the subnetworks that are used, our schemes will be denoted as "HT[B]", where H reflects the value of h, T indicates the type of subnetworks (= I, II, III, or IV), and an optional B indicates whether we attempt to achieve load balance in Phase 1 or not. With a B, attempts will be made to evenly distribute multicasts to each DDN and each node in a DDN. If the network type is II or IV, a no-load-balance option is possible by skipping Phase 1 (refer to the discussion in Section 4.1).

A) Effects of Numbers of Sources and Destinations: Fig. 3(a) shows the multicast latency when $T_s = 300\mu sec$, $T_c = 1\mu sec$, $|M_i| = 32$ flits, and $|D_i| = 80$ at various numbers of sources. Undirected subnetworks (types I and II) have higher latency than that of the U-torus scheme, while directed subnetworks (types III and IV) have lower latency than that of the U-torus scheme. This is because the later will utilize more subnetworks, thus giving higher communication parallelism. Generally speaking, subnetworks without link contentions perform better than those with link contentions, so type I is better than type II, and type III is better than type IV. Overall, type III performs the best.

In Fig. 3(b), (c), and (d), we enlarge the number of destination nodes to observe the effect. The relative trend remains the same, but the advantage of using our schemes over the U-torus becomes more evident as there are more destinations. When there are 240 destinations (Fig. 3(d)), all our schemes deliver better performance than the U-torus scheme. This shows the importance of load balance especially at high traffic load. When using type III subnetworks, the performance gain over the U-torus scheme ranges between 2 to 6 times.

B) Effects of T_s/T_c Ratio: We repeated the same simulations in part A using a smaller T_s/T_c ratio of 30. Fig. 4 shows the results. As compared to Fig. 3, we see that the advantage of our schemes over the U-torus scheme becomes slightly larger. Recall that in Phase 1 we have to pay for the costs of re-distributing the multicasts to achieve better load balance. The extra costs in fact reduce as the ratio T_s/T_c decreases.

C) Effects of Message Lengths: Fig. 5 shows the multicast latency at various message sizes. The gain of our schemes over the U-torus scheme enlarges as message size increases. This again indicates the importance of load balance at heavier traffic load. The same observation applies too if we compare Fig. 5(a) and (b) (the latter has more sources and destinations).

D) Effects of h: The value of h has two effects. First, it reflects the number of subnetworks, and thus the level of communication parallelism. So a larger h generally delivers

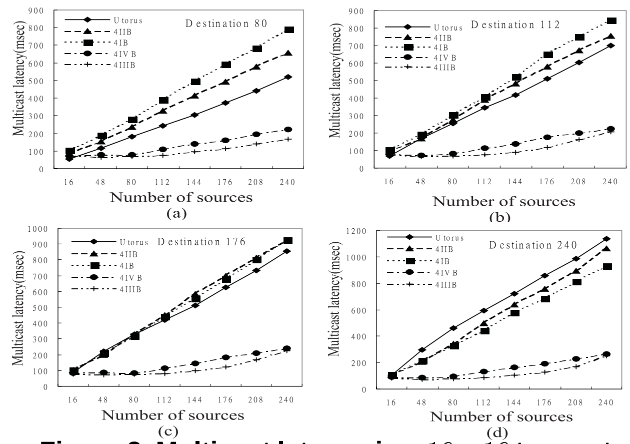


Figure 3. Multicast latency in a 16×16 torus at various numbers of sources when there are: (a) 80, (b) 112, (c) 176, and (d) 240 destination ($T_s = 300\mu sec$, $T_c = 1\mu sec$, and $|M_i| = 32$).

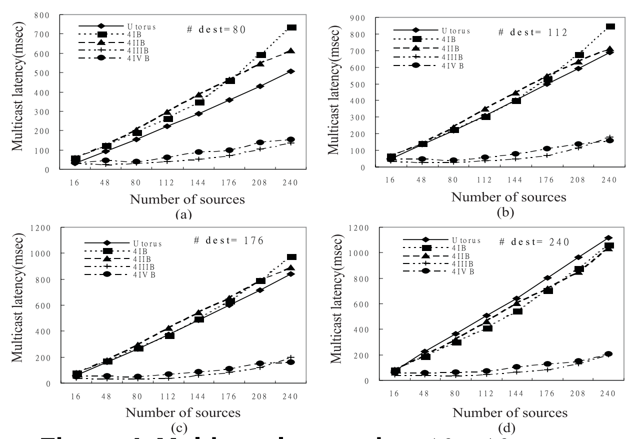


Figure 4. Multicast latency in a 16×16 torus at various numbers of sources when there are: (a) 80, (b) 112, (c) 176, and (d) 240 destination ($T_s = 30\mu sec$, $T_c = 1\mu sec$, and $|M_i| = 32$).

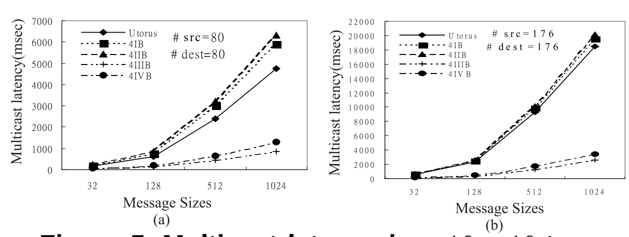


Figure 5. Multicast latency in a 16×16 torus at various message sizes: (a) 80 sources and destinations and (b) 176 sources and destinations ($T_s = 300\mu sec$ and $T_c = 1\mu sec$).

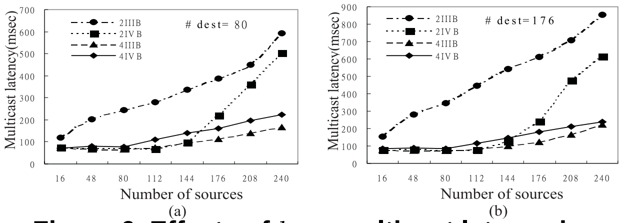


Figure 6. Effects of h on multicast latency in a 16×16 torus: (a) 80 destinations and (b) 176 destinations ($T_s = 300\mu sec$, $T_c = 1\mu sec$, and $|M_i| = 32$).

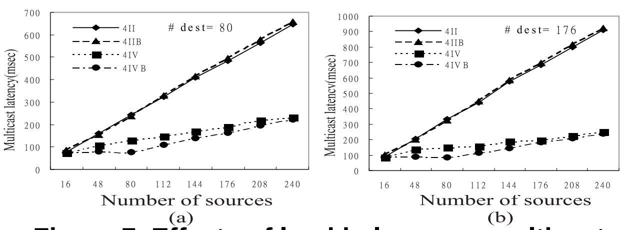


Figure 7. Effects of load balance on multicast latency in a 16×16 torus: (a) 80 destinations and (b) 176 destinations ($T_s = 300\mu sec$, $T_c = 1\mu sec$, and $|M_i| = 32$).

better performance. Second, for subnetwork types II and IV, it reflects the level of link contention, so a smaller h is better for these subnetworks. Fig. 6 compares subnetwork types III and IV when $h = 2$ and 4. The latency trend matches the above observations. One exception is type 2IVB, which delivers better performance than type 2IIIB. This is because type 2IVB offers 4 subnetworks with link contention $h/2 = 1$ (refer to Table 1).

E) Effects of Load Balance: As mentioned earlier, subnetwork types II and IV may be used with a no-load-balance option. Fig. 7 shows that the benefit of using load balance is more obvious when there are less sources. With more sources, the benefit is less obvious. In particular, for type II subnteworks, a no-load-balance option can even deliver slightly better performance when there are ≥ 112 sources. This is because when there are many sources spreading around the network, load balance can be achieved automatically.

F) Effects of Hot-Spot Factors: Fig. 8 shows how the hot-spot factor p affects multicast latency. A larger p will increase the latency. Among the three schemes that are compared, subnetwork type 4IIIB seems to be most insensitive to the hot-spot effect.

5. Conclusions

In this paper, we have developed a set of efficient schemes for multi-node multicast in a torus/mesh. One interesting feature of our approach is that the network is partitioned into several "dilated" subnetworks to achieve load balance and to increase communication parallelism. Contentions on links

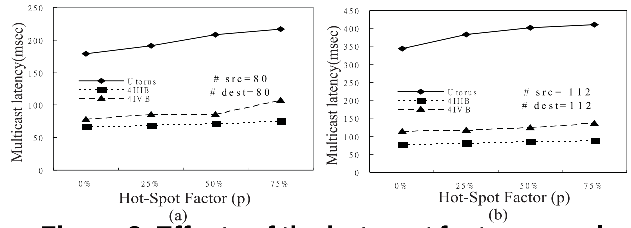


Figure 8. Effects of the hot-spot factor on multicast latency in a 16×16 torus: (a) 80 and (b) 112 sources and destinations ($T_s = 300\mu sec$, $T_c = 1\mu sec$, $|M_i| = 32$).

and nodes are thus separated evenly to the whole network. Extensive simulations have been conducted, which show significant improvement over existing U-torus, U-mesh, and SPU schemes. For space limit, simulation on meshes are omitted and can be found in [9].

Acknowledgements: This research is supported in part by the Ministry of Education, ROC, under grant 89-H-FA07-1-4 (Learning Technology) and the National Science Council, ROC, under grant NSC89-2213-E-008-025.

References

[1] W. J. Dally and C. L. Seitz. The torus routing chip. *J. of Distributed Computing*, 1(3):187–196, 1986.

[2] R. Kesavan and D. K. Panda. Multiple multicast with minimized node contention on wormhole k-ary n-cube networks. *IEEE Trans. on Paral. and Distrib. Sys.*, 10(4):371–393, April 1999. Also appeared in *Int'l Conf. on Parallel Processing*, 1996.

[3] P. K. Mckinley, H. Xu, A.-H. Esfahanian, and L. M. Ni. Unicast-based multicast communication in wormhole-routed networks. *IEEE Trans. on Paral. and Distrib. Sys.*, 5(12):1252–1265, Dec. 1994.

[4] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in directed network. *IEEE Computers*, 26(2):62–76, Feb. 1993.

[5] D. F. Robinson, P. K. Mckinley, and B. H. C. Cheng. Optimal multicast communication in wormhole-routed torus networks. *IEEE Trans. on Paral. and Distrib. Sys.*, 6(10):1029–1042, Oct. 1995.

[6] G. D. Stamoulis and J. N. Tsitsiklis. Efficient routing schemes for multiple broadcasts in hypercubes. *IEEE Trans. on Paral. and Distrib. Sys.*, 4(7):725–739, July 1993.

[7] Y.-C. Tseng, S.-Y. Wang, and C.-W. Ho. Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach. *IEEE Trans. on Paral. and Distrib. Sys.*, 10(1):44–61, Jan. 1999.

[8] S.-Y. Wang, Y.-C. Tseng, and C.-W. Ho. Efficient multicast in wormhole-routed 2d mesh/torus multicomputers: A network-partitioning approach. *Symp. on Frontiers of Massively Parallel Computation*, pages 42–49, 1996.

[9] S.-Y. Wang, Y.-C. Tseng, C.-S. Shiu, and J.-P. Sheu. Balancing traffic load for multi-node multicast in a wormhole 2D torus/mesh. Technical report, Dept. of Comp. Sci. and Info. Eng., Nat'l Central Univ., 1999.