Efficient All-to-All Personalized Exchange in Multidimensional Torus Networks

Young-Joo Suh

Dept. of Computer Science and Engineering
Pohang University of Science and Technology
San 31, Hyoja-Dong
Pohang 790-784, Korea
yjsuh@postech.ac.kr

Kang G. Shin

Real-Time Computing Laboratory
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122
kgshin@eecs.umich.edu

Abstract

This paper presents new algorithms for all-to-all personalized exchange in multidimensional torus-connected multiprocessors. Unlike existing message-combining algorithms in which the number of nodes in each dimension should be power-of-two and square, the proposed algorithms accommodate non-power-of-two tori where the number of nodes in each dimension need not be power-of-two. In addition, destinations remain fixed over a larger number of steps in the proposed algorithms, thus making them amenable to optimizations. Finally, the data structures used are simple, hence making substantial saving of message-rearrangement time.

1. Introduction

Interprocessor communication may become a main bottleneck to scalable parallel implementations of computation-intensive applications. This has motivated the development of efficient and innovative algorithms for demanding interprocessor communication patterns such as *collective communication* [4,6]. Among several collective communication patterns, *all-to-all personalized exchange* or *complete exchange* is generally the most demanding communication pattern, where every node communicates a distinct message to every other node in the system. In an N-node system, each node P_i, $1 \le i \le N$, has N blocks of data $B[i, 1], B[i, 2], \ldots, B[i, N]$, one distinct block for each other node. After the all-to-all personalized exchange operation, each node P_i has N blocks of data, $B[1, i], B[2, i], \ldots, B[N, i]$, one from each other node. Many scientific parallel applications require this all-to-all personalized exchange communication.

Bokhari and Berryman [1], Sunder et al. [10], and Tseng et al. [13] proposed all-to-all personalized exchange algorithms using message combining in $2^d \times 2^d$ meshes or tori. These algorithms incur an execution time of $O(2^d)$ due to message startups and $O(2^{3d})$ due to message transmissions. In [8,9], Suh and Yalamanchili proposed algorithms using message combining in $2^d \times 2^d$ and $2^d \times 2^d \times 2^d$ tori or meshes with $O(d)$ time complexities due to message startups and $O(2^{3d})$ (in 2D) or $O(2^{4d})$ (in 3D) due to message transmissions. These algorithms differ from each other primarily in the way that pairwise exchange operations are scheduled. However, they have all been defined for meshes or tori where the number of processors in each dimension is an integer power-of-two and square.

In this paper, we present new algorithms for all-to-all personalized exchange for multidimensional tori. The algorithms utilize message combining to reduce the time associated with message startups. They are suitable for a wide range of torus topologies. The salient features of the proposed algorithms are (i) unlike existing message-combining algorithms in which the number of nodes in each dimension should be power-of-two and square, they accommodate non-power-of-two and non-square tori, (ii) they are simple in that destinations remain fixed over a larger number of steps, and are thus amenable to optimizations, e.g., caching of message buffers, locality optimizations, etc., (iii) they are the first message-combining algorithms for such 3D or higher dimensional tori, (iv) the data structures are simple and save substantial message-rearrangement time, and (v) they can be extended to higher-dimensional networks.

The following section presents the performance model and parameters used in this paper. We propose the algorithm for 2D tori in Section 3. The algorithm is extended to multidimensional networks in Section 4. Section 5 evaluates the performance of the proposed algorithms. Our results are summarized in Section 6.

2. Performance Model and Parameters

The target architecture is torus-connected, wormhole-switched [5] multiprocessors. The proposed algorithms apply equally well to networks using virtual cut-through or packet switching. Each packet is partitioned into a number of *flits*. We assume that each processor has N distinct m-

byte message blocks. We also assume that the channel width is one flit, the flit size is one byte, and each processor has one pair of injection/consumption buffers for the internal processor-router channel (i.e., one-port architecture). All links are full duplex channels. In this paper, a *step* is the basic unit of a contention-free communication and a *phase* is a sequence of steps.

A common metric used to evaluate the performance of inter-processor communication is *completion time* or *communication time*. In general, the completion time includes startup time, message-transmission time, propagation delay, and data-rearrangement time between communication steps/phases. Performance parameters include the startup time per message (t_s), message-transmission time per flit (t_c), per-hop propagation delay (t_l), and data-rearrangement time per byte (ρ). Thus, completion time (T) for one communication step can be expressed as $T= t_s+mt_c+ht_l$, if one message block is transmitted to the destination over h hops in a contention-free manner using wormhole switching. It does not include the data-rearrangement time between steps.

In this paper, the logical data structure in each node is a 2D (in Section 3) or *n*D array (in Section 4). We also assume that if physically non-contiguous blocks are transmitted from this array, a message-rearrangement step must be taken place prior to transmission.

3. Algorithm for 2D Tori

For an $R \times C$ torus, where R and C are multiples of four and $R \leq C$, each node is identified by a label $P(r, c)$, where $0 \leq r \leq R-1$ and $0 \leq c \leq C-1$. Each node is included in one of 16 node groups according to the following rule:

IF *r mod 4 = i* and *c mod 4 =j,* P(r,c) is included in group *ij.*

For example, in a 12×12 torus shown in Figure 1(b), nine nodes of identical marking are included in the same group. The nodes in a group form an $\frac{R}{4} \times \frac{C}{4}$ subtorus. Figure 1(a) illustrates the 3×3 subtorus formed by group 00 to which nine nodes, P(0,0), P(0,4), P(0,8), P(4,0), P(4,4), P(4,8), P(8,0), P(8,4), and P(8,8) belong. In addition, if we divide an $R \times C$ torus into 4×4 contiguous submeshes (SMs), each node in a SM is included in one of 16 distinct groups.

3.1 An Overview

The proposed 2D algorithm consists of four phases. In phases 1 and 2, messages are exchanged, performing all-to-all personalized exchange, among nodes in the same group. For an illustrative purpose, we consider all-to-all personalized exchange in a 12×12 torus. Figure 1(c) is a simplified representation of Figure 1(a), where only SMs and nodes in group 00 are shown. Each node has 144 blocks to scatter, and the blocks are divided into nine 4×4 block groups (BGs) considering nine SMs (SM00, SM01, SM02, SM10, SM11, SM12, SM20, SM21, and SM22) and 16 nodes in each SM. In Figure 1(d), each node in group 00 has 9 BGs to scatter with distinct markings, where each BG is destined for the SM which has the same marking as the BG in Figure 1(c). Thus, BGs of identical marking will be gathered in one node in the SM which has the same marking as the BGs, when all-to-all personalized exchange operation is completed successfully. Before starting transmission, the BGs are stored in a 2D array and they are arranged by considering the following steps (to be described in Section 3.3). In step 1 of phase 1, each node transmits the BGs in the second and third columns while receiving the same number of blocks along a row as illustrated in Figure 1(d). The data arrays after step 1 are illustrated in Figure 1(e). In step 2 of phase 1, each node transmits the BGs in the third column while receiving the same number of BGs (see Figure 1(e)). After step 2, BGs in each node are those destined for nodes in its SM and SMs in the same column as shown in Figure 1(f). Now, phase 2 starts and each node changes dimensions and transmits BGs along a column. In step 1 of phase 2 (step 2 of phase 2), each node transmits the BGs in the second and third rows (third row) while receiving the same number of BGs along a column as shown in Figure 1(f) (Figure 1 (g)). After step 2 of phase 2, all BGs gathered in each node have the same marking (see Figure 1(h)), which indicates that all-to-all personalized exchange among nodes in group 00 is achieved successfully.

In phases 1 and 2, nodes in the same group performs all-to-all personalized exchange operation among them, just as described above. However, since nodes in 16 distinct groups perform the operations in parallel, we should schedule links to avoid channel contention. If we consider a row (or column), each node in the row (or column) is included in one of four node groups. Since nodes in four groups cannot transmit message blocks along two directions in the row (or column) in parallel without channel contention, two node groups should be assigned to two directions in the other dimension for contention-free transmissions. Since there are four directions, positive row (+r), negative row (-r), positive column (+c), and negative column (-c), four node groups share distinct directions according to the result of *(r+c) mod 4* operation (see Figure 1(b)). In phase 2, each node changes dimensions then performs transmission along the new dimension.

After phase 2, each node in a SM has blocks originated from nodes in the same node group and destined for the 16 nodes in the same SM to which the node belongs. In the next two phases (phases 3 and 4), message transmissions are performed among nodes in distinct groups and in the same SM. Each SM can be divided into four 2×2 sub-

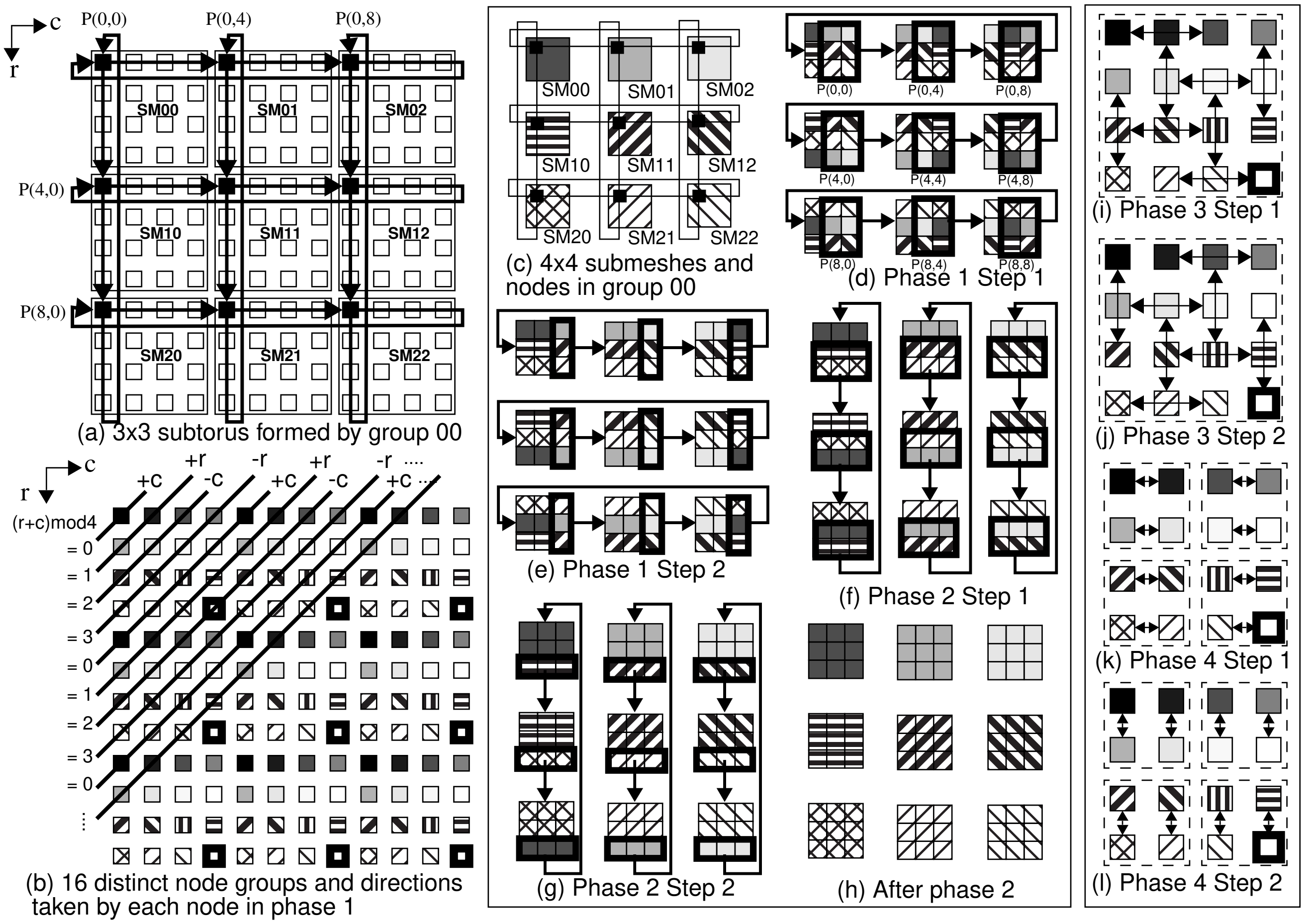

Figure 1. Node groups, a 3x3 subtorus formed by a node group, and all-to-all personalized exchange operation among nodes in a subtorus.

meshes. In each 2×2 submesh, there are four nodes in upper left, upper right, lower left, and lower right. In the two steps of phase 3, four nodes in the same position in 2×2 submeshes exchange blocks (see Figures 1(i) and (j), where only one SM is shown). In each step, each node transmits blocks destined for the destination node itself as well as blocks destined for the other three nodes in the 2×2 submesh to which the destination node belongs. After phase 3, each node in a 2×2 submesh has blocks originated from nodes in distinct four groups and destined for nodes in the same 2×2 submesh to which the node belongs. In the two steps of phase 4, four nodes in each 2×2 submesh exchange blocks to complete all-to-all personalized exchange (see Figures 1(k) and (l)). The following subsections describe the algorithm in detail.

3.2 Communication Pattern

In phase 1, the following operations are performed:

Phase 1:

IF *(r+c) mod 4 =0,* P(r,c) $\rightarrow$ P(r, (c+4) *mod C*). (1)

IF *(r+c) mod 4 =1,* P(r,c) $\rightarrow$ P((r+4) *mod R*, c). (2)

IF *(r+c) mod 4 =2,* P(r,c) $\rightarrow$ P(r, (c-4) *mod C*). (3)

IF *(r+c) mod 4 =3,* P(r,c) $\rightarrow$ P((r-4) *mod R*, c). (4)

Phase 1 requires $\frac{C}{4} - 1$ steps. Throughout these $\frac{C}{4} - 1$ steps of phase 1, each node transmits message blocks to a fixed destination node along the direction selected by the node. Since the size of a subtorus is $\frac{R}{4} \times \frac{C}{4}$, there are at most $\frac{C}{4}$ nodes in a row or column (note that $R \le C$). Consider blocks of a node (e.g., node A) to be scattered to all nodes. In step 1, node A transmits all of its blocks except those to be transmitted by itself in phases 2, 3, and 4, to the next node (e.g., node B) along the direction selected by the nodes. In step 2, node B extracts blocks to be transmitted by itself in phases 2, 3, and 4, then transmits the remaining blocks to the next node (e.g., node C) along the direction selected by the nodes. This procedure repeats and in the last step in phase 1, the last node (e.g., node L) along the direction receives only the blocks to be transmitted by the node in phases 2, 3, and 4. In the same manner, the other nodes also scatter their blocks to all nodes in the same node group and in the same column or row. If $R \ne C$, then each node that satisfies the above conditions (2) and (4) finish the operations in phase 1 in $\frac{R}{4} - 1$ steps, and idle or send empty messages during the remaining $\frac{C-R}{4}$ steps.

In phase 2, all nodes change dimensions then transmit message blocks along the new dimension. In phase 2, the following operations are performed:

Phase 2:
IF *(r+c) mod 4 =0,* P(r,c) → P((r+4) *mod R*, c). (5)
IF *(r+c) mod 4 =1,* P(r,c) → P(r, (c+4) *mod C*). (6)
IF *(r+c) mod 4 =2,* P(r,c) → P((r-4) *mod R*, c). (7)
IF *(r+c) mod 4 =3,* P(r,c) → P(r, (c-4) *mod C*). (8)

Phase 2 also requires $\frac{C}{4} - 1$ steps and the communication pattern is the same as that in phase 1. Each node in a row or column of phase 1 (e.g., each node A, B, C,..., L) transmits blocks along a column or row in its new dimension in parallel. In each step, each node extracts blocks for itself and blocks to be transmitted by itself in phases 3 and 4, then transmits the remaining blocks to the next destination node. Thus, after $\frac{C}{4} - 1$ steps of phase 2, each node has blocks originated from nodes in the same group, destined for itself and to be transmitted by the node in phases 3 and 4. As in phase 1, if $R \neq C$ then each node that satisfies the above conditions (2) and (4) finish the operations in phase 1 in $\frac{R}{4} - 1$ steps and idle or send empty messages during the remaining $\frac{C-R}{4}$ steps.

Now, the network can be divided into $\frac{RC}{16}$ 4×4 submeshes. All nodes in a 4×4 submesh are included in distinct node groups and have blocks originated from nodes in their respective groups. In the next two phases all-to-all personalized exchange operation is performed among nodes within each submesh. In phase 3, the following operations are performed:

Step 1 of Phase 3:
IF *(r+c) mod 4 =even* AND *c mod4=0 or 1*, P(r,c) → P(r, c+2).
IF *(r+c) mod 4 =even* AND *c mod4=2 or 3*, P(r,c) → P(r, c-2).
IF *(r+c) mod 4 =odd* AND *r mod4=0 or 1*, P(r,c) → P(r+2, c).
IF *(r+c) mod 4 =odd* AND *r mod4=2 or 3*, P(r,c) → P(r-2, c).
Step 2 of Phase 3:
IF *(r+c) mod 4 =even* AND *r mod4=0 or 1*, P(r,c) → P(r+2, c).
IF *(r+c) mod 4 =even* AND *r mod4=2 or 3*, P(r,c) → P(r-2, c).
IF *(r+c) mod 4 =odd* AND *c mod4=0 or 1*, P(r,c) → P(r, c+2).
IF *(r+c) mod 4 =odd* AND *c mod4=2 or 3*, P(r,c) → P(r, c-2).

In phase 4, the network is further divided into 2×2 submeshes and two steps are required as follows:

Step 1 of Phase 4:
IF *c mod 2=0,* P(r,c) → P(r, c+1).
IF *c mod 2=1,* P(r,c) → P(r, c-1).
Step 2 of Phase 4:
IF *r mod 2=0,* P(r,c) → P(r+1, c).
IF *r mod 2=1,* P(r,c) → P(r-1, c).

3.3 Data Array

In this subsection, the contents of transmitted blocks and the array structure in each communication step are described in detail.

Initially, $P(r, c)$ has RC distinct blocks to distribute to other nodes in two dimensional array B[u,v], where $0 \le u \le R-1$ and $0 \le v \le C-1$ if $(r+c) mod 4 = 0 \ or \ 2$ (i.e., nodes that transmit blocks along a row and a column in phases 1 and 2, respectively), or $0 \le u \le C-1$ and $0 \le v \le R-1$ if $(r+c) mod 4 = 1 \ or \ 3$ (i.e., nodes that transmit blocks along a column and a row in phases 1 and 2, respectively). We assume that the array is ordered in column major, and if blocks to be transmitted are not contiguous, then they should be rearranged before transmission. The initial data structure of a node is dependent upon the communication pattern in phases 1 and 2. A block destined for the node that is u hops away from the node along the direction that the node takes in phase 1 is located in B[u,0]. In B[u,v], a block destined for the node that is v hops away from the node in B[u,0] along the direction the node takes in phase 2 is located.

In step i, $1 \le i \le \frac{C}{4} - 1$, of phase 1, each node transmits blocks in columns $4i$ through $C-1$ to its destination node, while receiving the same number of blocks: In step 1, each node transmits all blocks except those to be transmitted by itself in phases 2, 3, and 4 (i.e., blocks in the first four columns). Among the blocks received in step 1, each node extracts the blocks to be transmitted by itself in following phases (i.e., blocks in the 5th through 8th columns), then transmits the remaining blocks to its destination node in step 2. This procedure repeats until the last step of phase 1.

In step j, $1 \le j \le \frac{C}{4} - 1$, of phase 2, each node transmits blocks in rows $4j$ through $C-1$ to its destination node in phase 2, while receiving the same number of blocks from its source node in phase 2: In step 1, each node transmits all blocks except those will be transmitted by itself in phases 3 and 4 (i.e., blocks in the first four rows). Among the blocks received in step 1, each node extracts the blocks to be transmitted by itself in phases 3 and 4 (i.e., blocks in the 5th through 8th rows) then transmits the remaining blocks to its destination node in step 2. This procedure repeats until the last step of phase 2.

After phase 2, each node in a 4×4 submesh has RC blocks originated from all nodes in the same group ($\frac{RC}{16}$ nodes) destined for nodes in the 4×4 submesh to which the node belongs. But blocks destined for each node in the 4×4 submesh are distributed. Thus, before phase 3, the blocks are rearranged: If we divide a 4×4 submesh into 2×2 submeshes, there are four 2×2 submeshes - one includes a node P (e.g., S0), another includes the partner node in step1 of phase 3 (e.g., S1), another includes the partner node in step 2 of phase 3 (e.g., S2), and the other submesh (e.g., S3). Blocks destined for S0, S1, S3, and S2 (e.g., B0, B1, B3, and B2, respectively) are arranged in that order in data array of node P. In step 1 of phase 3, node P sends blocks destined for S1 and S3 (i.e., B1 and B3) while receiving the same number of blocks, B0 and B2, from the partner node in step1 of phase 3. Now, blocks in node P's data array are B0, B0, B2, and B2, in that order. In the next step, node P sends B2's while receiving B0's.

After phase 3, each node in a 2×2 submesh has RC blocks originated from all nodes in four node groups destined for four nodes in the submesh to which the node belongs, and the blocks are distributed. Thus, before phase 4, the blocks are rearranged: blocks destined for the node itself (e.g., N0), partner node in step 1 of phase 4 (e.g., N1), partner node in step 2 of phase 4 (e.g., N2), and the other node (e.g., N3). Blocks destined for N0, N1, N3, and N2 are arranged in that order in data array of node N0, and the block transmissions in phase 4 are performed in the same manner as those in phase 3. Now, after phase 4, every node has RC blocks, one block from every node in the network.

3.4 Complexity Analysis

In this subsection, we analyze the time costs required by the proposed algorithm in terms of startup cost, message-transmission cost, data-rearrangement cost, and message propagation cost.

(a) Startup cost: For an $R \times C$ 2D torus, $\frac{C}{4} - 1$ steps per phase are required in phases 1 and 2, and two steps per phase are required in phases 3 and 4. Thus, a total of $\frac{C}{2} + 2$ steps are required.

(b) Message-transmission cost: In step p of phase 1, where $1 \le p \le \frac{C}{4} - 1$, $R(C - 4p)$ blocks (since $R \le C$) are transmitted. In step q of phase 2, where $1 \le q \le \frac{C}{4} - 1$, $R(C - 4q)$ blocks are transmitted. In phases 3 and 4, there are four steps and $\frac{RC}{2}$ blocks are transmitted in each step. Thus, the total number of transmitted blocks is $\frac{RC}{4}(C + 4)$.

(c) Data-rearrangement cost: At the end of each phase blocks are rearranged to prepare for the next phase. Since there are four phases, three data-rearrangement steps are required. Thus, the total data-rearrangement cost is $3(RC)m\rho$.

(d) Message propagation cost: In phases 1 and 2, there are $\frac{C}{2} - 2$ steps. In each step, the number of hops to the destination is four. In each of two steps in phases 3 and 4, the number of hops to the destination is two and one, respectively. Thus, the total number of hops is $2C - 2$ and the message propagation cost is expressed as $2(C - 1)t_l$.

4. Algorithm for *n*-Dimensional Tori

The algorithm for 2D tori can be extended to *n*-dimensional tori in a straightforward manner. Before describing the general *n*-dimensional algorithm, it may be helpful to first describe a 3D algorithm.

4.1 Algorithm for 3D Tori

For an $a_1 \times a_2 \times a_3$ 3D torus, where a_1, a_2, a_3 are a multiple of four and $a_1 \ge a_2 \ge a_3$, each node is labeled with $P(X, Y, Z)$, where $0 \le X \le a_1 - 1$, $0 \le Y \le a_2 - 1$, and $0 \le Z \le a_3 - 1$. Each node is included in one of 64 node groups according to the following rule:

IF *X mod4=i, Y mod4=j,* and *Z mod4=k,* P(X,Y,Z) is included in group *ijk*.

Communication Pattern:

The proposed algorithm requires five phases. In phase 1, the following operations are performed:

Phase 1:
IF *(X+Y) mod 4 =0* and *Z mod 4=0 or 2,* P(X,Y,Z) → P((X+4) *mod* a_1,Y,Z).
IF *(X+Y) mod 4 =1* and *Z mod 4=0 or 2,* P(X,Y,Z) → P(X,(Y+4) *mod* a_2,Z).
IF *(X+Y) mod 4 =2* and *Z mod 4=0 or 2,* P(X,Y,Z) → P((X-4) *mod* a_1,Y,Z).
IF *(X+Y) mod 4 =3* and *Z mod 4=0 or 2,* P(X,Y,Z) → P(X,(Y-4) *mod* a_2,Z).
IF *Z mod 4=1,* P(X,Y,Z) → P(X,Y,(Z+4) *mod* a_3).
IF *Z mod 4=3,* P(X,Y,Z) → P(X,Y,(Z-4) *mod* a_3).

The communication pattern of phase 1 in a 2D torus (*pattern A*) is performed in even numbered X-Y planes, while inter-plane communications (*pattern C*) are performed among nodes in odd numbered planes (see Figure 2(a)). There are $\frac{a_1}{4} - 1$ steps in phase 1.

In phase 2, the following operations are performed:

Phase 2:
IF *(X+Y) mod 4 =0,* P(X,Y,Z) → P(X,(Y+4) *mod* a_2,Z).
IF *(X+Y) mod 4 =1,* P(X,Y,Z) → P((X+4) *mod* a_1,Y,Z).
IF *(X+Y) mod 4 =2,* P(X,Y,Z) → P(X,(Y-4) *mod* a_2,Z).
IF *(X+Y) mod 4 =3,* P(X,Y,Z) → P((X-4) *mod* a_1,Y,Z).

In phase 2, every node in each X-Y plane follows the communication pattern of phase 2 in a 2D torus (*pattern B*) as shown in Figure 2(b). There are also $\frac{a_1}{4} - 1$ steps in phase 2.

In phase 3, the following operations are performed:

Phase 3:
IF *(X+Y) mod 4 =0* and *Z mod 4=1 or 3,* P(X,Y,Z) → P((X+4) *mod* a_1,Y,Z).
IF *(X+Y) mod 4 =1* and *Z mod 4=1 or 3,* P(X,Y,Z) → P(X,(Y+4) *mod* a_2,Z).
IF *(X+Y) mod 4 =2* and *Z mod 4=1 or 3,* P(X,Y,Z) → P((X-4) *mod* a_1,Y,Z).
IF *(X+Y) mod 4 =3* and *Z mod 4=1 or 3,* P(X,Y,Z) → P(X,(Y-4) *mod* a_2,Z).
IF *Z mod 4=0,* then P(X,Y,Z) → P(X,Y, (Z+4) *mod* a_3).
IF *Z mod 4=2,* then P(X,Y,Z) → P(X,Y, (Z-4) *mod* a_3).

In phase 3, nodes in even numbered planes follow *pattern C* while nodes in the other planes follow *pattern A* as shown in Figure 2(c). In phase 3, there too are $\frac{a_1}{4} - 1$ steps.

After phase 3, the network is divided into $\frac{a_1 a_2 a_3}{4^3}$ $4 \times 4 \times 4$ submeshes. Now, phase 4 initiates and has three steps. The following operations are performed in each step of phase 4 (see Figures 2(d)-(f), where only one $4 \times 4 \times 4$ submesh is shown):

Step 1 of Phase 4:
IF *(X+Y) mod 2=0, Y mod 4=0 or 1,* and *Z mod 2=0,* P(X,Y,Z) → P(X+2,Y,Z).
IF *(X+Y) mod 2=0, Y mod 4=2 or 3,* and *Z mod 2=0,* P(X,Y,Z) → P(X-2,Y,Z).
IF *(X+Y) mod 2=1, X mod 4=0 or 1,* and *Z mod 2=0,* P(X,Y,Z) → P(X,Y+2,Z).
IF *(X+Y) mod 2=1, X mod 4=2 or 3,* and *Z mod 2=0,* P(X,Y,Z) → P(X,Y-2,Z).
IF *Z mod 4=1,* P(X,Y,Z) → P(X,Y,Z+2).
IF *Z mod 4=3,* P(X,Y,Z) → P(X,Y,Z-2).
Step 2 of Phase 4:
IF *(X+Y) mod 2=0* and *X mod 4=0 or 1,* P(X,Y,Z) → P(X,Y+2,Z).
IF *(X+Y) mod 2=0* and *X mod 4=2 or 3,* P(X,Y,Z) → P(X,Y-2,Z).
IF *(X+Y) mod 2=1* and *Y mod 4=0 or 1,* P(X,Y,Z) → P(X+2,Y,Z).
IF *(X+Y) mod 2=1* and *Y mod 4=2 or 3,* P(X,Y,Z) → P(X-2,Y,Z).
Step 3 of Phase 4:
IF *(X+Y) mod 2=0, Y mod 4=0 or 1,* and *Z mod 2=1,* P(X,Y,Z) → P(X+2,Y,Z).
IF *(X+Y) mod 2=0, Y mod 4=2 or 3,* and *Z mod 2=1,* P(X,Y,Z) → P(X-2,Y,Z).
IF *(X+Y) mod 2=1, X mod 4=0 or 1,* and *Z mod 2=1,* P(X,Y,Z) → P(X,Y+2,Z).
IF *(X+Y) mod 2=1, X mod 4=2 or 3,* and *Z mod 2=1,* P(X,Y,Z) → P(X,Y-2,Z).
IF *Z mod 4=0,* P(X,Y,Z) → P(X,Y,Z+2).
IF *Z mod 4=2,* P(X,Y,Z) → P(X,Y,Z-2).

After phase 4, the network is further divided into $\frac{a_1 a_2 a_3}{8}$ $2 \times 2 \times 2$ submeshes. Now, phase 5 is initiated and there are three steps. In each step, every node exchanges messages along X-, Y-, and Z-dimension, respectively (see Figures 2(g)-(i), where only one $2 \times 2 \times 2$ submesh is shown). That is, the following operations are performed in each step of phase 5:

Step 1 of Phase 5:
IF *X mod 2=0,* P(X,Y,Z) → P(X+1,Y,Z).
IF *X mod 2=1,* P(X,Y,Z) → P(X-1,Y,Z).
Step 2 of Phase 5:
IF *Y mod 2=0,* P(X,Y,Z) → P(X,Y+1,Z).
IF *Y mod 2=1,* P(X,Y,Z) → P(X,Y-1,Z).
Step 3 of Phase 5:
IF *Z mod 2=0,* P(X,Y,Z) → P(X,Y,Z+1).
IF *Z mod 2=1,* P(X,Y,Z) → P(X,Y,Z-1).

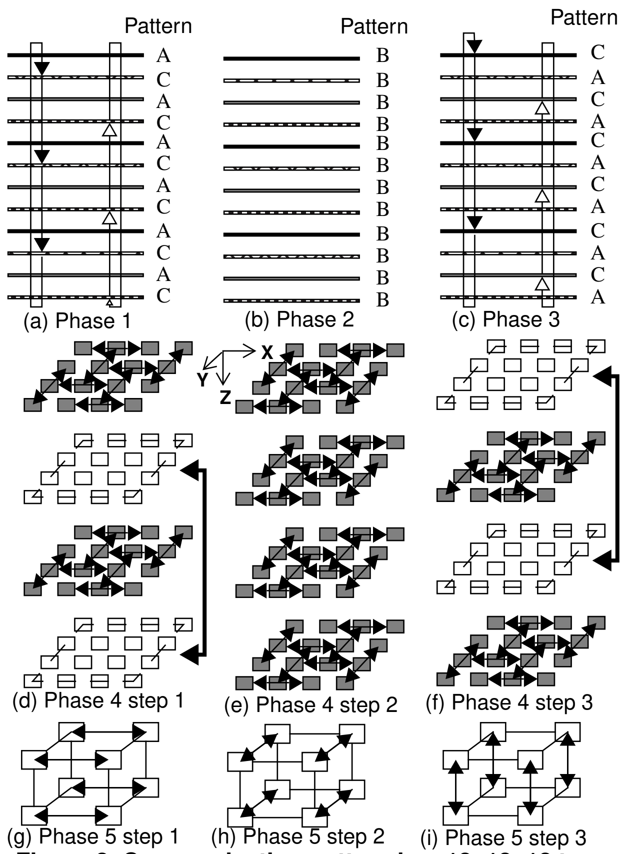

Figure 2. Communication pattern in a 12x12x12 torus.

Data Array:

Now, consider the data array of each node. Initially each node has $a_1 a_2 a_3$ distinct blocks in a three dimensional array B[u,v,w], where $0 \le u \le a_1 - 1$, $0 \le v \le a_2 - 1$, and $0 \le w \le a_3 - 1$. Since the data array structure in 3D tori is very similar to that in 2D tori and can be extended in a straightforward manner, we just examine the communication requirements in node P(0,0,0). In step 1 of phase 1, P(0,0,0) sends to P(4,0,0) blocks B[4..a_1-1,*,*], while receiving the same number of blocks from node P(a_1-4,0,0). The notation B[4..a_1-1,*,*] identifies all blocks from B[4,0,0] to B[a_1-1,a_2-1,a_3-1]. In the next step, P(0,0,0) transmits blocks B[8..a_1-1,*,*] to P(4,0,0). In general, in step s_1 of phase 1, $1 \le s_1 \le \frac{a_1}{4} - 1$, P(0,0,0) transmits blocks B[4s_1..a_1-1,*,*]. In step s_2 of phase 2, $1 \le s_2 \le \frac{a_2}{4} - 1$, P(0,0,0) transmits blocks B[*, 4s_2..a_2-1,*] to P(0,4,0). In step s_3 of phase 3, $1 \le s_3 \le \frac{a_3}{4} - 1$, P(0,0,0) transmits blocks B[*,*, 4s_3..a_3-1] to P(0,0,4). The blocks transmitted by node P(0,0,0) in each step of phases 1, 2, and 3 in a $12 \times 12 \times 12$ torus are shown in Figure 3. After

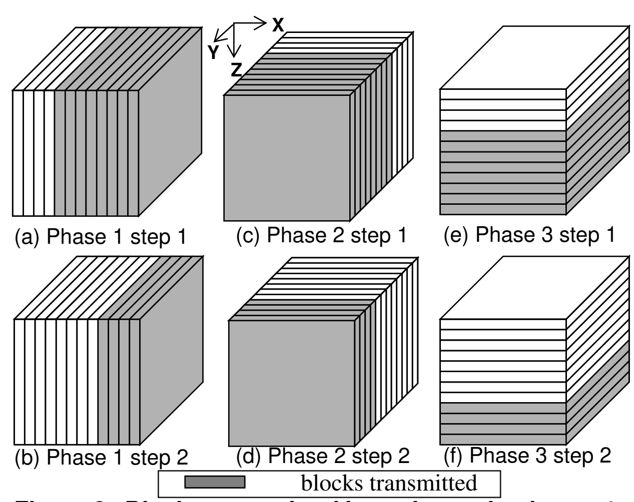

Figure 3. Blocks transmitted in each step in phases 1, 2, and 3 for a 12x12x12 torus.

phase 3, blocks originated from nodes in the same group destined for nodes in the $4 \times 4 \times 4$ submesh in which P(0,0,0) is included are gathered in P(0,0,0). Thus, in six steps in phases 4 and 5, the blocks destined for the other nodes in the $4 \times 4 \times 4$ submesh are transmitted.

4.2 Extension to *n*-Dimensional Tori

Now, we describe a general *n*-dimensional algorithm. Since the extension for *n*-dimensional tori can be made similarly to the 2D-3D extension, we describe the *n*-dimensional algorithm briefly in this subsection.

For an $a_1 \times \ldots \times a_n$ *n*-dimensional tori, where $a_1, \ldots, a_n$ are a multiple of four and $a_1 \geq \ldots \geq a_n$, there are *n*+2 phases. In the first *n* phases, messages are transmitted among nodes in the same group which form an $\frac{a_1}{4} \times \ldots \times \frac{a_n}{4}$ subtorus. To avoid channel contention, the dimensions in which messages are transmitted are distributed in each phase. In general, for *n*-dimensional tori, nodes in the even numbered unit along the dimension *n* follow the communication patterns of (*n*-1)-dimensional networks during the first *n*-1 phases and then perform the communications along the last dimension (i.e., dimension *n*) in phase *n*, while the other nodes perform the communications along the dimension *n* in phase 1 and then follow the communications of (*n*-1)-dimensional networks during the remaining *n*-1 phases. In phases *n*+1 and *n*+2, message exchanges are performed among nodes in $4 \times \ldots \times 4$ and $2 \times \ldots \times 2$ *n*-dimensional submeshes, respectively.

4.3 Complexity Analysis

In this subsection, we analyze the time cost required by the proposed algorithm.

(a) Startup cost: For an $a_1 \times \ldots \times a_n$ *n*-dimensional torus, $a_1 \geq \ldots \geq a_n$, there are *n*+2 phases. In the first *n* phases, $\frac{a_1}{4} - 1$ steps per phase are required. In phases *n*+1 and *n*+2, *n* steps are required in each phase. Thus, a total of $n\left(\frac{a_1}{4} + 1\right)$ steps is required.

(b) Message-transmission cost: In step s, $1 \leq s \leq \frac{a_1}{4} - 1$, in each of the first *n* phases, $(a_1 - 4 \cdot s)(a_2 \ldots a_n)$ blocks are transmitted (since $a_1 \geq a_2 \geq \ldots \geq a_n$). In each step of phases *n*+1 and *n*+2, $\frac{1}{2}(a_1 a_2 \ldots a_n)$ blocks are transmitted. Thus, the total number of transmitted blocks is $\frac{n}{8}(a_1 + 4)(a_1 \ldots a_n)$.

(c) Data-rearrangement cost: At the end of each phase blocks are rearranged to prepare for the next phase. Since there are *n*+2 phases, *n*+1 data-rearrangement steps are required. Thus, the total data-rearrangement cost is $(n+1)(a_1 a_2 \ldots a_n) m\rho$.

(d) Message propagation cost: In the first *n* phases, there are $\frac{a_1}{4} - 1$ steps per phase. In each step, the number of hops to the destination is four. In phases *n*+1 and *n*+2, *n* steps are required in each phase and the number of hops to the destination is two and one, respectively. Thus, message propagation cost is expressed as $n(a_1 - 1)t_l$.

5. Performance Evaluation

Thus far, we analyzed the time cost required by the proposed algorithm in terms of startup cost, message-transmission cost, data-rearrangement cost, and message propagation cost. In this section, the performance of the proposed algorithms are evaluated and compared with that of existing algorithms.

Network	$R \times C$ torus	$a_1 \times \ldots \times a_n$ torus
Startup Cost	$\left(\frac{C}{2} + 2\right)t_s$	$n\left(\frac{a_1}{4} + 1\right)t_s$
Message Trans. Cost	$\frac{RC}{4}(C+4)mt_c$	$\frac{n}{8}(a_1+4)(a_1 a_2 \ldots a_n)mt_c$
Data-Rearrangement Cost	$3(RC)m\rho$	$(n+1)(a_1 a_2 \ldots a_n)m\rho$
Propagation Cost	$2(C-1)t_l$	$n(a_1 - 1)t_l$

Table 1: Performance summary of the proposed algorithms.

The time complexities of the proposed algorithms are summarized in Table 1. We are not aware of any existing message-combining algorithms for *n*-dimensional tori, where the number of nodes in each dimension is not power-of-two. For 2D tori, Tseng et al. [13] proposed an algorithm using message combining. In the algorithm, the tori are assumed to be power-of-two square networks. If we apply the proposed 2D algorithm to power-of-two square tori, the startup time and message-transmission time are equivalent to those in [13] (see Table 2). But, the proposed algorithm is advantageous with respect to data-rearrangement time and message propagation time. In the proposed 2D algorithm, data rearrangement is required between phases to prepare for the next phase. In a $2^d \times 2^d$ torus, there are four phases in the proposed algorithm, thus only three rearrangement steps are required, regardless of the network size. However, in the algorithm [13], data rearrangement is required between steps rather than phases (in our physical model of data array: if non-contiguous blocks are transmitted, the blocks should be rearranged or copied). Thus, the algorithm [13] requires $2^{d-1} + 1$ data-rearrangement steps. With respect to the total propagation delay, the proposed algorithm requires four hops (in phases 1 and 2), two hops (in phase 3), and one hop (in phase 4) per step, regardless of the network size. Thus, this algorithm which exhibits time complexity of $O(2^d)$ compares favorably to the algorithm [13] which exhibits time com-

Network	[13]	[9]	Proposed
Startup Cost	$(2^{d-1}+2)t_s$	$(3d-3)t_s$	$(2^{d-1}+2)t_s$
Message-Transmission Cost	$(2^{3d-2}+2^{2d})mt_c$	$\{9\cdot 2^{3d-4}+(d^2-5d+3)2^{2d-1}\}mt_c$	$(2^{3d-2}+2^{2d})mt_c$
Data-Rearrangement Cost	$(2^{d-1}+1)2^{2d}m\rho$	$\{9\cdot 2^{3d-4}+(d^2-5d+3)2^{2d-1}\}m\rho$	$(3)2^{2d}m\rho$
Propagation Cost	$\frac{1}{3}(2^{2d-1}+10)t_l$	$(13\cdot 2^{d-2}-3d-3)t_l$	$(2^{d+1}-2)t_l$

Table 2: Comparison of completion time in two algorithms for a 2^dx2^d torus.

plexity of $O(2^{2d})$ due to propagation time. Thus, overall the proposed algorithm exhibits better performance than the existing algorithm [13] for power-of-two and square tori, even though the proposed algorithm is targeted at the networks whose size of each dimension need not be power-of-two and square. In [9], Suh and Yalamanchili proposed an algorithm using message combining for power-of-two tori. For a $2^d \times 2^d$ torus, message startup cost is $O(d)$ for the algorithm [9] while it is $O(2^d)$ for the proposed algorithm. The message-transmission cost of the proposed algorithm is $O(2^{3d})$ as the algorithm [9] but lower than that of the algorithm [9]. The time complexity due to data rearrangement for the algorithm [9] is $O(2^{3d})$, while that of the proposed algorithm is $O(2^{2d})$. With respect to the total propagation time, the proposed algorithm exhibits time complexity of $O(2^d)$ as the algorithm [9], but a little lower than that of the algorithm [9]. Thus, the proposed algorithm is advantageous over the algorithm [9] in all parameters except the startup cost.

6. Conclusions

This paper has proposed new algorithms for all-to-all personalized exchange for multidimensional torus-connected networks. Although the algorithms targeted at wormhole-switched networks, they can be efficiently used in virtual cut-through or circuit-switched networks. The proposed algorithms utilize message combining to reduce the time complexity of message startups. Unlike existing message-combining algorithms, the proposed algorithms accommodate non-power-of-two networks of arbitrary dimensions. In addition, destinations remain fixed over a larger number of steps in the proposed algorithms, thus making them amenable to optimizations. Finally, the data structures used are simple and hence make substantial saving of message-rearrangement time.

Although we assumed that the number of nodes in each dimension is multiple of four, the proposed algorithms can be used in tori with an arbitrary number of nodes in each dimension. If the number of nodes in each dimension is not a multiple of four, the proposed algorithms can be used by adding virtual nodes, then having every node perform communication steps as proposed in this paper.

When applied to power-of-two square tori, the proposed algorithms exhibit better performance than the algorithm [13], but the algorithm [9] shows much lower startup costs than that of the proposed algorithm although the proposed algorithms are favorable in other parameters. Thus, it may be interesting to study the comparative performance of the proposed algorithms and the algorithm [9].

References

[1] S. H. Bokhari and H. Berryman, “Complete Exchange on a Circuit Switched Mesh,” *Scalable High Performance Computing Conference*, pp. 300-306, 1992.
[2] S. H. Bokhari, “Multiphase Complete Exchange on Paragon, SP2, and CS-2,” *IEEE Parallel & Distributed Technology*, pp. 45-59, Fall 1996.
[3] W. J. Dally, “Performance Analysis of *k*-ary *n*-cube Interconnection Networks,” *IEEE Trans. on Computer*, vol. 39, no. 6, pp. 775-785, June 1992.
[4] P. K. McKinley and Y.-J. Tsai and D. Robinson, “Collective Communication in Wormhole-routed Massively Parallel Computers,” IEEE Computer, pp. 39-50, December 1995.
[5] L. M. Ni and P. K. McKinley, “A Survey of Wormhole Routing Techniques in Direct Networks,” *IEEE Computer*, vol. 26, pp. 62-76, February 1993.
[6] D. K. Panda, “Issues in Designing Efficient and Practical Algorithms for Collective Communication on Wormhole-Routed Systems,” Technical Report TR-25, Dept. of Computer and Information Science, Ohio State University.
[7] D. S. Scott, “Efficient All-to-All Communication Patterns in Hypercube and Mesh Topologies,” *Proceedings of 6th Conference. Distributed Memory Concurrent Computers*, pp. 398-403, 1991.
[8] Y. J. Suh and S. Yalamanchili, “Algorithms for All-to-All Personalized Exchange in 2D and 3D Tori,” Proceedings of the 10th International Parallel Processing Symposium, pp. 808-814, April 1996.
[9] Y. J. Suh and S. Yalamanchili, “All-to-All Communication with Minimum Start-Up Costs in 2D/3D Tori and Meshes,” IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 5, pp. 442-458, May 1998.
[10] N. S. Sundar, D. N. Jayasimha, D. K. Panda, and P. Sadayappan, “Complete Exchange in 2D Meshes,” *Scalable High Performance Computing Conference*, pp. 406-413, 1994.
[11] R. Thakur and A. Choudhary, “All-to-All Communication on Meshes with Wormhole Routing,” *Proceedings of 8th International Parallel Processing Symposium*, pp. 561-565, 1994.
[12] Y.-C. Tseng and S. Gupta, “All-to-All Personalized Communication in a Wormhole-Routed Torus,” *Proc. of International Conference on Parallel Processing*, vol. 1, pp. 76-79, 1995.
[13] Y.-C. Tseng, S. Gupta, and D. Panda, “An Efficient Scheme for Complete Exchange in 2D Tori,” *Proceedings of International Parallel Processing Symposium*, pp. 532-536, 1995.
[14] Message Passing Interface Forum, “MPI: A Message-Passing Interface Standard,” Technical Report CS-93-214, University of Tennessee, April 1994.
[15] Cray T3D, *System Architecture Overview*, 1994.