The Journal of Supercomputing, 20, 217–241, 2001

Circuit-Switched Broadcasting in Multi-Port Multi-Dimensional Torus Networks*

SAN-YUAN WANG sywang@isu.edu.tw

Department of Information Engineering, I-Shou University, Kaohsiung, 84008, Taiwan

YU-CHEE TSENG yctseng@csie.nctu.edu.tw

Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu, 30050, Taiwan

SZE-YAO NI AND JANG-PING SHEU {nee, sheujp}@csie.ncu.edu.tw

Department of Computer Science and Information Engineering, National Central University, Chung-Li, 32054, Taiwan

Abstract. The *one-to-all broadcast* is the most primary collective communication pattern in a multicomputer network. This paper studies this problem in a *circuit-switched* torus with *α-port* capability, where a node can simultaneously send and receive α messages at one time. This is a generalization of the one-port and all-port models. We show how to efficiently perform broadcast in tori of any dimension, any size, square or nonsquare, using near optimal numbers of steps. The main techniques used are: (i) a "span-by-dimension" approach, which makes our solution scalable to torus dimensions, and (ii) a "squeeze-then-expand" approach, which makes possible solving the difficult cases where tori are non-square. Existing results, as compared to ours, can only solve very restricted sizes or dimensions of tori, or use more numbers of steps.

Keywords: broadcast, circuit switching, collective communication, interconnection network, parallel processing, torus

1. Introduction

Efficient inter-processor communication is critical for a multicomputer network to deliver high performance. One primary communication is the *one-to-all broadcast*, where a source node needs to send a message to all other nodes in the network. Broadcast has applications in algebraic problems, parallel graph and matrix algorithms, cache coherence in distributed-shared-memory systems, and data re-distribution in HPF. In addition to one-to-all broadcast, many *collective communication* patterns, such as *all-to-all broadcast, complete exchange, scatter, gather*, and *reduction*, have received intensive attention recently [2, 5, 6, 15, 16].

In *circuit switching*, a routing header, containing the destination address and some routing control information, is injected into the network to build a physical path

*This work is supported by the National Science Council of the Republic of China under Grant # NSC88-2213-E-008-027. A preliminary version of this paper has appeared in the *EURO-PAR99* conference.

from the source to the destination. This can be done by connecting the input and output ports of intermediate nodes and preserving the links as the header progresses toward the destination. When the header reaches the destination, an acknowledgement is sent back to the source. The message contents are then sent in a pipeline fashion on the reserved path. The path can be released by the destination or by the last few bits of the message. For instance, the latter approach is adopted by Intel iPSC/2 [8] to release the path.

In this paper, we study the scheduling of one-to-all broadcast in a circuit-switched kD torus. The network is assumed to use the *α-port* communication model, in which a node can send up to α messages and simultaneously receive up to α messages at a time, where $1 \leq \alpha \leq 2k$. This is a generalization of the *one-port* model ($\alpha = 1$) and the *all-port* model ($\alpha = 2k$). Following the formulation in many works [3, 9, 10, 11, 14, 17], this is achieved by constructing a sequence of *steps*, where a step consists of a set of congestion-free communication paths each indicating a message delivery. The goal is to minimize the total number of steps used.

One-to-all broadcast has been studied for meshes and torus based on different port models and switching models [4, 9, 10, 11, 14, 15, 18]. Based on all-port circuit switching, the scheme in [11] uses optimal numbers of steps for any 2D torus of size $5^p \times 5^p$ or $(2 \times 5^p) \times (2 \times 5^p)$, where p is any integer. Likewise, the schemes of [9, 10] remain optimal, but can be applied to any square kD torus with $(2k+1)^p$ nodes on each side. Generalization to square 2D/3D tori/meshes supporting multi-port capability is shown in [4]. Drawbacks of these works include limitations on torus dimension, size, or that the networks must be square. Another direction is to assume the *wormhole-routing* model with *dimension-ordered* routing. For instance, based on the all-port model, [14] solves the cases of 2D and 3D torus of sizes $2^p \times 2^p$ and $2^p \times 2^p \times z$, respectively (z is an integer); [15] solves the case of 2D tori of any size, square or non-square; and [18] further extends [15] to square kD tori. With circuit switching, routing has less restrictions; routing does not have to conform to dimension ordering. In Table 1, we compare these solutions against the yet-to-be-presented results in this paper. As can be seen, our results can improve over existing results in either the numbers of communication steps required or the network dimension/size restrictions.

The difficulty of this problem lies in how we disseminate the broadcast message in a congestion-free manner. In order for a solution to be optimal, it typically relies on recursively partitioning the torus into $\alpha + 1$ smaller subnetworks. For instance, when the number of ports $\alpha = 1$ (resp., $\alpha = 3$), a recursively doubling (quadrupling) approach [13] can easily achieve optimality. Unfortunately, there is no known systematic solution based on this approach, especially when α is even (say, $\alpha = 2$ or 4 in a 3D torus). One readily sees additional difficulties when tori are non-square and of more dimensions.

We first show how to efficiently perform broadcast in 2D and 3D tori of any size, square or non-square, using near optimal numbers of steps. We then generalize the result to higher-dimensional tori. The main techniques used here are: (i) a "span-by-dimension" approach, and (ii) a "squeeze-then-expand" approach similar to [15] (here we extend the technique in [15] from 2D to kD). For instance, given a non-square 3D torus, we first "squeeze" the torus into a square one using (ii). Then, we

Table 1. Comparison of broadcast algorithms on the solvable network dimensions/sizes and required numbers of steps, assuming a network size of $n_1 \times n_2 \times \cdots \times n_k$

Algorithm	Ours	Lee-Lee [4]	Park-Choi [9]	Peters-Syska [11]	Wang-Tseng [14]	Tsai-McKinley [14]
Switching Technique	CS	CS	CS	CS	WH & DO	WH & DO
Port Model	α-port	α-port	all-port	all-port	all-port	all-port
2D						
Size	$n_1 \times n_2$	$n_1 = n_2$	$n_1 = n_2 = 5^p$	$n_1 = n_2 = 5^p$ or 2×5^p	$n_1 = n_2$	$n_1 = n_2 = 2^p$
Steps	$\begin{cases} LB(2)_\alpha + 1 & \text{if } n_1 = n_2, \\ LB(2)_\alpha + 4 & \text{otherwise} \end{cases}$	$LB(2)_\alpha + 2$	$LB(2)_4$	$LB(2)_4$	$LB(2)_\alpha + 2$	$\log_4 5 \times LB(2)_4$
3D						
Size	$n_1 \times n_2 \times n_3$	$n_1 = n_2 = n_3$	$n_1 = n_2 = n_3 = 7^p$	N/A	$n_1 = n_2 = n_3$	$n_1 = n_2 = 2^p$
Steps	$\begin{cases} LB(3)_\alpha + 2 & \text{if } n_1 = n_2 = n_3, \\ LB(3)_\alpha + 6 & \text{otherwise} \end{cases}$	$LB(3)_\alpha + 2$	$LB(3)_6$	N/A	$LB(3)_\alpha + 4$	$\log_4 7 \times LB(3)_6$
kD						
Size	$\prod_{i=1}^{k} n$	N/A	$\prod_{i=1}^{k}(2k+1)^p$	N/A	$\prod_{i=1}^{k} n$	N/A
Steps	$\begin{cases} LB(k)_\alpha & \text{if } n = (\alpha+1)^p, \\ LB(k)_\alpha + k - 1 & \text{otherwise} \end{cases}$	N/A	$LB(k)_{2k}$	N/A	$LB(k)_{2k} + 2(k-1)$	N/A

k = dimension; CS = circuit switching; WH = wormhole routing; DO = dimension-ordered routing; $LB(k)_\alpha$ = the lower bound for α-port kD tori in Lemma 1.

use (i) to "span" the nodes receiving the broadcast message from the source node to a line of nodes, to a plane of nodes, and then to a cube of nodes. Finally, the torus is "expanded" back to the original (non-square) torus. Technique (i) makes our results scalable to torus dimensions, while technique (ii) makes possible solving the difficult cases where tori are non-square.

The rest of this paper is organized as follows. Preliminaries are given in Section 2. Our solutions for 2D and 3D tori are in Section 3 and Section 4, respectively. Section 5 briefly summaries how to extend our results to higher-dimensional tori. Finally, conclusions are drawn in Section 6.

2. Preliminaries

A kD torus of size $n_1 \times n_2 \times \cdots \times n_k$ is an undirected graph denoted as $T_{n_1 \times n_2 \times \cdots \times n_k}$. Each node is denoted as $p_{x_1,x_2,\ldots,x_k}$, $0 \le x_i < n_i$, $1 \le i \le k$. Each node is of degree $2k$. Node $p_{x_1,x_2,\ldots,x_k}$ has an edge connecting to $p_{(x_1 \pm 1) \bmod n_1,x_2,\ldots,x_k}$ along dimension one, an edge to $p_{x_1,(x_2 \pm 1) \bmod n_2,\ldots,x_k}$ along dimension two, and so on. (Hereafter, we will omit using "mod" whenever the context is clear.)

In the *one-to-all broadcast* problem, a source node needs to send a message to the rest of the network. To achieve this, we will construct a sequence of *steps*, where a step consists of a number of *link-disjoint* paths each indicating one message delivery; paths of different lengths can co-exist in one step, but the corresponding communications are assumed to complete in about the same time due to the distance-insensitive characteristic of circuit switching. Or, alternatively, a hardware barrier synchronization such as that supported by CM5 [1, 12] can be used after each step. An *α-port* model will be assumed, in which a node can send up to α messages, and simultaneously receive up to α messages, along any α of its $2k$ outgoing and incoming channels, respectively. As in the best case one can multiply the number of nodes owning the broadcast message by $\alpha + 1$ after each step, a lower as follows can be derived.

Lemma 1 *In a kD α-port torus $T_{n_1 \times n_2 \times \cdots \times n_k}$, a lower bound on the number of steps to perform one-to-all broadcast is* $\lceil \log_{\alpha+1}(n_1 n_2, \ldots, n_k) \rceil$.

The following discussion is only concerned with a square $T_{n \times n \times \cdots \times n}$ torus (extension to non-square tori, although possible, is unnecessary for the development of this paper; this will become clear later). We will map $T_{n \times n \times \cdots \times n}$ into a modulo Euclidean integer space $\mathbb{Z}^k$, where $\mathbb{Z} = \{0, 1, \ldots, n-1\}$. We may interchangeably represent node $p_{x_1,x_2,\ldots,x_k}$ in $T_{n \times n \times \cdots \times n}$ as a point $(x_1, x_2, \ldots, x_k)$ in $\mathbb{Z}^k$. A vector in $\mathbb{Z}^k$ is a k-tuple $\vec{v} = (v_1, v_2, \ldots, v_k)$. The *ith positive* (*resp., negative*) *elementary vector* $\vec{e}_i$ (resp., $\vec{e}_{-i}$) of $\mathbb{Z}^k$, $i = 1, \ldots, k$, is the vector with all entries being 0, except the ith entry being 1 (resp., -1). We may write $\vec{e}_{i_1} + \vec{e}_{i_2}$ as $\vec{e}_{i_1,i_2}$, $\vec{e}_{i_1} + \vec{e}_{-i_2}$ as $\vec{e}_{i_1,-i_2}$, and similarly $\vec{e}_{i_1} + \cdots + \vec{e}_{i_m}$ as $\vec{e}_{i_1,\ldots,i_m}$. For instance, $\vec{e}_{1,3} = \vec{e}_1 + \vec{e}_3$ and $\vec{e}_{1,-3} = \vec{e}_1 - \vec{e}_3$. The *linear combination* of vectors (say $a_1\vec{v}_1 + a_2\vec{v}_2$, where a_1 and a_2 are integers) follows the typical definitions in linear algebra, except that a "mod n" is implicitly applied.

Definition 1 In $\mathbb{Z}^k$, given a node x, an m-tuple of vectors $B = (\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_m)$, and an m-tuple of integers $N = (n_1, n_2, \ldots, n_m)$, we define the *span of x by vectors B and distances N* as a set of nodes

$$SPAN(x, B, N) = \left\{ x + \sum_{i=1}^{m} a_i \vec{b}_i | 0 \le a_i < n_i \right\}.$$

Note that the above definition is different from the typical definition of SPAN in linear algebra [7]. We aim at identifying a portion of the torus. Below are some examples:

- The main diagonals of $T_{n\times n}$ and $T_{n\times n\times n}$ can be written as $SPAN(p_{0,0}, (\vec{e}_{1,2}), (n))$ and $SPAN(p_{0,0,0}, (\vec{e}_{1,2,3}), (n))$, respectively.
- The XY-plane passing node $p_{0,0,i}$ in $T_{n\times n\times n}$ is $SPAN(p_{0,0,i}, (\vec{e}_1, \vec{e}_2), (n, n))$.
- $T_{n\times n\times n}$ can be written in several ways: $SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_2, \vec{e}_1), (n, n, n))$ and $SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}, \vec{e}_3), (n, n, n))$ (refer to Figure 7(a) and Figure 8(a), respectively, for illustrations).

3. Broadcasting in 2D tori

Reference [15] already shows a near-optimal broadcast scheme for *wormhole-routed* all-port 2D tori. Below, we first modify the scheme for *circuit-switched* all-port tori using even less numbers of steps. We then generalize to other port models. The discussion is separated into two parts, depending on whether the torus is square or non-square.

3.1. Square cases

3.1.1. When $\alpha = 4$. Consider a square 2D $T_{n\times n}$. Without loss of generality, let the source node be $p_{0,0}$. We denote by M the message to be broadcast.

Stage 1. In this stage, M will be sent to the main diagonal $L_0 = SPAN(p_{0,0}, (\vec{e}_{1,2}), (n))$. For simplicity, we temporarily assume that n is a multiple of five, $n = 5t$. We regard $p_{0,0}$ as the center of L_0 and horizontally slice L_0 into five segments S_i, $i = -2, \ldots, 2$, each containing t nodes (see Figure 1(a) for an illustration). With one step, $p_{0,0}$ can send M to four nodes $p_{-2t,-2t}$, $p_{-t,-t}$, $p_{t,t}$, and $p_{2t,2t}$. The routing is clearly congestion-free.

Now on each S_i, $i = -2, \ldots, 2$, the center node already has M. So we can recursively execute step 1 on each S_i from its center node. This is illustrated in Figure 1(b). The recursion stops when the length of each S_i reduces to one or zero. When n is not a multiple of 5, we simply make the lengths of S_is as even as possible. One important invariant is to always keep the nodes already owning M at the centers of S_is. Overall, this stage takes $\lceil \log_5 n \rceil$ steps to complete.

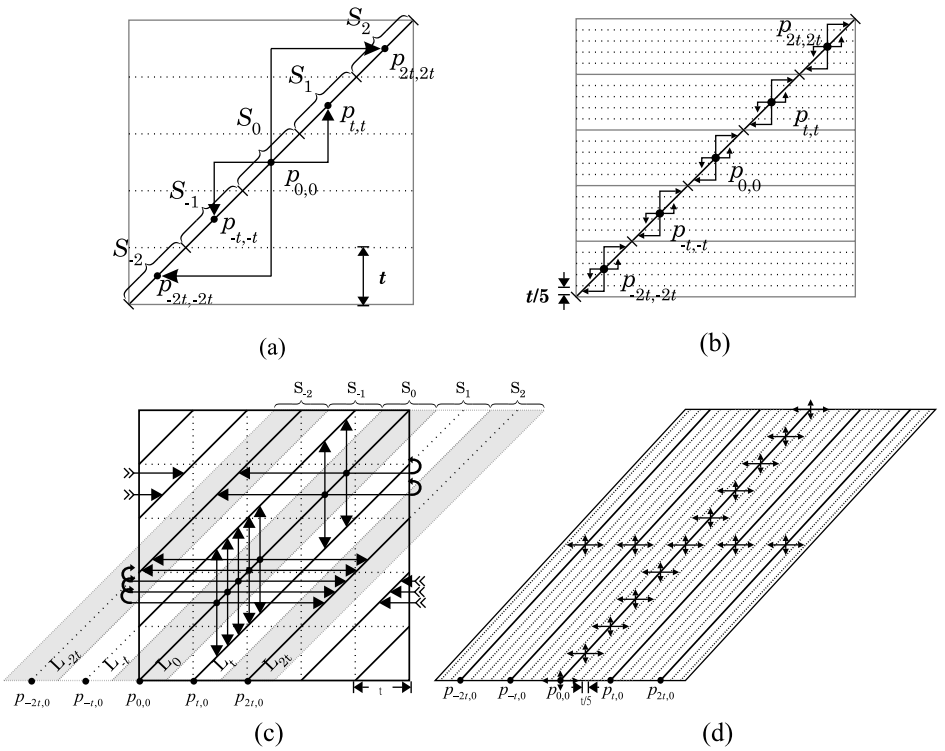

Figure 1. The broadcasting steps in a square 2D torus: (a) step 1 of Stage 1, and (b) step 2 of Stage 1, (c) step 1 of Stage 2, and (d) step 2 of Stage 2.

Stage 2. In this stage, the torus is viewed as n diagonals ($i = -\lfloor \frac{n-1}{2} \rfloor, -\lfloor \frac{n-1}{2} \rfloor + 1, \ldots, \lceil \frac{n-1}{2} \rceil$)

$$L_i = SPAN(p_{i,0}, (\vec{e}_{1,2}), (n)). \tag{1}$$

For simplicity, we again let $n = 5t$. With these diagonals, we then partition the torus into 5 strips S_i, $i = -2, \ldots, 2$, such that first strip consists of the first t diagonals in Eq. (1), the second strip the next t diagonals, and so on. This is illustrated in Figure 1(c).

In the first step, we let L_0 send M to L_{-2t}, L_{-t}, L_t, and L_{2t}. This can be done by having each node $p_{i,i}$ in L_0 send M to nodes $p_{i-2t,i}$, $p_{i,i+t}$, $p_{i,i-t}$, and $p_{i+2t,i}$. The communication, as illustrated in Figure 1(c), is clearly congestion-free.

Now on each S_i, $i = -2 \ldots 2$, the center diagonal L_{it} already has M. So we can recursively repeat step 1 in each S_i. The second step is illustrated in Figure 1(d). The recursion terminates when each S_i contains one or zero diagonal. The modification for n not a multiple of 5 is similar to stage 1. This stage takes $\lceil \log_5 n \rceil$ steps to complete. The result is one step less than that of [15].

Theorem 1 *In a circuit-switched all-port $T_{n\times n}$ torus, broadcast can be done in $2\lceil \log_5 n \rceil$ steps, which number of steps is at most 1 step more than the lower-bound Lemma 1.*

3.1.2. When $\alpha \leq 3$. First, recall Stage 1 of Section 3.1.1, where the diagonal $L_0 = SPAN(p_{0,0,0}, (\vec{e}_{12}), (n))$ is evenly partitioned into five segments recursively. To cope with the α-port model, $\alpha \leq 3$, here only $\alpha + 1$ segments should be obtained in each recursion. The message delivery should be straight-forward. So totally $\lceil \log_{\alpha+1} n \rceil$ steps are needed for the modified Stage 1.

Similarly, the Stage 2 of Section 3.1.1 should be modified to have $\alpha + 1$ strips instead of five strips in each recursion. One can easily derive the result. So we have the following theorem.

Theorem 2 *In a circuit-switched α-port $T_{n\times n}$ torus, broadcast can be done in $2\lceil \log_{\alpha+1} n \rceil$ steps, which number of steps is at most 1 steps more than the lower-bound Lemma 1.*

3.2. Non-square cases

Consider a non-square torus $T_{n_1\times n_2}$ such that $n_1 < n_2$. First, we embed a square dilated tori on the non-square one. Then, we combine the result in Section 3.1.1 and the "squeeze-then-expand" approach in [15] to solve the broadcasting problem. Intuitively, the torus is squeezed into a square one, on which M is distributed. Then, the squeezed torus is expanded to the original one, on which M is further distributed.

Again, a scheme is first proposed for all-port ($\alpha = 4$) tori. Then, we generalize to other port models.

3.2.1. When $\alpha = 4$. We begin with a definition which embeds a square dilated torus on an non-square one so that we can perform the "squeeze-then-expand" approach.

Definition 2 [15] Given a non-square torus $T_{n_1\times n_2}$ such that $n_1 < n_2$, the *dilated torus* induced by $T_{n_1\times n_2}$, denoted as $\widetilde{T}_{n_1\times n_2}$, is an $n_1 \times n_1$ torus consisting of n_1^2 nodes each denoted by $\tilde{p}_{i,j}$, $0 \leq i < n_1$, $0 \leq j < n_1$, where $\tilde{p}_{i,j} = p_{i,\lfloor jn_2/n_1 \rfloor}$.

Intuitively, in the dilated torus $\widetilde{T}_{n_1\times n_2}$, adjacent nodes along the same column are dilated by $\lfloor \frac{n_2}{n_1} \rfloor$ or $\lceil \frac{n_2}{n_1} \rceil$ links in the original torus $T_{n_1\times n_2}$, but there is no dilation for adjacent nodes along the same row. For instance, in Figure 2(a) a non-square $T_{n_1\times n_2}$ is "squeezed" into a square $\widetilde{T}_{n_1\times n_2}$, with n_1 nodes in each side. Now that $\widetilde{T}_{n_1\times n_2}$ is a square torus, we can use it almost like an ordinary torus based on the distance-insensitive characteristic of circuit switching.

Below, we use four stages to perform broadcast. We temporarily assume that n_1 is even. The first two stages will distribute M to every alternatively diagonal

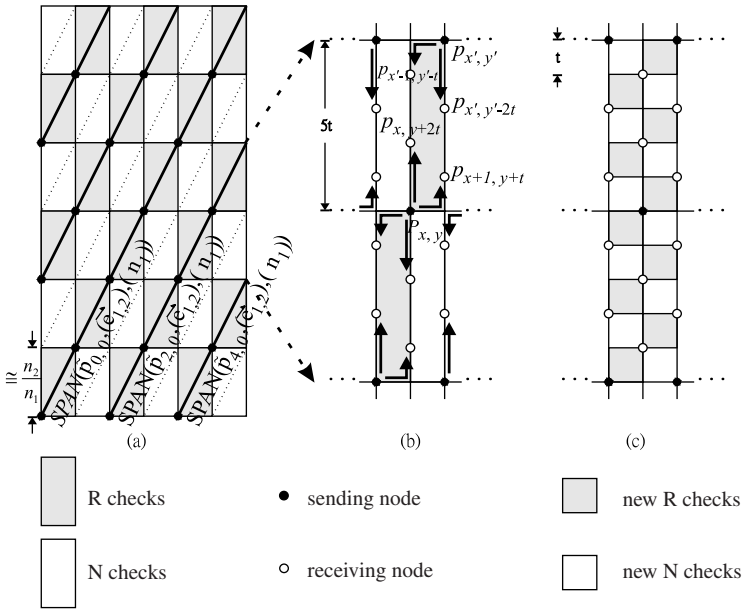

Figure 2. (a) A torus which is considered as a checkerboard after performing stage 2. Here we assume $n_1 = 6$. (b) The communication pattern in a step for R-marked checks. (c) The new checkerboard with smaller checks after performing the step in part (b).

of $\widetilde{T}_{n_1 \times n_2}$. (As mentioned earlier, the torus is squeezed into a square one, on which M is distributed.) The last two stages disseminate M to all nodes in $T_{n_1 \times n_2}$. (Also mentioned intuitively, the squeezed torus is expanded to the original one, on which M is further distributed.) At the end, we will comment on the case of n_1 being odd. As this part is extended from [15], whenever the result of [15] is borrowed, the reader is referred to that paper for more details.

Stage 1. Distribute the broadcast message to $SPAN(\tilde{p}_{0,0}, (\vec{e}_{1,2}), (n_1))$ (i.e., the main diagonal) of $\widetilde{T}_{n_1 \times n_2}$, by applying Stage 1 in Section 3.1.1 to $\widetilde{T}_{n_1 \times n_2}$. This takes $\lceil \log_5 n_1 \rceil$ steps.

Stage 2. The goal of this stage is to send M to the following $\frac{n_1}{2}$ diagonals: $SPAN(\tilde{p}_{2i,0}, (\vec{e}_{1,2}), (n_1)), i = 0, \ldots, \frac{n_1}{2} - 1$ (i.e., alternative diagonals). This can be done by having $SPAN(\tilde{p}_{0,0}, (\vec{e}_{1,2}), (n_1))$ send M to the four diagonals that are $\approx -\frac{2n_2}{5}, -\frac{n_2}{5}, \frac{n_2}{5}$, and $\frac{2n_2}{5}$ hops away, and repeat this process recursively (similar to Section 3.1.1). This stage takes $\lceil \log_5 \frac{n_1}{2} \rceil$ steps.

Stage 3. This stage borrows the result in [15]. We regard $T_{n_1 \times n_2}$ as a checkerboard which contains n_1^2 checks. The checkerboard is formally defined below.

Definition 3 [15] *In* $T_{n_1 \times n_2}$, each smallest submesh in which the lower-left and upper-right corner nodes are the *only* two nodes that have received the broadcast message is regarded as a *check marked by R* (received). Excluding the R-marked

checks, the rest of the checkerboards are considered as a number of *checks marked by N* (non-received).

For instance, Figure 2(a) illustrates a torus (with $n_1 = 6$) after performing Stage 2. Only three alternative diagonals have received M. So each R-marked check's lower-left and upper-right corners must match with two consecutive nodes in some dilated diagonal. The rest of the checkerboard are all marked by N. Note that R-marked checks and N-marked checks must interleave with each other in $T_{n_1 \times n_2}$ [15].

With the checkerboard structure, we will recursively increase the number of checks and decrease size of checks, until the height of each check is less than 5 (*height* = the number of links along the y-axis). Below, we show one recursive step. Consider any R-marked check of height $h \geq 5$ with its lower-left corner being $p_{x,y}$ and upper-right corner being $p_{x',y'}$. For simplicity, let the height $h = 5t$ (a multiple of five). We perform the following communications: (i) $p_{x,y}$ sends two messages to nodes $p_{x+t,y+t}$ and $p_{x,y+2t}$, and (ii) $p_{x',y'}$ sends two messages to nodes $p_{x'-t,y'-t}$ and $p_{x',y'-2t}$. The communication is illustrated in Figure 2(b).

After this step, each R-marked check will be partitioned into five smaller checks, three of which marked R and two of which marked N. Similarly, each N-marked check is partitioned into five smaller checks, but only two of them are marked R, and the rest three N. For instance, the four checks in Figure 2(b), after performing this step, will be partitioned into 20 smaller checks as shown in Figure 2(c). The above recursion is repeated until the height of every check is less than 5. The total number of steps required in this stage is $\lceil \log_5 \frac{n_2}{n_1} \rceil - 1$.

Stage 4. It remains to distribute M to every un-received node in R-marked checks. One possible solution is shown in Figure 3. As each N-marked check is surrounded by R-marked checks, this implies the completion of broadcast. The number of steps required is at most two.

Comment. When n_1 is odd, the alternative diagonals used in Stage 2 are not well-defined. So we first translate $T_{n_1 \times n_2}$ into a smaller torus $T_{(n_1-1)\times n_2}$, by removing any column in the former, using the dilation concept. Then, apply Stages 1–4 on $T_{(n_1-1)\times n_2}$. Finally, send M to nodes on the removed column using one step.

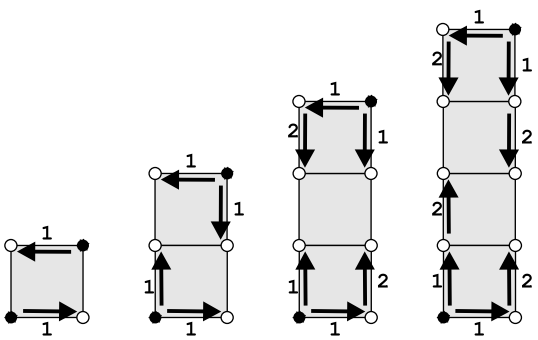

Figure 3. The message distribution in an R-marked check of height 1, 2, 3, and 4.

Theorem 3 *In a circuit-switched all-port $T_{n_1 \times n_2}$ torus such that $n_1 < n_2$, broadcast can be done within*

$$\lceil \log_5 n_1 \rceil + \left\lceil \log_5 \frac{n_1}{2} \right\rceil + \left\lceil \log_5 \frac{n_2}{n_1} \right\rceil + c$$

steps, where $c = 1$ (resp., 2) when n_1 is even (resp., odd), which number of steps is at most 3 (*resp., four*) *more than the lower bound in Lemma* 1.

3.2.2. When $\alpha = 3$. Below, we modify the all-port algorithm in Section 3.2.1 for a 3-port model. The main difficulty is in redefining Definition 2 (dilated torus) and Definition 3 (check structure). In the following discussion, we assume that n_1 is even, with the understanding that one more step is required when n_1 is odd (refer to the comment at the end of Section 3.2.1). Also, to avoid the tedium of using floor and ceiling functions, we assume that n_2 is a multiple of n_1.

Definition 4 Given a non-square torus $T_{n_1 \times n_2}$ such that $n_1 < n_2$, the dilated torus induced by $T_{n_1 \times n_2}$, denoted as $\widehat{T}_{n_1 \times n_2}$, is an $n_1 \times n_1$ torus consisting of nodes from the following four $\frac{n_1}{2} \times \frac{n_1}{2}$ tori:

$$T_{0,0} = SPAN(p_{0,0}, B_2, N_2),$$
$$T_{1,0} = SPAN(p_{1,0}, B_2, N_2),$$
$$T_{0,1} = SPAN(p_{0,\frac{4}{3}\frac{n_2}{n_1}}, B_2, N_2),$$
$$T_{1,1} = SPAN(p_{1,\frac{4}{3}\frac{n_2}{n_1}}, B_2, N_2),$$

where $B_2 = (2\vec{e}_1, \frac{2n_2}{n_1}\vec{e})$ and $N_2 = (\frac{n_1}{2}, \frac{n_1}{2})$. $\widehat{T}_{n_1 \times n_2}$ has n_1^2 nodes which are denoted by $\hat{p}_{i,j}$, for $i, j = 0, \ldots, n_1 - 1$.

Intuitively, the dilated torus in Definition 4 is partitioned into four sub-tori, each being dilated two times longer and having half of the nodes in each dimension than the earlier one in Definition 2. For instance, Figure 4(a) shows the four dilated tori in an $n_1 \times n_2$ torus ($n_1 = 6$). Naturally, $\widehat{T}_{n_1 \times n_2}$ still has n_1 diagonals. However, we now do not have "straight" diagonals as opposed to Figure 2.

Broadcasting is still done in four stages with similar philosophy.

Stage 1. Spread M to $SPAN(\hat{p}_{0,0}, (\vec{e}_{1,2}), (n_1))$, by applying the Stage 1 in Section 3.1.2. This takes $\lceil \log_4 n_1 \rceil$ steps.

Stage 2. Spread M to $\frac{n_1}{2}$ diagonals, $SPAN(\hat{p}_{2i,0}, (\vec{e}_{1,2}), (n_1)), 0, \ldots, \frac{n_1}{2} - 1$, by applying the Stage 2 of Section 3.1.2. This takes $\lceil \log_4 \frac{n_1}{2} \rceil$ steps. Now, nodes of $T_{0,0}$ and $T_{1,1}$ already have M.

Stage 3. This stage is based on the recursive structure below. First, we regard the torus $T_{n_1 \times n_2}$ as a checkerboard using Definition 3. We further classify R-marked checks on the checkerboard as follows (see Figure 4(a) for an example).

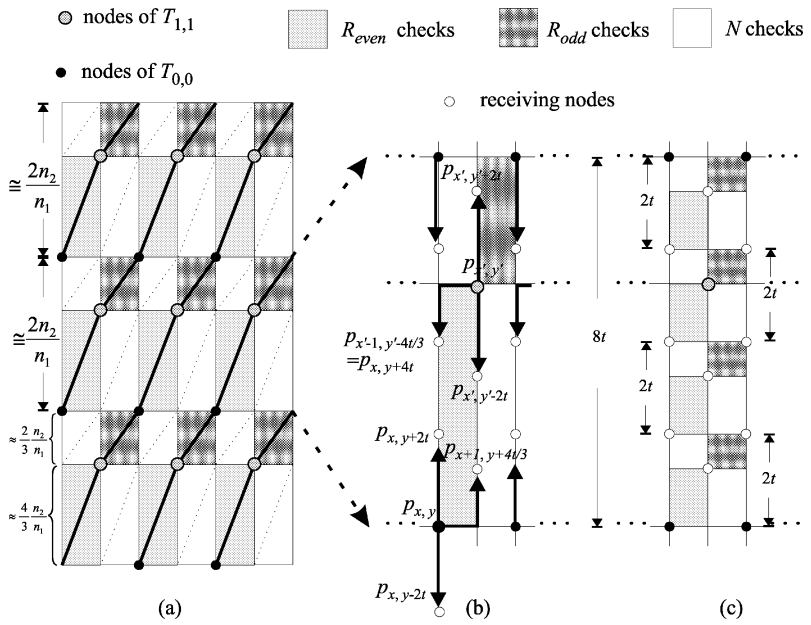

Figure 4. (a) A 3-port $T_{n_1 \times n_2}$ which is considered as four dilated $\frac{n_1}{2} \times \frac{n_1}{2}$ tori ($n_1 = 6$). The lines in bold are the alternative diagonals used in Stage 2. (b) The communication pattern in a recursive step of Stage 3. (c) The new checkerboard with four smaller rectangles after performing the step in part (b).

Definition 5 A check marked by R is classified as R_{even} if its lower-left node's index along the x-axis is even, and classified as R_{odd} otherwise.

The recursion should proceed as long as the sum of the heights of two consecutive R_{even} and R_{odd} is ≥ 8. Let's consider two consecutive R_{even}, R_{odd} checks, and two neighboring N checks that form a rectangle (refer to Figure 4(b)). For ease of presentation, let the height h of the rectangle be a multiple of eight, $h = 8t$. We perform the following communications:

- for each $p_{x,y}$ located at the lower-left corner of a R_{even}-check, $p_{x,y}$ sends three messages to $p_{x,y+2t}$, $p_{x,y-2t}$, and $p_{x+1,y+\frac{4}{3}t}$, and
- for each $p_{x,y}$ located at the lower-left corner of a R_{odd}-check, $p_{x,y}$ sends three messages to nodes $p_{x,y+2t}$, $p_{x,y-2t}$, and $p_{x-1,y+\frac{4}{3}t}$.

After this step, the rectangle will be partitioned into four smaller rectangles as shown in Figure 4(c). The recursion maintains an important invariant:

I1. The ratio of the height of R_{even}-checks and the height of R_{odd}-checks is (or close to) 2 : 1.

To prove, first observe that **I1** already holds true from Definitions 4 and 5. Furthermore, the communication step maintains this invariant.

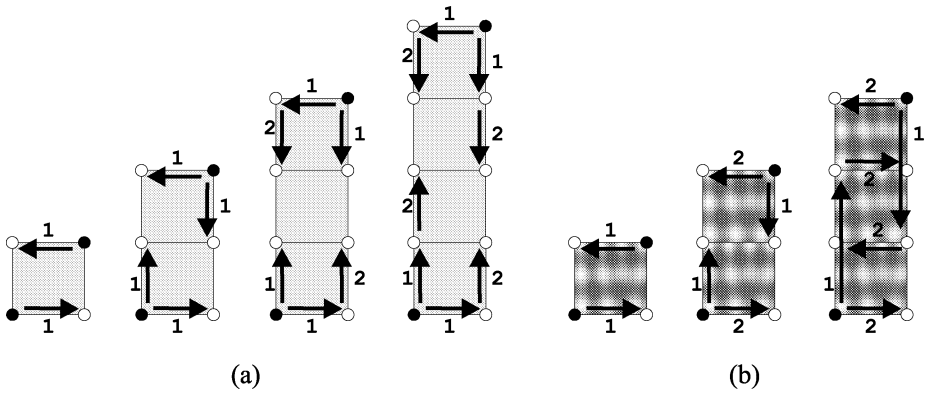

Figure 5. (a) Sending M in a R_{even}-check of heights 1, 2, 3, and 4, and (b) sending M in a R_{odd}-check of heights 1, 2, and 3.

The above recursion is repeated until the height of every rectangle is less than 8. As the initial height of the first rectangle is upper-bounded by $\lceil \frac{2n_2}{n_1} \rceil$ and the rectangle height is reduced by a factor of 4 after each recursion, this stage will take $\lceil \log_4 \frac{2n_2}{n_1} \rceil - 1$ steps.

Stage 4. At the end of Stage 3, it is possible to manage the height of each R_{even} and R_{odd} checks not exceeding 4 and 3, respectively. For each possible height, we show one possible solution in Figure 5 to send M to nodes in R_{even} and R_{odd} checks. Note how the 3-port model is observed in the communication. Also note that no communication is scheduled for N checks as their boundaries must be parts of some R_{even}- and R_{odd}-checks. The number of steps required is at most two.

Theorem 4 *In a circuit-switched non-square* 3*-port* $T_{n_1 \times n_2}$ *torus such that* $n_1 < n_2$*, broadcast can be done within*

$$\lceil \log_4 n_1 \rceil + \left\lceil \log_4 \frac{n_1}{2} \right\rceil + \left\lceil \log_4 \frac{2n_2}{n_1} \right\rceil + c$$

steps, where $c = 1$ (*resp*. 2) *when* n_1 *is even* (*resp*. *odd*), *which number of steps is at most* 4 (*resp*. 5) *steps more than the lower bound in Lemma* 1.

***3.2.3. When* $\alpha = 1$ *and* $= 2$.** As our scheme follows a dimension-by-dimension approach, when $\alpha = 1$ or 2, the rows/columns of the torus already give a natural solution. So a simple recursive doubling/tripling on rows and columns of the torus will do the job.

4. Broadcasting in 3D tori

Next, we extend our results to α-port 3D tori. Similar to Section 3, we will span the nodes receiving M from the source to a line of nodes, to a plane of nodes, and then

to the whole torus. We extend the "squeeze-then-expand" approach in [15] for 2D tori to 3D tori. The case of α being odd will bring more difficulty to the "expand" part. We will propose a general solution to it.

The cases of $\alpha = 1$ and 2 can be solved trivially as commented in Section 3.2.3. So the following discussion will focus on the $\alpha \geq 3$ cases.

4.1. Square cases

***4.1.1. When* $\alpha = 6$.** Consider a 3D $T_{n\times n\times n}$ with any n. Without loss of generality, let $p_{0,0,0}$ be the source node. The basic idea is to distribute the broadcast message M in three stages: (i) from $p_{0,0,0}$ to the line $SPAN(p_{0,0,0}, (\vec{e}_{1,3}), (n))$, (ii) from the above line to the plane $SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n))$, and then (iii) from the above plane to the whole torus. For simplicity, we may use X-, Y-, and Z-axes to refer to the first, second, and third dimensions, respectively.

Stage 1: From the source node to a line. To send M to $L_0 = SPAN(p_{0,0,0}, (\vec{e}_{1,3}), (n))$, we use the following recursive structure. For simplicity, let n be a multiple of 7, $n = 7t$. We view L_0 as consisting of n nodes $p_{i,0,i}$, $i = -\lfloor \frac{n-1}{2} \rfloor, -\lfloor \frac{n-1}{2} \rfloor + 1, \ldots, \lceil \frac{n-1}{2} \rceil$. We then partition L_0 horizontally into 7 segments S_j, $j = -3, \ldots, 3$, such that the first segment consists of the first t nodes, the second segment the next t nodes, etc. Let's identify the center node of S_j as m_j. In one step, node m_0 can forward M to $m_{\pm 1}, m_{\pm 2}, m_{\pm 3}$, as illustrated in Figure 6.

Clearly, we can recursively distribute M to nodes of S_j from m_j, $j = -3 \ldots 3$. This stage will take $\lceil \log_7 n \rceil$ steps to complete.

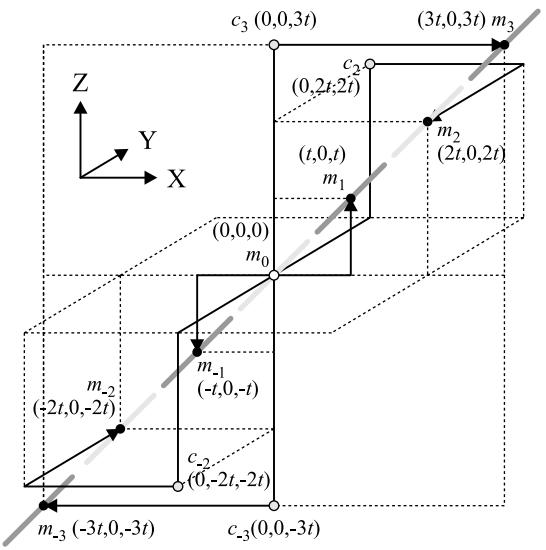

Figure 6. Stage 1 of broadcasting in a square 3D torus: the first step.

Stage 2: From a line to a plane. In this stage, M will be distributed from the L_0 to the plane $P_0 = SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_2), (n, n))$. However, we view the plane as consisting of n lines:

$$L_i = SPAN(p_{i,i,0}, (\vec{e}_{1,3}), (n)),$$
$$i = -\left\lfloor \tfrac{n-1}{2} \right\rfloor, -\left\lfloor \tfrac{n-1}{2} \right\rfloor + 1, \ldots, \left\lceil \tfrac{n-1}{2} \right\rceil. \tag{2}$$

See Figure 7(a) for an illustration.

It is easy to send messages from a line to another parallel line in one communication step. For instance, to deliver messages from line $SPAN(p_{0,0,0}, (\vec{e}_{1,3}), (n))$

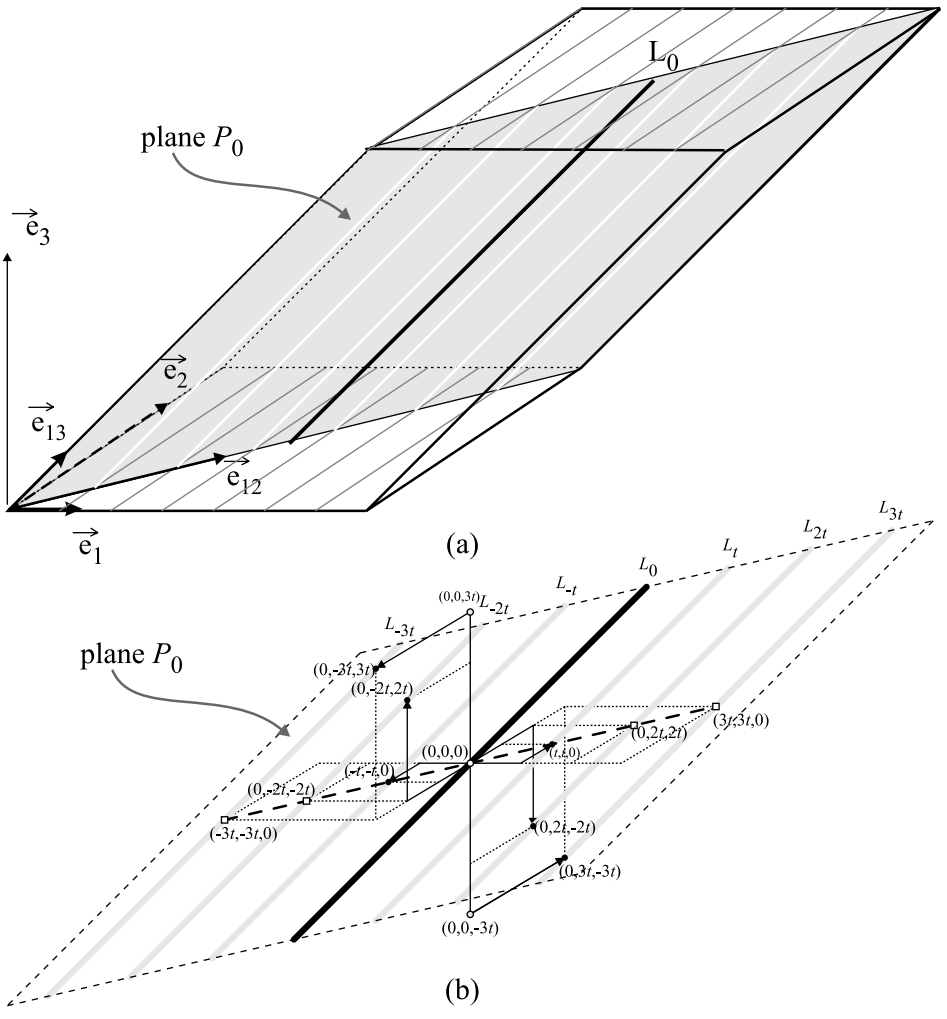

Figure 7. Stage 2 of broadcasting in a square 3D torus: (a) viewing the torus from the perspective $P_0 = SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_2, \vec{e}_1), (n, n, n))$ which is partitioned into seven strips, and (b) the communication pattern in one step.

to line $SPAN(p_{2,3,4}, (\vec{e}_{1,3}), (n))$, we simply let each $p_{i,0,i}$ (of the former line) send to $p_{i+2,3,i+4}$ (of the latter line). One can easily generalize this to a line sending to six other parallel lines in one step.

This stage is based on a recursive structure as follows. For simplicity, let $n = 7t$. We partition the plane P_0 into seven strips S_j, $j = -3, \ldots, 3$, such that S_{-3} consists of the first t lines in Eq. (2), S_{-2} the next t lines, etc. (refer to Figure 7(b)). By having each $p_{i,0,i} \in L_0$ send M to the following six nodes:

$$p_{i+t,t,i}, \quad p_{i,2t,i-2t}, \quad p_{i,3t,i-3t}, \quad p_{i-t,-t,i}, \quad p_{i,-2t,i+2t}, \quad p_{i,-3t,i+3t},$$

we can distribute M to the following six lines in one step:

$$\begin{aligned} L_{\pm t} &= SPAN(p_{\pm t,\pm t,0}, (\vec{e}_{1,3}), (n)), \\ L_{\pm 2t} &= SPAN(p_{0,\pm 2t,\mp 2t}, (\vec{e}_{1,3}), (n)), \\ L_{\pm 3t} &= SPAN(p_{0,\pm 3t,\mp 3t}, (\vec{e}_{1,3}), (n)). \end{aligned}$$

This communication step, as illustrated in Figure 7(b), is congestion-free. The resulting line L_{jt} is on the central line of S_j for all $j = -3, \ldots, 3$. To see this, let's prove the case of L_{2t}:

$$\begin{aligned} L_{2t} &= SPAN(p_{0,2t,-2t}, (\vec{e}_{1,3}), (n)), \\ &= SPAN(p_{0,2t,-2t} + 2t\vec{e}_{1,3}, (\vec{e}_{1,3}), (n)), \\ &= SPAN(p_{2t,2t,0}, (\vec{e}_{1,3}), (n)), \\ &\in P_0, \end{aligned}$$

which is indeed the central plane of S_2. The other cases can be proved similarly.

Next, we can recursively perform the similar line-to-line distribution in each S_j using L_{jt} as the source. The recursion is repeated until each S_j is reduced to one or zero line. This stage takes $\lceil \log_7 n \rceil$ steps to complete.

Stage 3: From a plane to more planes. In this stage, we view the torus from another perspective:

$$SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}, \vec{e}_1), (n, n, n)), \tag{3}$$

which is illustrated in Figure 8(a). With this view, we partition the torus along the direction $\vec{e}_1$ into n planes:

$$P_i = SPAN(p_{i,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n)), \quad i = -\left\lfloor \frac{n-1}{2} \right\rfloor, \ldots, \left\lceil \frac{n-1}{2} \right\rceil. \tag{4}$$

For simplicity, let $n = 7t$. Following the same philosophy as before, we divide the torus into seven cubes C_j, $j = -3, \ldots, 3$, such that the first cube consists of the first t planes, the second cube the next t planes, etc. This is shown in Figure 8(b).

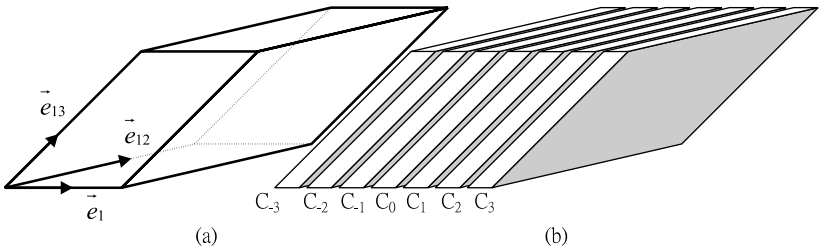

Figure 8. (a) Viewing the torus from the perspective $SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}, \vec{e}_1), (n, n, n))$, and (b) partitioning the torus into seven cubes C_j, $j = -3, \ldots, 3$.

The central plane P_0 in Eq. (4) already owns message M. In this stage, plane-to-plane message distribution will be performed. For instance, if every node on plane $SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n))$ sends M along the Y- and Z-axes to nodes that are $+3$ and $+5$ hops way, respectively, then two planes will receive M:

$$SPAN(p_{0,3,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n)) = SPAN(p_{-3,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n)) = P_{-3}$$
$$SPAN(p_{0,0,5}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n)) = SPAN(p_{-5,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n)) = P_{-5}.$$

Thus, by having each node $p_{a,b,c} \in P_0$ send M to the following six nodes:

$$p_{a+t,b,c}, p_{a,b-2t,c}, p_{a,b,c-3t}, p_{a-t,b,c}, p_{a,b+2t,c}, p_{a,b,c+3t},$$

we can distribute M to six other planes in one step:

$$P_{jt} = SPAN(p_{jt,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n)), j = -3 \ldots 3.$$

The resulting planes P_{jt} is the central plane of C_j for all $j = -3 \ldots 3$.

Next, we can recursively perform the similar plane-to-plane distribution in each C_j using P_{jt} as the source. Totally this stage takes $\lceil \log_7 n \rceil$ steps.

Theorem 5 *In a circuit-switched all-port $T_{n \times n \times n}$ torus, broadcast can be done in $3\lceil \log_7 n \rceil$ steps, which number of steps is at most two steps more than the lower bound in Lemma* 1.

4.1.2. When $3 \leq \alpha \leq 5$. For an α-port torus $T_{n \times n \times n}$, $3 \leq \alpha \leq 5$, we modify the scheme developed in Section 4.1.1 as follows. In Stage 1, the line $L_0 = SPAN(p_{0,0,0}, (\vec{e}_{12}), (n))$ is evenly partitioned into $\alpha + 1$ segments recursively. The message is delivered from source node to one representative node of each segment recursively. At the end of recursion, the nodes in L_0 receive the message. Similarly, in Stage 2, the plane $P_0 = SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}), (n, n))$ is evenly partitioned into $\alpha + 1$ strips recursively. We can recursively distribute the message from L_0 to one representative line of nodes in each strip. Eventually, the plane P_0 receives the message. In Stage 3, the torus $SPAN(p_{0,0,0}, (\vec{e}_{1,3}, \vec{e}_{1,2}, \vec{e}_1), (n, n, n))$ is evenly partitioned into $\alpha + 1$ cubes recursively. The message delivery should be straight-forward.

Theorem 6 *In a circuit-switched α-port $T_{n\times n\times n}$ torus, broadcast can be done in $3\lceil \log_{\alpha+1} n\rceil$ steps, which number of steps is at most two steps more than the lower bound in Lemma* 1.

4.2. Non-square cases

Consider a non-square and α-port torus $T_{n_1\times n_2\times n_3}$, $3 \le \alpha \le 6$. Without loss of generality, let $n_1 = \min\{n_1, n_2, n_3\}$. Similar to Section 3.2, we assume that n_1 is even; otherwise, we can translate the torus into $T_{(n_1-1)\times n_2\times n_3}$, which will require one more step to perform broadcast.

As before, we will "squeeze" the torus into a square one and then "expand" it back. As will be seen later, the "expand" part is more difficult, especially when α is odd.

4.2.1. When α is even. Here, our approach is still based on recursively partitioning lines/planes/cubes into $\alpha + 1$ parts. Below we first discuss the case of $\alpha = 6$. The case of $\alpha = 4$ can be solved similarly and we will briefly comment on it at the end of this section.

To support the "squeeze-then-expand" approach, we first define a dilated tori as follows.

Definition 6 The dilated torus induced by $T_{n_1\times n_2\times n_3}$, denoted as $\check{T}_{n_1\times n_2\times n_3}$, is an $n_1 \times n_1 \times n_1$ torus consisting of n_1^3 nodes each denoted as $(0 \le i, j, k < n_1)$:

$$\check{p}_{i,j,k} = p_{i,\lfloor jn_2/n_1\rfloor,\lfloor kn_3/n_1\rfloor}.$$

Intuitively, in $\check{T}_{n_1\times n_2\times n_3}$, adjacent nodes along the same y-axis (resp., z-axis) are dilated by $\lfloor \frac{n_2}{n_1}\rfloor$ or $\lceil \frac{n_2}{n_1}\rceil$ (resp., $\lfloor \frac{n_3}{n_1}\rfloor$ or $\lceil \frac{n_3}{n_1}\rceil$) links in the original torus $T_{n_1\times n_2\times n_3}$, but there is no dilation along the x-axis. On $\check{T}_{n_1\times n_2\times n_3}$, we further define two (dilated) subtori:

$$\check{T}_{0,0,0} = SPAN\left(\check{p}_{0,0,0}, (2\vec{e}_1, 2\vec{e}_2, 2\vec{e}_3), \left(\frac{n_1}{2}, \frac{n_1}{2}, \frac{n_1}{2}\right)\right),$$

$$\check{T}_{1,1,1} = SPAN\left(\check{p}_{1,1,1}, (2\vec{e}_1, 2\vec{e}_2, 2\vec{e}_3), \left(\frac{n_1}{2}, \frac{n_1}{2}, \frac{n_1}{2}\right)\right).$$

That is, $\check{T}_{0,0,0}$ and $\check{T}_{1,1,1}$ are dilated by two links along each dimension in $\check{T}_{n_1\times n_2\times n_3}$.

Assuming $\check{p}_{0,0,0}$ as the source node, the broadcast is done in five stages.

Stage 1. The source $\check{p}_{0,0,0}$ sends M to $\check{p}_{1,1,1}$.

Stage 2. Node $\check{p}_{0,0,0}$ performs broadcast on $\check{T}_{0,0,0}$, and node $\check{p}_{1,1,1}$ performs broadcast on $\check{T}_{1,1,1}$. This can be done in parallel as $\check{T}_{0,0,0}$ and $\check{T}_{1,1,1}$ are independent of each other. We can apply the scheme in Section 4.1.1, which will take $3\lceil \log_7 \frac{n_1}{2}\rceil$ steps to complete.

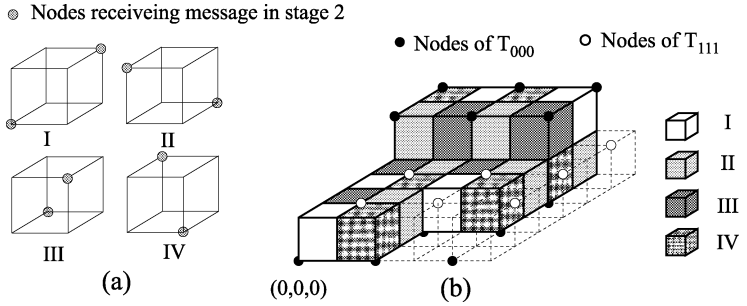

Figure 9. (a) Four types of unit cubes and (b) the geometric relationship of these four types of unit cubes.

Stage 3. After stage 2, there are 1/4 nodes on $\check{T}_{n_1 \times n_2 \times n_3}$ already having M. If we look at each unit cube (size $1 \times 1 \times 1$) in $\check{T}_{n_1 \times n_2 \times n_3}$, two out of eight nodes in the unit cube already have M. According the distribution of the nodes having M, we can classify a unit cube into four types, as shown in Figure 9(a). In Figure 9(b), we further show the geometric relationship of these four types of unit cubes in the torus.

Now we show how the recursion proceeds in one step. On each unit cube of type I, let its two corner nodes already owning M be: $p_{x,y,z}$ and $p_{x',y',z'}$ (refer to Figure 10). Let the width of the unit cube on the original torus be $w = y' - y$. The recursion should proceed as long as $w \geq 7$. For ease of presentation, suppose w is a multiple of 7, $w = 7t$. Then we perform the following communications:

- $p_{x,y,z}$ sends M to three nodes $p_{x,y+2t,z}$, $p_{x,y+4t,z}$, and $p_{x,y+6t,z}$, and
- $p_{x',y',z'}$ sends M to three nodes $p_{x',y'-2t,z'}$, $p_{x',y'-4t,z'}$, $p_{x',y'-6t,z'}$.

The communication is illustrated in Figure 10(a).

Now observe the type I unit cube in Figure 10(a). More nodes have received M. If we consider the pattern of the nodes owning M, then after the above communication step the unit cube can be further partitioned into seven smaller unit cubes, four of which are of type I and three of which are of type III. Similarly, the type II, III, and IV unit cubes in Figure 10(a) are now each partitioned into seven more smaller

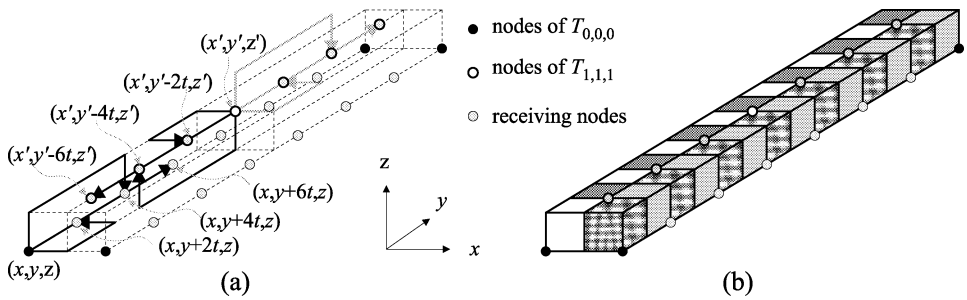

Figure 10. (a) The communication pattern of one step in a type I unit cube. (b) Each unit cube is partitioned into seven smaller unit cubes after performing the step in (a).

cubes of different types. The rest is shown in Figure 10(b). An interesting property is that there are now totally 28 smaller unit cubes (which are evenly distributed to each type) obtained after the communication step. Also, the width of each cube along the Y axis is reduced by a factor of about $\frac{1}{7}$. As the initial value of w is no more than $\lceil \frac{n_2}{n_1} \rceil$, the total number of steps required in this stage is $\lceil \log_7 \frac{n_2}{n_1} \rceil - 1$.

Stage 4. Let's summarize what we have done in stage 3. In stage 3, we have "expanded" $\breve{T}_{n_1 \times n_2 \times n_3}$ from one with n_1^3 unit cubes to one with more unit cubes along the Y-axis. Furthermore, each unit cube is dilated along the Y-axis by no more than six links. In this stage, we will further "expand" the torus such that each unit cube is dilated along the Z-axis by no more than six links, too.

Now we show how the recursion proceeds in one step. The geometric relationship of unit cubes is shown in Figure 11(a). On each unit cube of type I, let its two corner nodes already owning M be: $p_{x,y,z}$ and $p_{x',y',z'}$ (refer to Figure 11(b)). Let the height of the unit cube be $h = z' - z$. The recursion should proceed as long as $h \geq 7$. For ease of presentation, suppose h is a multiple of 7, $h = 7t$. Then we perform the following communications:

- $p_{x,y,z}$ sends three messages to nodes $p_{x,y+2t,z}$, $p_{x,y+4t,z}$, and $p_{x,y+6t,z}$.
- $p_{x',y',z'}$ sends three messages to nodes $p_{x',y'-2t,z'}$, $p_{x',y'-4t,z'}$, and $p_{x',y'-6t,z'}$.

The communication is illustrated in Figure 11(b).

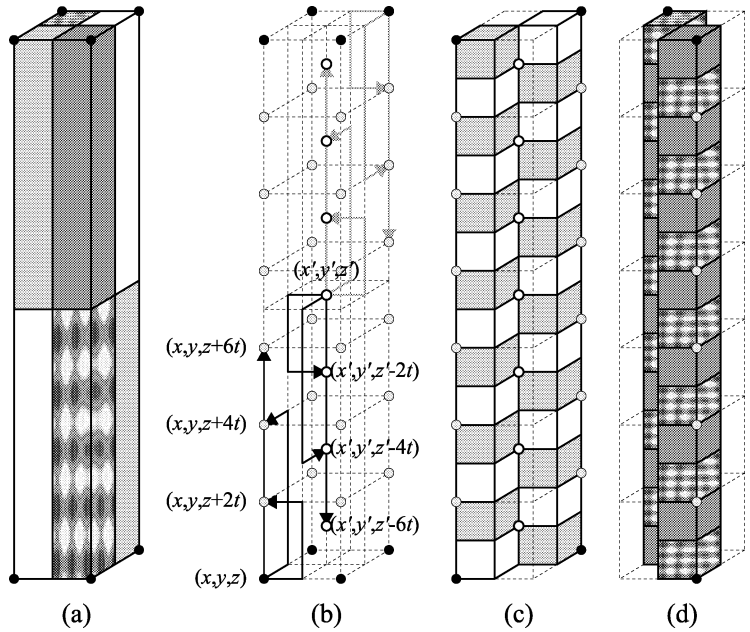

Figure 11. (a) The geometric relationship of eight neighboring unit cubes. (b) The communication pattern in one step for a type I unit cube. (c) and (d) Each unit cube is partitioned into seven smaller new cubes after performing the step in (a).

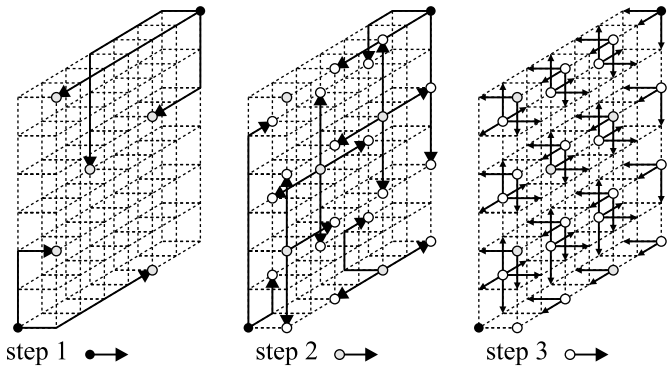

Figure 12. The message distribution in a type I unit cube which is a $2 \times 7 \times 7$ mesh.

After the communication step, more nodes will own M. Using the pattern in Figure 9, we can define new unit cubes. The result is shown in Figure 11(c) and (d), where one can see that each unit cube is now partitioned into seven smaller unit cubes.

The above recursion is repeated until $h < 7$. As the initial value of h is no more than $\lceil \frac{n_3}{n_1} \rceil$, the total number of steps required is $\lceil \log_7 \frac{n_3}{n_1} \rceil - 1$.

Stage 5. After Stage 4, each unit cube of type I is dilated by at most six links along each of Y and Z axes (there is no dilation along the X-axis). So there are 36 possible sizes of the cube. Due to space limitation, we only show how the largest case is solved in Figure 12. This stage takes at most three steps.

Theorem 7 *In a circuit-switched* 6*-port non-square torus* $T_{n_1 \times n_2 \times n_3}$*, broadcast can be done within*

$$3\left\lceil \log_7 \frac{n_1}{2} \right\rceil + \left\lceil \log_7 \frac{n_2}{n_1} \right\rceil + \left\lceil \log_7 \frac{n_3}{n_1} \right\rceil + c$$

steps, where $c = 2$ (*resp.*, 3) *when* n_1 *is even* (*odd*)*, which number of steps is at most five* (*six*) *steps more than the lower bound in Lemma* 1.

Comment. When $\alpha = 4$, the communication patterns in Stages 3 and 4 should be modified to ones as shown in Figure 13. As can be seen, each unit cube is partitioned into 5, instead of 7, more smaller unit cubes after each communication step. Also, the recursion should be proceeded as long as the value of w or h is larger than 4.

Theorem 8 *In a circuit-switched* 4*-port torus* $T_{n_1 \times n_2 \times n_3}$*, broadcast can be done within*

$$3\left\lceil \log_{\alpha+1} \frac{n_1}{2} \right\rceil + \left\lceil \log_{\alpha+1} \frac{n_2}{n_1} \right\rceil + \left\lceil \log_{\alpha+1} \frac{n_3}{n_1} \right\rceil + c$$

steps, where $c = 2$ (*resp.*, *three*) *when* n_1 *is even* (*resp.*, *odd*)*, which number of steps is at most five* (*resp.*, *six*) *steps more than the lower bound in Lemma* 1.

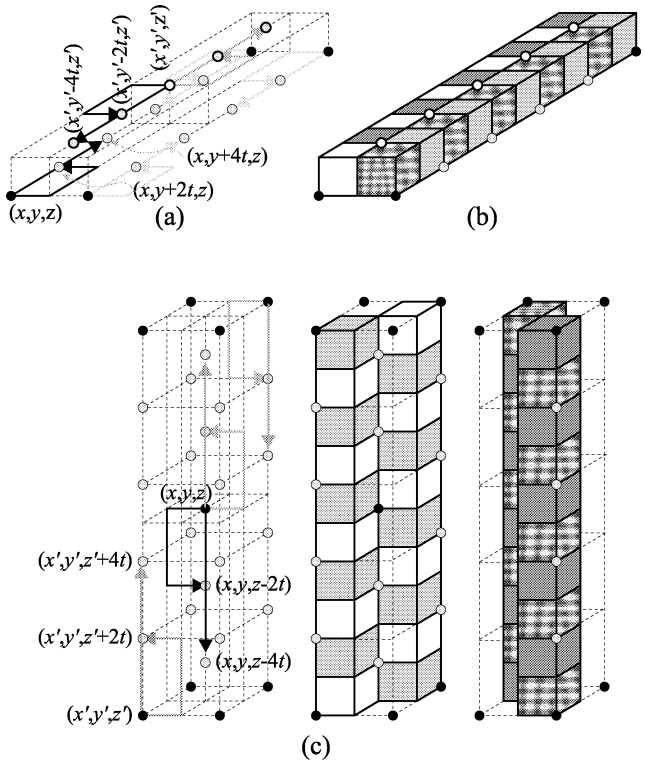

Figure 13. The communication patterns of Stages 3 and 4 in 4-port tori: (a) the communication pattern of one step in Stage 3, (b) each unit cube is partitioned into five smaller unit cubes after performing the step in (a), (c) the communication pattern of one step in Stage 4 and each unit cube is partitioned into five smaller unit cubes after performing the step in the leftmost of (c).

4.2.2. When α is odd. Earlier, when α is even, we used the concept of unit cube in a recursive manner. After one recursive step, the number of nodes owning M in a unit cube remains as an *even* number. This is important to maintain the recursive structure. When α is odd, in order to maintain such structure, we have to redefine the dilated torus. To avoid the tedium of using floor and ceiling functions, we assume that n_2 and n_3 are each a multiple of n_1.

Definition 7 Given a non-square torus $T_{n_1 \times n_2 \times n_3}$, the dilated torus induced by $T_{n_1 \times n_2 \times n_3}$, denoted as $\breve{T}_{n_1 \times n_2 \times n_3}$, is an torus consisting of nodes from the following eight $\frac{n_1}{2} \times \frac{n_1}{2} \times \frac{n_1}{2}$ tori:

$$T_{a,b,c} = SPAN\left(p_{a,b\frac{\alpha+1}{\alpha}\frac{n_2}{n_1},c\frac{\alpha+1}{\alpha}\frac{n_3}{n_1}}, B_3, N_3\right),$$

where $a, b, c = 0 \ldots 1$, $B_3 = (2\vec{e}_1, \frac{2n_2}{n_1}\vec{e}_2, \frac{2n_3}{n_1}\vec{e}_3)$ and $N_3 = (\frac{n_1}{2}, \frac{n_1}{2}, \frac{n_1}{2})$. $\breve{T}_{n_1 \times n_2 \times n_3}$ has n_1^3 nodes, which are denoted by $\breve{p}_{i,j,k}$ for $i, j, k = 0, \ldots, n_1 - 1$.

One important property of this definition is that the ratio of the dilations on the Y axis is $(\alpha + 1) : (\alpha - 1)$, and that the ratio of the dilations on the Y-axis

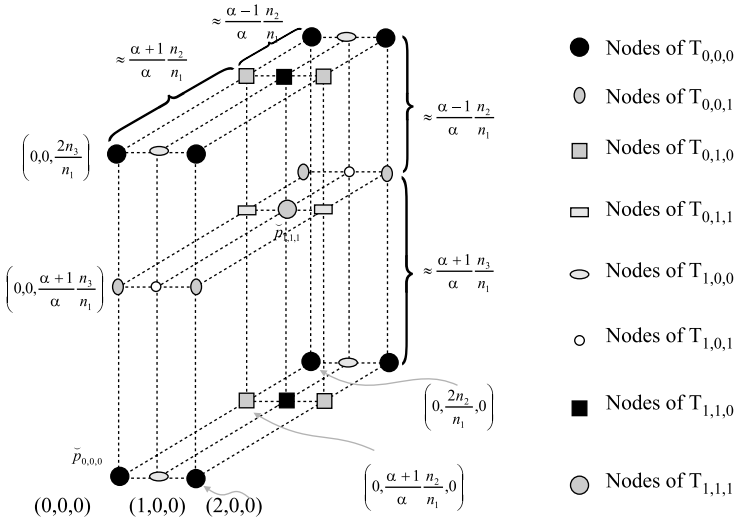

Figure 14. Illustration of Definition 7.

is $(\alpha + 1) : (\alpha - 1)$, too. Recall Section 3.2.2 (3-port, 2D torus), the dilated torus is comprised of sub-tori $(T_{0,0}, T_{0,1}, T_{1.0}, T_{1,1})$. Here for a 3D torus, we use eight sub-tori. Figure 14 illustrates this definition.

Let $\breve{p}_{0,0,0}$ be the source node. There are five stages to perform broadcast.

Stage 1. Node $\breve{p}_{0,0,0}$ sends M to $\breve{p}_{1,1,1}$.

Stage 2. Node $\breve{p}_{0,0,0}$ and node $\breve{p}_{1,1,1}$ concurrently perform broadcast on $T_{0,0,0}$ and $T_{1,1,1}$, respectively, by applying the result in Section 4.1.2. This takes $3\lceil \log_{(\alpha+1)} \frac{n_1}{2} \rceil$ steps.

Stage 3. In this stage, we will "expand" nodes owning M along the Y axis. From nodes already owning M, we can define a number of unit cubes as in the previous section. Also, according to the distribution of the nodes owning M, we can classify the unit cubes into four types, I, II, III, and IV, according to Figure 9(a).

Now consider two consecutive unit cubes of type I as shown in Figure 15(a). Let $p_{x,y,z}$, $p_{x',y',z'}$, $p_{x'',y'',z''}$ be the nodes already owning M. We can assume without loss of generality that $p_{x,y,z}$ is a node in $T_{0,0,0}$, and $p_{x',y',z'}$ one in $T_{1,1,1}$. By Definition 7, it must be that $(y' - y) : (y'' - y') \approx (\alpha + 1) : (\alpha - 1)$. For easy of presentation, let $y'' - y$ be a multiple of $2(\alpha + 1)$. Then we perform the following communication: $(i = 1 \ldots \frac{\alpha-1}{2}$ and $j = 1 \ldots \frac{\alpha+1}{2})$.

- $p_{x,y,z}$ sends $\frac{\alpha+1}{2}$ messages to nodes $p_{x,y+2it,z}$,
- $p_{x',y',z'}$ sends $\frac{\alpha+1}{2}$ messages to nodes $p_{x',y'-2it,z'}$ and $\frac{\alpha-1}{2}$ messages to nodes $p_{x',y'+2jt,z'}$,
- $p_{x'',y'',z''}$ sends $\frac{\alpha-1}{2}$ messages to nodes $p_{x'',y''-2jt,z''}$.

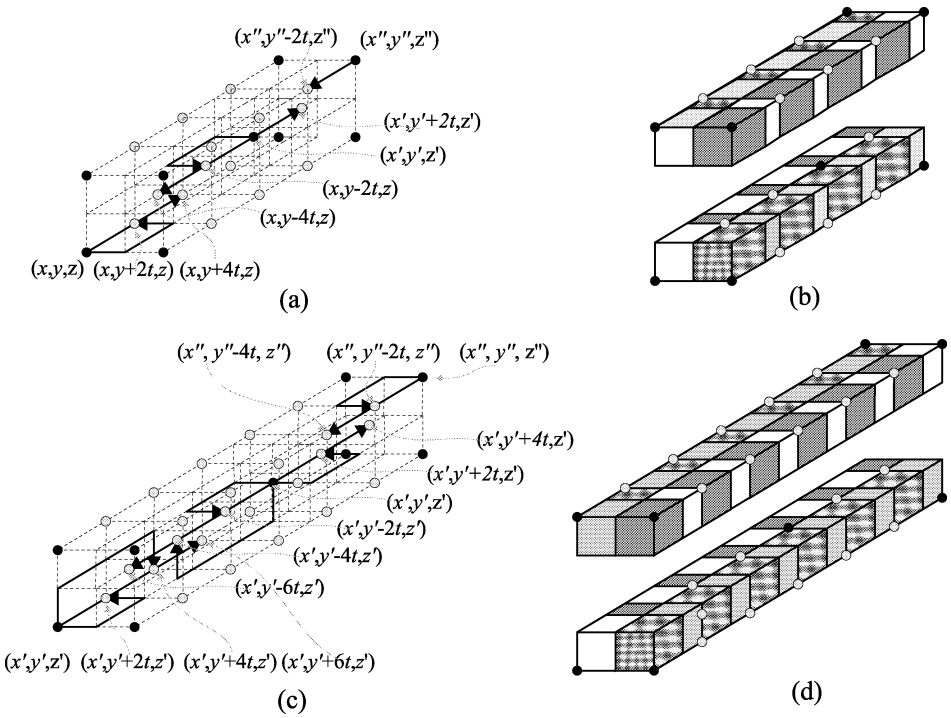

Figure 15. One communication step of Stage 3 in eight neighboring unit cubes: (a) the communication pattern of 5-port tori and (b) each unit cube is partitioned into six smaller cubes after one step; (c) the communication pattern of 3-port tori and (d) each unit cube is partitioned into four smaller cubes after one step.

For instance, assuming $\alpha = 3$, Figure 15(a) shows the communication paths. Apparently, after this step, each unit cube will be partitioned into a number of smaller unit cubes. In this example, each unit cube will generate three or five smaller unit cubes. In Figure 15(b), it is shown that totally 32 smaller cubes are generated from the eight larger cubes.

As another example, Figures 15(c) and (d) show the case of $\alpha = 5$. In general, the number of unit cubes will be multiplied by a factor of $\alpha + 1$ after each communication step. In order to regularly reduce the size of unit cubes, the recursion should maintain an important invariant:

I2. For any two consecutive unit cubes of type I, the ratio of the width of the larger cube to that of the smaller one should be approximately $(\alpha + 1) : (\alpha - 1)$.

To prove that our approach follows this invariant, one can first verify Definition 7 and then the communication pattern given above.

The recursion is repeated until $y'' - y < 2(\alpha + 1)$. This stage takes $\lceil \log_{(\alpha+1)} \frac{2n_2}{n_1} \rceil - 1$ steps to complete.

Stage 4. This Stage is similar to Stage 3, except that we want to "expand" the nodes owning M along the Z axis. The derivation is similar to Stage 3 and we omit the details.

Stage 5. Now each unit cube of type I will be of width at most $\alpha + 1$ and of height at most $\alpha + 1$. The approach is similar to the Stage 5 of Section 4.2.1, so we omit the details. This stage takes three steps.

Theorem 9. *In a circuit-switched α-port 3D torus $T_{n_1 \times n_2 \times n_3}$ such that α is odd, broadcast can be done in*

$$3\left\lceil \log_{\alpha+1} \frac{n_1}{2} \right\rceil + \left\lceil \log_{\alpha+1} \frac{2n_2}{n_1} \right\rceil + \left\lceil \log_{\alpha+1} \frac{2n_3}{n_1} \right\rceil + c$$

steps, where $c = 2$ (resp., 3) when n_1 is even (resp. odd), which number of steps is at most six (resp., 7) steps more than the lower bound in Lemma 1.

5. Extensions to higher-dimensional tori

In this section, we briefly show how to extend the result to a kD torus $T_{n_1 \times n_2 \times \cdots \times n_k}$. If the torus is square, then a dimension-by-dimension approach can be used to distribute M. Otherwise, assuming the first dimension to be the one of smallest length, we can first "squeeze" the torus into a square, dilated one $T_{n_1 \times \cdots \times n_1}$. Then we try to reduce the dilation along the 2nd, 3rd, ..., kth dimensions one by one. If the number of ports α is even, then we can simply partition the torus into subnetworks of even size. Otherwise, an invariant similar to **I2** should be developed so that the recursion can proceed.

6. Conclusions

In this paper, we have presented a systematic solution to solve the broadcasting problem in an α-port torus of any dimension and any size with circuit switching. The problem has posed a great challenge because a good solution should try to utilize as many of the available ports as possible. Further, when the torus is non-square, we should try to distribute the broadcast message to nodes in the network as evenly as possible to avoid congestion in the subsequent communications. The "dimension-by-dimension" and "squeeze-then-expand" approaches proposed in this paper have successfully conquered these difficulties and have delivered performance very close to the lower bound of this problem.

References

1. CM-5 technical summary. Thanking Machines Corp., 1991.
2. V. Bala, J. Bruck, R. Cypher, P. Elustondo, A. Ho, C. T. Ho, S. Kipmis, and Snir. CCL: A portable and tunable collective communication library for scalable parallel computers. *In International Parallel Processing Symposium*, Cancun, Mexico, pp. 835–843, April 1994.

3. C.-T. Ho and M.-Y. Kao. Optimal broadcasting in all-port wormhole-routed hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 6(2):200–204, 1995.
4. S.-K. Lee and J.-Y. Lee. Optimal broadcast in α-port wormhole-routed mesh networks. In *International Conference on Parallel and Distributed Systems*, pp. 109–114, 1997.
5. P. K. McKinley, Y.-J. Tsai, and D. F. Robinson. Collective communication in wormhole-routed massively parallel computers. *IEEE Computers*, 28(12):39–50, 1995.
6. Message Passing Interface Forum. Document for standard message-passing interface, November 1993.
7. W. K. Nicholson. *Linear Algebra with Applications*, 3rd ed. PWS Publishing Company, 1995.
8. S. F. Nugent. The iPSC/2 direct-connect technology. In *Proceedings of 3rd ACM Conference on Hypercube Concurrent Computers and Applications*, pp. 51–60, 1988.
9. J. L. Park and H.-A. Choi. Circuit-switched broadcasting in tori and meshes networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(2):184–190, 1996.
10. J. L. Park, S.-K. Lee, and H.-A. Choi. Circuit-switched broadcasting in d-dimensional torus and mesh networks. In *International Parallel Processing Symposium*, pp. 26–29, 1994.
11. J. G. Peters and M. Syska. Circuit-switched broadcasting in torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(3):246–255, 1996.
12. R. Ponnusamy, A. Choudhary, and G. Fox. Communication overhead on CM5: an experimental performance evaluation. In *Symposium on Frontiers of Massively Parallel Computation*, pp. 108–115, 1992.
13. D. F. Robinson, P. K. Mckinley, and B. H. C. Cheng. Optimal multicast communication in wormhole-routed torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 6(10):1029–1042, 1995.
14. Y.-J. Tsai and P. K. McKinley. A broadcasting algorithm for all-port wormhole-routed torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(8):876–885, 1996.
15. Y.-C. Tseng. A dilated-diagonal-based scheme for broadcast in a wormhole-routed 2d torus. *IEEE Transactions on Computing*, 46:947–952, 1997.
16. Y.-C. Tseng, S.-Y, Ni, and J.-P. Sheu. Toward optimal complete exchange on wormhole-routed tori. *IEEE Transactions on Computing*, 48(10):1065–1082, 1999.
17. C.-M. Wang and C.-Y. Ku. A near-optimal broadcasting algorithm in all-port wormhole-routed hypercubes. In *ACM International Conference on Supercomputing*, pp. 147–153, 1995.
18. S.-Y. Wang and Y.-C. Tseng. Algebraic foundations and broadcasting algorithms for wormhole-routed all-port tori. *IEEE Transactions on Computing*, 49(3):246–258, 2000.