# 行政院國家科學委員會專題研究計畫(第三年)結案報告

## 以 WWW 為介面的高效能平行計算編譯環境<br>子計畫三：多處理機網路上通訊指令的編譯及最佳化

A Web-Based Compiling Environment for High-Performance Parallel Computing
Compilation and Optimization of Communication Instructions for Multicomputer Networks

計畫編號：NSC 89-2218-E-009-095
執行期限：89 年 8 月 1 日至 90 年 7 月 31 日

主持人：曾煜棋 教授 交通大學資訊工程系

## 一、 中文摘要

在此報告中，我們針對一個以蟲洞繞徑的二維 torus/mesh 中，考慮多點群播的問題；這裡所謂的多點群播指的是：任意的起始點個數，每一個起始點都有任意數目的終點欲做傳播。為了要解決因繞徑所產生的競爭與壅塞的問題，我們提出了切割網路成子網路的方法，如此可平衡各網路鏈結的流量負載。我們討論了幾種切割網路的方法。同時，實驗模擬的結果也顯示，這樣的做法比現有在二維 torus/mesh 中的做法要好，可獲得大大的改善。

**關鍵詞：群集通訊、內部連結網路、群播、多處理機網路、網格結構、蟲洞繞徑。**

### Abstract

This report 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.

**Keywords:** collective communication, interconnection network, multicast, multicomputer networks, torus, wormhole routing.

## 二、 緣由與目的

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 report 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 the next section, 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 report 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].

## 三、研究方法

### 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. 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 $C' \subseteq C$.

For instance, Figure 1 shows four subnetworks, $G_i$, $i$=0..3, in a 16 x 16 torus. Our approach in this report 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.

### 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 $0 \le i <$ ~ and $0 \le j <$ .

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.. -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..= -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:** $DDN_0$, $DDN_1$,…, $DDN_{\alpha-1}$ are isomorphic.

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

### 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 using the DDNs and DCNs defined above.

### 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]. 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 $|D'_i| \approx |D_i|$ / ~.

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.. -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.

a The torus size is 16x16.

a Startup time $T_s$= 30 or 300 *ìsec*; transmission time per flit $T_c$ = 1 ìsec.

a Dilation $h$ = 2 or 4.

a The problem instance is $\{(s_i, M_i, D_i), i=1..m\}$ with $|M_i|$ =32 ~ 1024 flits, and $m= |D_i|$ = 16 ~ 240 nodes.

a 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 in Figures 2, 3, and 4 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, 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.

Figure 2 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. Figure 3 shows that the benefit of using load balance is more obvious when there are less sources. With more sources, the benefit is less obvious. Figure 4 shows how the hot-spot factor $p$ affects multicast latency. A larger $p$ will increase the latency.

## 四、結論與討論

In this report, 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 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.

## 五、參考文獻

[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 netwroks. *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.

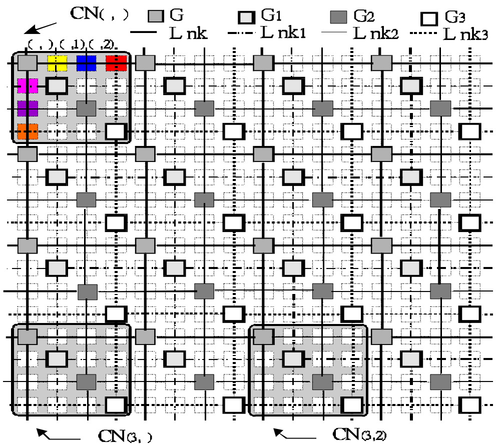

**Figure 1.** Four dilated-4 subnetworks, each as an undirected 4x4 torus, in a 16x16 torus.

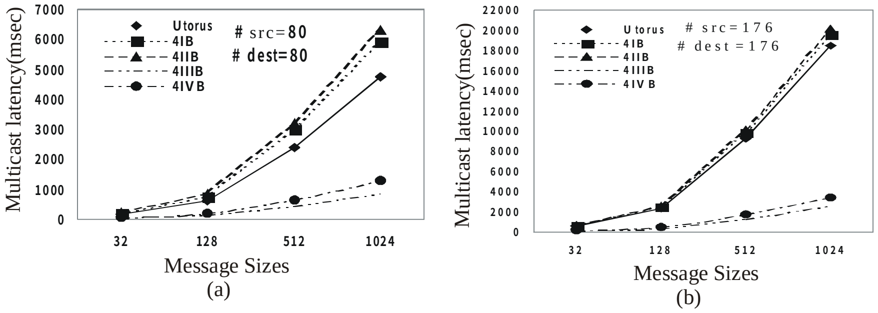

**Figure 2.** Multicast latency in a 16x16 torus at various message sizes: (a) 80 sources and destinations and (b) 176 sources and destinations (*$T_s$*= 300 *ì sec* and *$T_c$*= 1 *ì sec*).

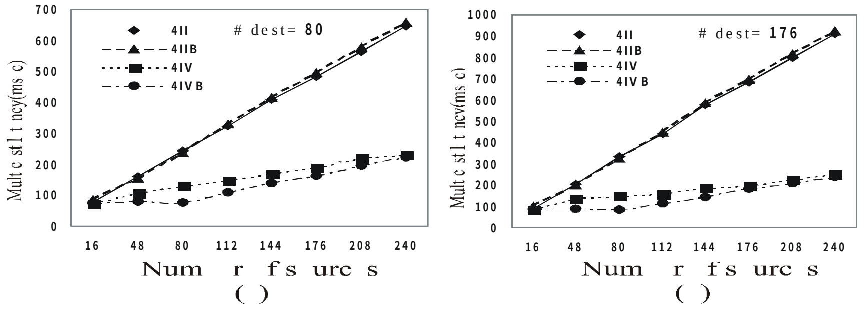

**Figure 3.** Effects of load balance on multicast latency in a 16x16 torus: (a) 80 destinations and (b) 176 destinations (*$T_s$*= 300 *ì sec*, *$T_c$*= 1 *ì sec*, and |*$M_i$*|=32).

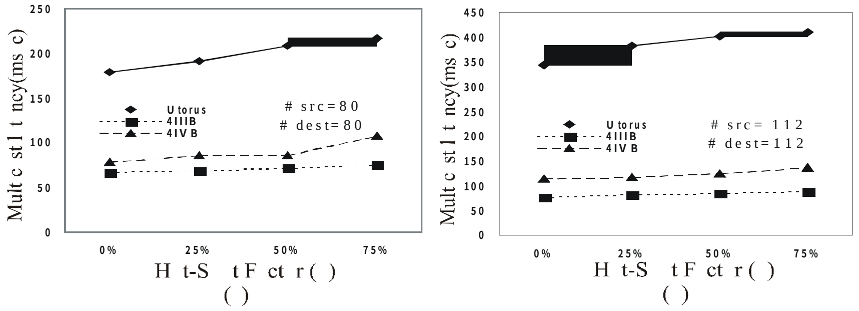

**Figure 4.** Effects of the hot-spot factor on multicast latency in a 16x16 torus: (a) 80 and (b) 112 sources and destinations (*$T_s$*= 300 *ì sec*, *$T_c$*= 1 *ì sec*, and |*$M_i$*|=32).