Design and Analysis of Pipelined Broadcast Algorithms for the All-Port Interlaced Bypass Torus Networks

Peng Zhang and Yuefan Deng, *Member*, *IEEE*

Abstract—Broadcast algorithms for the interlaced bypass torus networks (iBT networks) are introduced to balance the all-port bandwidth efficiency and to avoid congestion in multidimensional cases. With these algorithms, we numerically analyze the dependencies of the broadcast efficiencies on various packet-sending patterns, bypass schemes, network sizes, and dimensionalities and then strategically tune up the configurations for minimizing the broadcast steps. Leveraging on such analysis, we compare the performance of networks with one million nodes between two cases: one with an added fixed-length bypass links and the other with an added torus dimension. A case study of $\mathrm{iBT}(1000^2; \boldsymbol{b} = \langle 8, 32 \rangle)$ and $\mathrm{Torus}(100^3)$ shows that the former improves the diameter, average node-to-node distance, rectangular and global broadcasts over the latter by approximately 80 percent. It is reaffirmed that strategically interlacing short bypass links and methodically utilizing these links is superior to adding dimensionalities to torus in achieving shorter diameter, average node-to-node distances and faster broadcasts.

Index Terms—Parallel computing, collective communication, broadcast, network performance, interlaced bypass torus

1 Introduction

Massive parallel processing systems [1], [2] continue to improve and are capable of incorporating millions of powerful processor cores [3], [4], [5], [6], [7] and of supporting simultaneous communications over multiple links at the hardware level [4], [5], [6], [8]. For accommodating such rapid expansion of systems, the advanced cellular networks like a 3D torus network such as those in the IBM's Blue Gene [4], [5], [6] and Cray's Red Storm [3], [7], [9] are proposed, analyzed [10], [11], and manufactured. However, the progress is too slow to satisfy the increasing demands of large-scale scientific computations for exploring the unknown world [12]. This stimulates active research in new scalable interconnection networks.

An interconnection network is scalable only if it, for a given node degree, has a small network diameter and small average node-to-node distance, high bandwidth and simple topology. The iBT network introduced by interlacing bypass rings to torus satisfies most of the above conditions and has improved the performance of the original torus network [13] from which the iBT network outgrows.

In addition to these node-to-node communication gains, the collective communications [8], [10], [14], [15], [16], [17], [18], [19] are also critical for the sustained performance and scalability of such applications as matrix multiplication, LU-factorization, Householder transformation and Fast Fourier Transform (FFT) [20], [21], [22], [23], [24], [25]. Collective communication algorithms have been widely studied on clusters [17], [26], [27], [28], [29], meshes and torus networks [8], [19], [30], [31], [32], [33], [34], [35], [36], [37], [38] recently. Some of them are automatically tuned [26], [39] and others are machine-optimized [5], [15], [19], [31], [37], [40]. Particularly, the new parallel architectures like the IBM Blue Gene with the nodal all-port ability, i.e., a node can simultaneously communicate with multiple neighbors, trigger the redesign of the many MPI collective algorithms for exploiting the architectural merits [8], [15], [30], [38]. For example, the optimized MPI collective operations improve the high-performance LINPACK (HPL) [22] efficiency by 6.2 percent over a default implementation [19] and achieved more effective bandwidth in general messaging [15].

Collective operations include broadcast, scatter, reduce, gather, and barrier. This paper focused on studying the one-to-n (n can be many or all nodes) broadcast in which a node sends the same data to all other n nodes. Broadcast is a basic collective operation that lays the foundation for many other collective algorithms. For example, it can be enhanced to form the scatter operation in which a node scatters its data to n other nodes. The gather operation, in which a node gathers data from a group of nodes, is the reverse operation of the scatter. The reduce-to-one, in which data on all nodes are reduced to data on a single node, is the reverse operation of the one-to-many broadcast. Many important global operations are accelerated at the hardware level. Those operations include Blue Gene's barrier interface in the global interrupt network, and the reduce and allreduce executors in the collective tree network [41]. Broadcast is extensively utilized in many applications, for example, in solving linear equations in LINPACK [22].

In this paper, we assume:

1. An iBT network that supports simultaneous communication over all links [8], [19], [30], [38].

• *The authors are with the Department of Applied Mathematics, Stony Brook University, Stony Brook, NY 11794-3600.*
E-mail: {Peng.Zhang, Yuefan.Deng}@StonyBrook.edu.

Manuscript received 14 May 2011; revised 5 Oct. 2011; accepted 29 Feb. 2012; published online 9 Mar. 2012.
Recommended for acceptance by D. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-05-0311.
Digital Object Identifier no. 10.1109/TPDS.2012.93.

2. A pipelined k-packet message broadcast, i.e., a broadcast message contains a sequence of k packets [32], [33], [36], [42], [43], [44] that are ordered as $pkt(1), \ldots, pkt(k)$. These packets are individually pipelined to the network to multiple destinations.
3. The time for a packet to travel from a node to its nearest neighbor is defined as one broadcast step. As soon as all packets arrive, a complete message is assembled.
4. A broadcast time is measured as the total number of broadcast steps required to populate all nodes with the original message.

Our paper considers the design, analysis, and optimization of the broadcast algorithms and it is organized as follows: we review the definition of the iBT network in Section 2. In Section 3, we present the notations and performance metrics. In Section 4, we present a line broadcast algorithm for an iBT network and through substantial experiments, we collect and quantify the dependencies of the line broadcast efficiency on the packet-sending patterns, network sizes, dimensionalities, and bypass schemes. Also, we characterize this line broadcast algorithm in terms of these impacts, while comparing with torus networks. Accordingly, in Sections 5 and 6, we present and analyze the rectangular and global broadcast algorithms for iBT networks. Section 7 presents the discussions and conclusions.

2 Interlaced Bypass Torus

The iBT network [13] is constructed by interlacing bypass rings to torus A general d-dimensional $\mathrm{iBT}(n^d; \boldsymbol{b} = \langle b_1, \ldots, b_k \rangle)$ network starts from a d-dimensional torus network $\mathrm{Torus}(n^d)$. The bypass vector $\boldsymbol{b} = \langle b_1, \ldots, b_k \rangle$ indicates the scheme of the added bypass rings, i.e., interlacing b_i-hop bypass rings $(i = 1, \ldots, k)$ recursively into the d dimensions to generate a new network. This new network has a node degree of $2d + 2$ where $2d$ are from the original torus connections while the additional 2 are from the bypass links.

3 Notations and Performance Metrics

3.1 Notations

This paper discusses two network families: $\mathrm{iBT}(n^d; \boldsymbol{b})$ and $\mathrm{Torus}(n^d)$ networks. Each node in these networks is related to a d-tuple coordinate $\vec{x} = (x_1, \ldots, x_d)$ in which $x_i \in [0, n-1]$ and $i \in [1, d]$. In the case of a d-dimensional network, an integer set Ω based on the coordinates of two nodes $\vec{x}$ and $\vec{y}$ is defined as: $\Omega(\vec{x}, \vec{y}) = \{i | x_i = y_i\}$. The universe is $U(d) = [1, d]$ and thus $\Omega(\vec{x}, \vec{y}) \subseteq U(d)$. $\Omega^c(\vec{x}, \vec{y})$ refers to the complement of $\Omega(\vec{x}, \vec{y})$ with respect to $U(d)$ and is defined as $\Omega^c(\vec{x}, \vec{y}) = U(d) \backslash \Omega(\vec{x}, \vec{y}) = \{i | x_i \neq y_i\}$.

An integer set $\chi(i, k)$ is defined as

$$\chi(i,k) = \{ j | j = (i + l)_d + 1, \forall l \in [-1, k] \} \subseteq U(d),$$

where $(x)_d = x (\mathrm{mod}\ d)$. Then $|\chi(i,k)| = k + 2 \in U(d)$, where $i \in U(d)$ and $k \in [-1, d-2]$. Particularly, the largest and smallest sets are $\chi(i, d-2) = U(d)$ and $\chi(i, -1) = \{i\}$. $\chi^c(i,k)$ refers to the complement of $\chi(i,k)$ with respect to $U(d)$.

Property. *If $i \in \Omega^c(\vec{x}, \vec{y})$, then there exists an integer $k \in [-1, d-2]$ such that*

$$\Omega^c(\vec{x}, \vec{y}) \subseteq \chi(i,k). \tag{1}$$

Here, $i \in U(d)$. Particularly for a given i, the minimum integer k_m that satisfies (1) is defined as

$$k_m(\vec{x}, \vec{y}) = \min\{k | \Omega^c(\vec{x}, \vec{y}) \subseteq \chi(i,k)\}.$$

In other words, $\chi(i, k_m(\vec{x}, \vec{y}))$ is the minimum superset of $\Omega^c(\vec{x}, \vec{y})$. $\chi^c(i, k_m(\vec{x}, \vec{y}))$ refers to the complement of $\chi(i, k_m(\vec{x}, \vec{y}))$ with respect to $U(d)$.

For example, $\vec{x} = (1, 3, 4)$ and $\vec{y} = (4, 3, 2)$ are two points in $iBT(12^3, \boldsymbol{b} = 3)$ so we have $\Omega(\vec{x}, \vec{y}) = \{2\}$, $\Omega^c(\vec{x}, \vec{y}) = [1, 3] \backslash \{2\} = \{1, 3\}$ and all of the possible $\chi(i,k)$ are

i \ k	-1	0	1
1	{1}	{1,2}	{1,2,3}
2	{2}	{2,3}	{1,2,3}
3	{3}	{1,3}	{1,2,3}

Thus, we see that $\chi(3, 0)$ is the minimum superset of $\Omega^c(\vec{x}, \vec{y})$ and thus $k_m(\vec{x}, \vec{y}) = 0$ and $\chi^c(3, 0) = \{2\}$.

3.2 Performance Metrics

A performance metric of a network, f includes the diameter (λ), average node-to-node distance (μ), line (T_x), rectangular (T_{xy}), or one-to-all (T_{all}) broadcast time.

To compare performance metrics between iBT and torus networks of the same network dimensions or node degree, we define two coefficients as follows.

$\alpha(f; n, d, \boldsymbol{b})$ is a real number defined as the ratio of a metric f of $\mathrm{iBT}(n^d; \boldsymbol{b})$ over that of $\mathrm{Torus}(n^d)$

$$\alpha(f; n, d, \boldsymbol{b}) = 1 - \frac{f(\mathrm{iBT}(n^d; \boldsymbol{b}))}{f(\mathrm{Torus}(n^d))} < 1, \tag{2}$$

$\beta(f; n, d, \boldsymbol{b})$ is a real number defined as the ratio of a metric f of $\mathrm{iBT}(n^d; \boldsymbol{b})$ over that of $\mathrm{Torus}(m^{d+1})$:

$$\beta(f; n, d, \boldsymbol{b}) = 1 - \frac{f(\mathrm{iBT}(n^d; \boldsymbol{b}))}{f(\mathrm{Torus}(m^{d+1}))} < 1. \tag{3}$$

Here, $m = n^{d/(d+1)}$ in (3).

Actually, $\alpha(f; n, d, \boldsymbol{b})$ or $\beta(f; n, d, \boldsymbol{b})$ shows the percentage performance improvement of an iBT network over a torus that has both the same network size N and the same dimensions n^d or the same node degree $2(d+1)$ in terms of f. A table of notations and their definitions and examples is given in Appendix 1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2012.93.

4 Line Broadcast

In Sections 4 to 6, we design and analyze the broadcast algorithms involving one-to-many and one-to-all operations in iBT networks. Broadcast in this context involves sending one identical message from a node (the source) to all other nodes in a given broadcast group. Upon completion, all nodes in the group own a copy of this message.

Depending on the topology on which we broadcast, we group these broadcast algorithms into three categories: the line, rectangular, and global algorithms. A line or rectangular broadcast algorithm handles a broadcast group in a line or a rectangle while a global broadcast algorithm handles all nodes in the network. The design and analysis depends recursively on those of the lower dimensional ones.

Furthermore, depending on the message size, we group the broadcast algorithms into those for single- and multiple-packet broadcast. Broadcasting a single-packet message allows utilizing all possible optimal paths to route the message to the farthest nodes regardless of any packet contention or network congestion. On the other hand, broadcasting a multiple-packet message requires a bandwidth-efficient algorithm capable of balancing the link utilization, avoiding packet contention, alleviating the network congestion and thus maximizing the messaging bandwidth. The time required to broadcast a short message depends solely on the network diameter while the time to broadcast a long message depends both on the network architecture and the broadcast algorithm. Our algorithms are designed for multiple-packet or k-packet messages where $k \geq 2$.

Meanwhile, the message size that can be enqueued for simultaneous communication over all links is also constrained by buffer of the participating routers [4], [5], [6]. Blue Gene/L router chip's processor interface consists of 8 injection and 14 reception queues for a total of 58 KB of SRAMs and a full-sized packet size is 256 Bytes. On average, the SRAMs among these queues can allow 10 full-sized packets per queue, or six normal-priority injection queues enable sixty packets enqueued per processor concurrently. Thus, a moderate message size $k = 32$ was chosen for our experiments.

4.1 Algorithms

Broadcasting in n-ary d-cube networks, i.e., $\mathrm{Torus}(n^d)$, is the most similar case to that in iBT networks. Generally, the broadcast time for a k-packet message along a 1D $\mathrm{Torus}(n^d)$ is

$$T_x(\mathrm{Torus}(n^d)) = n/2 + \lceil k/2 \rceil - 1. \quad (4)$$

A node in a ring has two opposing links so an obviously optimal algorithm is to initiate the packets in the two reversing orders along two opposing directions from the source node while all other nodes forward a packet to its nearest neighbors. This is illustrated in Fig. 1a in the Appendix 2, which is available in the online supplemental material. Similarly, we construct a line broadcast algorithm for the iBT networks. Starting with an assumption: a message is to broadcast from node $\vec{s}$ in the ith dimension, we have the procedures of a line broadcast.

1. $\vec{s}$ sends a sequence of packets to each of its outgoing links in the ith dimension in a specific order.
2. A node $\vec{x}$ that satisfies $\Omega^e(\vec{x}, \vec{s}) = \{i\}$ forward a newly received packet to its nearest ith-dimensional neighbors that has not received it yet.

Sending packets in order means: a link first sends $pkt(1)$, then $pkt(2)$, and so on, until $pkt(k)$. On the other hand, sending packets in reverse order means: a link first sends $pkt(k)$, then $pkt(k-1)$, and so on, until $pkt(1)$.

However, a node of a 1D iBT network has at most four links so the optimal decision on selecting packet-sending orders for these links is not straightforward. We must analyze the performance dependencies on the packet-sending patterns, network sizes, and dimensionalities (Section 4.2), and on T_x (Section 4.3).

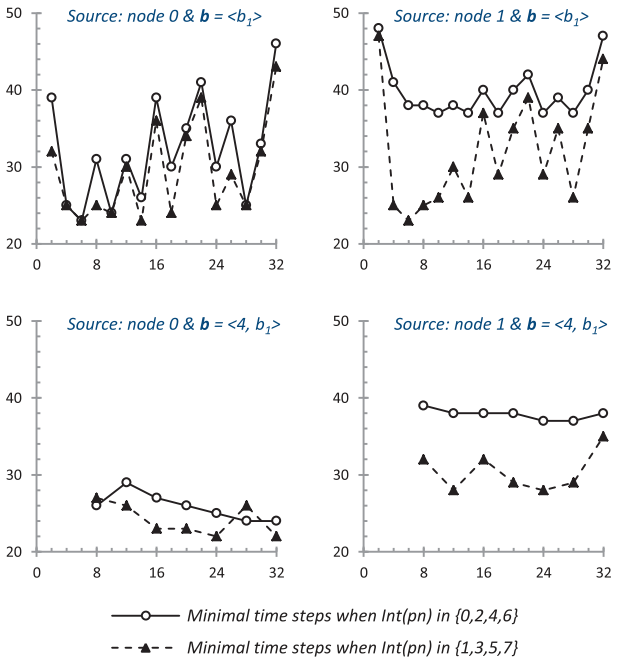

Fig. 1. $T_x(\mathrm{iBT}(64^2; \boldsymbol{b}), s)$ versus b_1.

4.2 Performance Dependencies

Packet-sending patterns. The order of sending packets through a link is referred to as a link packet-sending order and it is either in order or in reverse order, whose numerical representations are digit 0 or 1, respectively.

To identify all of the link packet-sending orders, we assign the source node of a line broadcast with an integer $Int(pn) \in [0, 7]$. Its 4-binary representation, $Bin(pn) = rRLl$, indicates a corresponding link packet-sending order. Of the representation, r and l indicate the packet-sending orders of the right and left torus links while R and L for the bypass links. For example, $Int(pn) = 6 = Bin(pn) = 0110$ indicates sending packets in order through right and left torus links while in reverse order through right and left bypass links. This is illustrated in Fig. 1b of Appendix 2, which is available in the online supplemental material.

Fig. 1 shows the numerical analysis of the dependencies of $T_x(\mathrm{iBT}(64^2; \boldsymbol{b}), s)$ on various values of $Int(pn) \in [0, 7]$. $\boldsymbol{b}$ is either uniform $\langle b_1 \rangle$ or hybrid $\langle 4, b_1 \rangle$ thus b_1 indicating the longest bypass length. The message is of 32 packets long hereinafter, unless otherwise stated.

Table 1 compares selected $iBT(64^2; \boldsymbol{b})$ in terms of λ, μ, and T_x. Here, $T_x(\mathrm{iBT}(64^2; \mathbf{b}), \mathrm{s})$ are the minima for any $Int(pn) \in [0, 7]$ as shown in Fig. 1.

Network sizes. While increasing network sizes N from 2K to 65K, Fig. 2 presents $T_x(\mathrm{iBT}(n^2; \boldsymbol{b}), s)$ for selected bypass schemes $\boldsymbol{b} \in \{\langle b_1 \rangle, \langle 8, b_1 \rangle\}$ where $b_1 \in \{8, 16, 24\}$.

TABLE 1
Performance Comparisons of Selected iBT $(64^2; \boldsymbol{b})$

$\boldsymbol{b}$ \ Perf.	λ	μ	T_x		T_{xy}	
			$s=0$	$s=1$	$s=0$	$s=1$
⟨6⟩	14	8.096	23	23	28	28
⟨14⟩	12	7.482	24	23	27	27
⟨4,16⟩	10	6.568	23	32	27	27
⟨8,24⟩	10	6.633	22	22	25	26

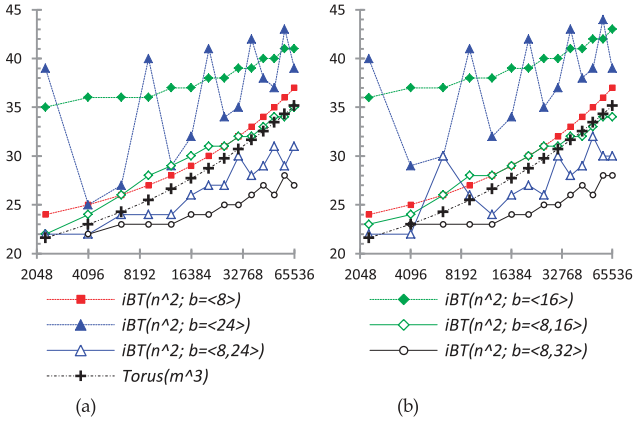

Fig. 2. (a) $T_x(\mathrm{iBT}(n^2; \boldsymbol{b}), 0)$ versus N. (b) $T_x(\mathrm{iBT}(n^2; \boldsymbol{b}), 1)$ versus N.

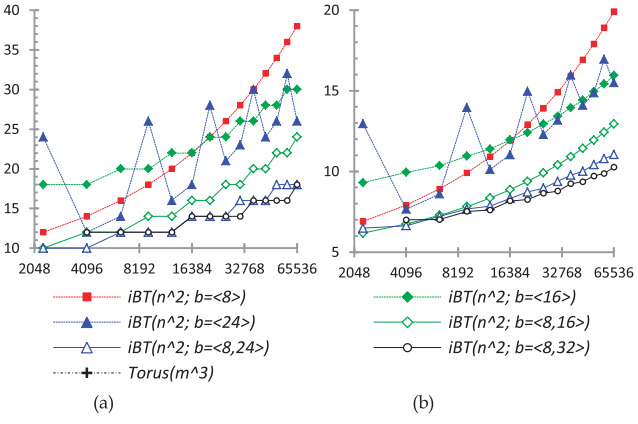

Fig. 3. (a) $\lambda(\mathrm{iBT}(n^2; \boldsymbol{b}))$ versus N. (b) $\mu(\mathrm{iBT}(n^2; \boldsymbol{b}))$ versus N.

Figs. 2a and 2b show the cases for $s = 0$ and $s = 1$, respectively. Here, $Int(pn) \in \{1, 3, 5, 7\}$.

Additionally, Figs. 3a and 3b present the diameters λ and average distances μ of these iBT networks.

By fitting the data in Figs. 2 and 3, we obtain T_x, λ, and μ as functions of network sizes N for 2D iBT with $\boldsymbol{b} \in \{\langle 8\rangle, \langle 8, 32\rangle\}$ as shown in Table 2.

Dimensionalities. While increasing the dimension d from 2 to 4, we present $T_x(\mathrm{iBT}(n^d; \boldsymbol{b}), 0)$ and $T_x(\mathrm{Torus}(m^{d+1}))$ as functions of the same network size in Fig. 4. One uniform bypass schemes $2d$ and one hybrid scheme $2d, 4d$ are selected. Here, $Int(pn) \in \{1, 3, 5, 7\}$.

By fitting the data in Fig. 4, we obtain T_x as functions of network sizes N for $\mathrm{iBT}(n^3; \boldsymbol{b} = \langle 6\rangle)$ and $\mathrm{iBT}(n^4; \boldsymbol{b} = \langle 8\rangle)$ as shown, also, in Table 2.

4.3 Performance Analysis

We compare the line broadcast performance between an iBT network with a fixed bypass scheme and a torus network, for a set of networks with increasing sizes. The torus network has the same network dimensions or the same node degree as the iBT network. Examining the functions in Table 2, we introduce $\alpha(f; n, d, \boldsymbol{b})$ and $\beta(f; n, d, \boldsymbol{b})$ for $f \in \{\lambda, \mu\}$ in Figs. 5 and 6, and for $f = T_x$ in Figs. 7 and 8. Here, $d \in \{2, 3, 4\}$ and $\boldsymbol{b} \in \{\langle 8, 32\rangle, \langle 6\rangle, \langle 8\rangle\}$.

Analysis of experiments in Section 4.2 leads to the following.

The line broadcast performance of iBT networks is highly sensitive to the packet-sending pattern as shown in Fig. 1 and cases $Int(pn) \in \{1, 3, 5, 7\}$ are more favorable. Fig. A of Appendix 3, which is available in the online supplemental material, also illustrates the link utilization of these packet-sending patterns.

The line broadcast performance of iBT networks also depends on a node bypass species, i.e., a node always

TABLE 2
Fitting Functions of T_x, λ, and μ for iBT Networks

Networks \ Functions	T_x	λ	μ
$\mathrm{iBT}(n^2; \boldsymbol{b} = \langle 8\rangle)$	$0.0625 \cdot N^{0.5} + 21$	$0.63 \cdot N^{0.37}$	$0.45 \cdot N^{0.34}$
$\mathrm{iBT}(n^2; \boldsymbol{b} = \langle 8,32\rangle)$	$2.15 \cdot N^{0.17} + 12.51$	$3.02 \cdot N^{0.15}$	$1.92 \cdot N^{0.15}$
$\mathrm{iBT}(n^3; \boldsymbol{b} = \langle 6\rangle)$	$2.79 \cdot N^{0.16}$	$0.46 \cdot N^{0.30}$	$0.30 \cdot N^{0.29}$
$\mathrm{iBT}(n^4; \boldsymbol{b} = \langle 8\rangle)$	$4.01 \cdot N^{0.1}$	$3.04 \cdot N^{0.14}$	$1.77 \cdot N^{0.13}$

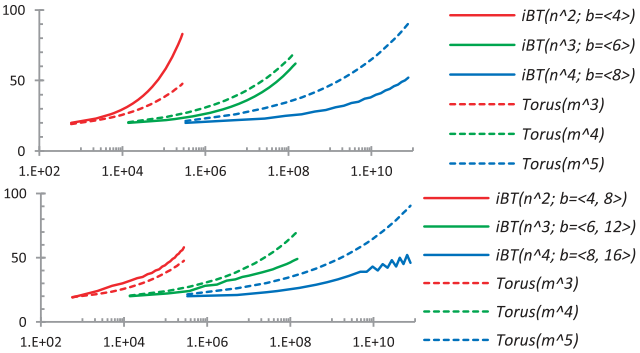

Fig. 4. $T_x(\mathrm{iBT}(n^d; \boldsymbol{b}), 0)$ and $T_x(\mathrm{Torus}(m^{d+1}))$ versus N.

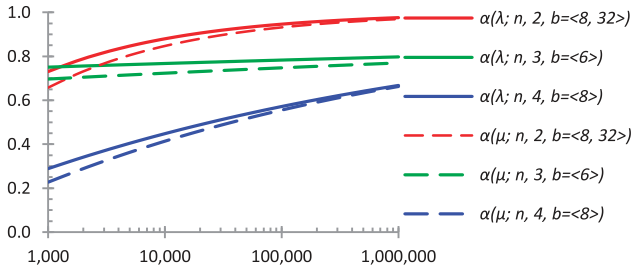

Fig. 5. Performance improvement (α) versus network size N in terms of λ and μ.

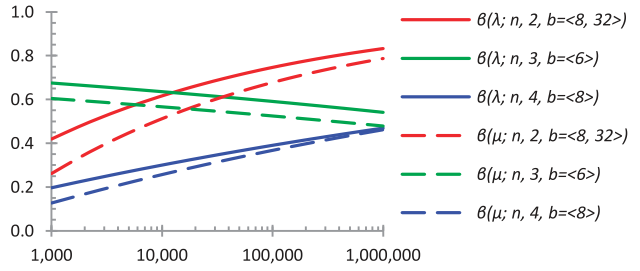

Fig. 6. Performance improvement (β) versus network size N in terms of λ and μ.

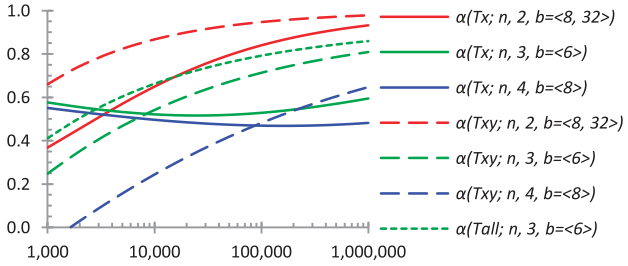

Fig. 7. Performance improvement (α) versus N in terms of T_x and T_{xy} $(k = 32)$.

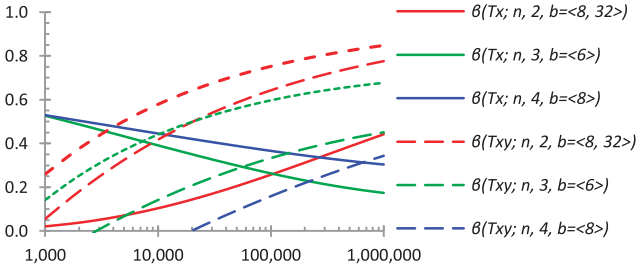

Fig. 8. Performance improvement (β) versus N in terms of T_x and T_{xy} $(k = 32)$.

broadcasts faster along its bypass dimension. As shown by Fig. 1, node 0 finishes line broadcast sooner than node 1 because its bypass dimension happens to be the broadcast dimension. However, this advantage diminishes as the network becomes larger. For example, in Fig. 2, as the network size increases from 2K to 65K, the differences between $T_x(\text{iBT}(n^2, \boldsymbol{b}), 0)$ in Fig. 2a and $T_x(\text{iBT}(n^2, \boldsymbol{b}), 1)$ in Fig. 2b become negligible.

Besides, the diameter λ and average distance μ, bypass schemes per se also determines the line broadcast performance. As concluded in [13], an iBT network with extreme bypass lengths, too short or too long, has larger diameter and larger average distance than that with middle-sized bypass lengths. It is the same for broadcast as shown in Fig. 1. Zhang et al. [13] also found that an iBT network with diverse bypass lengths always leads to a shorter diameter and average distance. This, however, does not guarantee a shorter line broadcast time. For example, Table 1 shows that $\text{iBT}(64^2; \boldsymbol{b} = \langle 6 \rangle)$ with larger λ and μ broadcasts faster than $\text{iBT}(64^2; \boldsymbol{b} = \langle 4, 16 \rangle)$. Fig. C of Appendix 3, which is available in the online supplemental material, also illustrates: $\boldsymbol{b} = \langle 6 \rangle$ boosts the link utilization faster and higher than $\boldsymbol{b} = \langle 4, 16 \rangle$ does. This very fact necessitates optimization for broadcast over iBT networks.

Moreover, for larger networks, we observe counter-intuitive variations in λ and μ resulting from an improper bypass scheme. Such behavior appears obvious for $\lambda(\text{iBT}(n^2; \boldsymbol{b} = \langle 24 \rangle))$ and $\mu(\text{iBT}(n^2; \boldsymbol{b} = \langle 24 \rangle))$ in Fig. 3 as well as in $T_x(\text{iBT}(n^2; \boldsymbol{b} = \langle 24 \rangle))$ in Fig. 2. Thus, an iBT network with $\boldsymbol{b} = \langle 24 \rangle$ is inferior in terms of scalability over an iBT network with other bypass schemes.

We also noticed that the inferior performance in λ and μ is propaged to the inferior broadcast performance as illustrated in Fig. B of Appendix 3, which is available in the online supplemental material.

These facts help guide the search for the optimal bypass scheme for given communication patterns. For example, while performing Fast Fourier Transform in $\text{iBT}(64^2; \boldsymbol{b})$, $\boldsymbol{b} = \langle 6 \rangle$, a favorable bypass scheme for broadcast, is sufficient for FFT which requires much more long-ranged (broadcasts) than short-ranged communications. Our second example is the high-performance LINPACK benchmark that uses the recursive panel factorization with the pivot search and column broadcast combined. The pivot search has fewer local data exchange than column broadcast. Thus, local data exchange requires a network with a short diameter to minimize latency while the column broadcast requires maximal bandwidth capability. To meet both demands, $\text{iBT}(64^2; \boldsymbol{b} = \langle 8, 24 \rangle)$ appears to be the best option as shown in Table 1.

In summary, as the network size increases to a million nodes with dimensionality from 2 to 4, the iBT networks with fixed bypass schemes always outperform the torus networks of the same dimensions measured in λ and μ and T_x as shown in Figs. 5, 6, and 7.

Interlacing bypass links is more efficient than adding a dimension to torus for improving the network performance for most applications. For a one-million-node system iBT can improve λ and μ by 40 percent in Fig. 6. The iBT network with a fixed bypass scheme performs line broadcasts faster than a torus of the same node degree though it has more nodes in the broadcast group. This advantage intensifies as the network increases, for 2D case.

5 Rectangular Broadcast

We study the rectangular broadcast algorithm for the iBT network and examine the performance dependencies on bypass schemes, network sizes, and dimensionalities.

5.1 Algorithms

For a rectangular broadcast in an iBT network, we combine the multi-color spanning tree algorithm for a torus [19] with the line broadcast algorithm described and optimized in Section 4. The multicolor spanning tree algorithm is effective in avoiding congestions while the line broadcast is optimizable for the iBT networks. For example, in the $2d$ link-disjoint spanning trees in $\text{Torus}(n^d)$ $(d \geq 2)$, a line broadcast algorithm can be applied recursively in each of these trees [19]. The broadcast time of a k-packet message evenly over these $2d$ separate spanning trees is

$$T_{all}(\text{Torus}(n^d)) = d(n-1) + \lceil k/2d \rceil - 1. \tag{5}$$

The first packet spends $d(n-1)$ steps to reach the farthest node and the remaining packets spend $(k/2d - 1)$ steps to pipeline into that final node.

Particularly, the broadcast time over $\text{Torus}(n^d)$ is

$$T_{xy}(\text{Torus}(n^d)) = 2n + \lceil k/4 \rceil - 3. \tag{6}$$

Similarly, we decompose the links in $\text{iBT}(n^d)$ into d, instead of $2d$, link-disjoint groups and recursively apply this line broadcast algorithm along every dimension of the groups. An example of a two-color scheme is illustrated in Fig. 10 of Appendix 4, which is available in the online supplemental material.

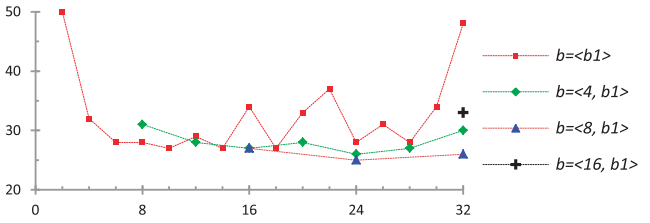

Fig. 9. $T_{xy}(\text{iBT}(64^2;\boldsymbol{b}),0)$ versus b_1.

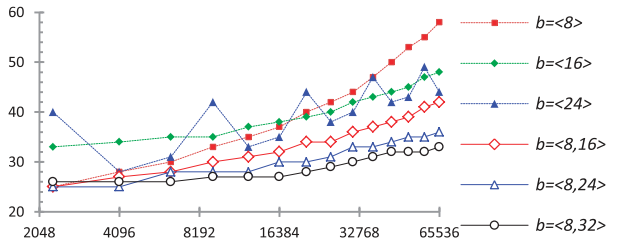

Fig. 10. $T_{xy}(\text{iBT}(n^2;\boldsymbol{b}),0)$ versus N.

Accordingly, a two-color rectangular broadcast is applied on this iBT network. A k-packet broadcast message is decomposed into these two groups of packets: k_1 and k_2 packets to broadcast simultaneously in the 1-color and 2-color groups: $k = k_1 + k_2$. These k_1 packets to broadcast in the 1-color groups are referred to as 1-color packets; likewise, the k_2 packets as 2-color packets.

We construct a rectangular broadcast algorithm for iBT networks starting with the assumption: a message is to broadcast from node $\vec{s}$ in the i_1^{th} and i_2^{th} dimensions where $i_1, i_2 \in U(d)$ and $i_1 \neq i_2$. Then, we have the procedures of a rectangular broadcast algorithm.

1. $\vec{s}$ initiates two independent line broadcasts in parallel: broadcast k_1 1-color (or k_2 2-color) packets in the i_1^{th} (or i_2^{th}) dimension.
2. A node $\vec{x}$ that satisfies $\Omega^c(\vec{x},\vec{s}) = \{i_1\}$ broadcasts the i_1-color packets in the i_1^{th} dimension; likewise, a node $\vec{x}$ that satisfies $\Omega^c(\vec{x},\vec{s}) = \{i_2\}$ broadcasts the i_2-color packets in the i_2^{th} dimension.
3. A node $\vec{x}$ that satisfies $\Omega^c(\vec{x},\vec{s}) = \{i_1, i_2\}$ forward a newly received packet to its neighbors in the dimension the packet comes from.
4. A node $\vec{x}$ that satisfies $\Omega^c(\vec{x},\vec{s}) = \{i_1, i_2\}$ forward an i_1-color or i_2-color packet to its nearest neighbor $\vec{y}$ if $\Omega^c(\vec{y},\vec{s}) = \{i_2\}$ or $\Omega^c(\vec{y},\vec{s}) = \{i_1\}$.

In the first step, a source node always selects a favorable pn for each of these two line broadcasts.

As observed in Section 4.3, a node always broadcasts faster in its bypass dimension than in any other dimension so we adjust k_1 and k_2 in a range, i.e., $|k_1 - k_2| \leq d_k$, for minimizing the total broadcast time T_{xy}. Also, a node always selects one of the favorable packet-sending patterns: $Int(pn) \in \{1, 3, 5, 7\}$ in a line broadcast hereinafter.

5.2 Performance Dependencies

Bypass schemes. Fig. 9 depicts the impact of various bypass schemes $\boldsymbol{b}$ on $T_{xy}(\text{iBT}(n^2;\boldsymbol{b}),0)$. $\boldsymbol{b}$ is uniform $\langle b_1\rangle$ or hybrid $\langle b_0, b_1\rangle$ with $b_0 \in \{4, 8, 16\}$ and thus b_1 indicates the longest bypass length. The horizontal and vertical axes are b_1 and $T_{xy}(\text{iBT}(n^2;\boldsymbol{b}),0)$.

Network sizes. While increasing the network sizes N from 2K to 65K, Fig. 10 presents $T_{xy}(\text{iBT}(n^2;\boldsymbol{b}),0)$ with selected $\boldsymbol{b} \in \{\langle b_1\rangle, \langle b_0, b_1\rangle\}$ where $b_1 \in \{8, 16, 24\}$. The vertical and horizontal axes are T_{xy} and the network size N expressed in logarithmic scale to the base 2. In this figure, we found that the T_{xy} is the lowest at $\boldsymbol{b} = \langle 8, 32\rangle$.

Dimensionalities. When the dimensionality d runs from 2 to 4, Fig. 11 presents $T_{xy}(\text{iBT}(n^d;\boldsymbol{b}),0)$ where $\boldsymbol{b} \in \{\langle 2d\rangle, \langle 4d\rangle\}$ and $T_{xy}(\text{Torus}(m^{d+1}))$ versus network sizes.

Through fitting these date, we present T_{xy} as functions of network size N:

$$T_{xy}(\text{iBT}(n^2;\boldsymbol{b} = \langle 8, 32\rangle)) = 1.75 \cdot N^{0.2} + 16.22, \quad (7.1)$$

$$T_{xy}(\text{iBT}(n^3;\boldsymbol{b} = \langle 6\rangle)) = 0.54 \cdot N^{0.27} + 14.97, \quad (7.2)$$

$$T_{xy}(\text{iBT}(n^4;\boldsymbol{b} = \langle 8\rangle)) = 1.09 \cdot N^{0.16} + 14.15. \quad (7.3)$$

5.3 Performance Analysis

We compare the performance of the rectangular broadcast over an iBT network and a torus network. In our analysis, the torus network has either the same network dimensions or the same node degree as the iBT network. Using the functions in Table 2, we present $\alpha(f; n, d, \boldsymbol{b})$ and $\beta(f; n, d, \boldsymbol{b})$ where $f \in \{\lambda, \mu, T_x\}$ in Figs. 5, 6, 7, and 8.

The impact of the bypass schemes on the rectangular broadcast is remarkably similar to that on the line broadcast, i.e., an iBT network with a suitable uniform bypass scheme broadcasts as fast as those with hybrid and longer bypass schemes. For example, Fig. 9 shows that $\text{iBT}(64^2;\boldsymbol{b} = \langle 6\rangle)$ is one of the 2D 4K-node iBT networks that broadcast k packets the fastest but it requires the shortest bypass length. Similar to Fig. 3a, Fig. 10 shows that some bypass schemes, e.g., $\boldsymbol{b} = \langle 24\rangle$, have an inferior rectangular broadcast than other schemes.

Fig. 7 shows: as the network size N increases, the fixed-length bypass schemes improve the rectangular broadcast over torus by more than 60 percent and such improvement improves further with increasing network sizes.

Fig. 8 shows: 2D $\text{iBT}(n^2;\boldsymbol{b} = \langle 8, 32\rangle)$ broadcasts faster than 3D $\text{Torus}(m^3)$ in a larger rectangular broadcast group by more than 70 percent. Particularly, $\text{iBT}(64^2;\boldsymbol{b} = \langle 8, 32\rangle)$ is more efficient than $\text{Torus}(16^3)$ in boosting the link utilization as illustrated as in Fig. D of Appendix 3, which is

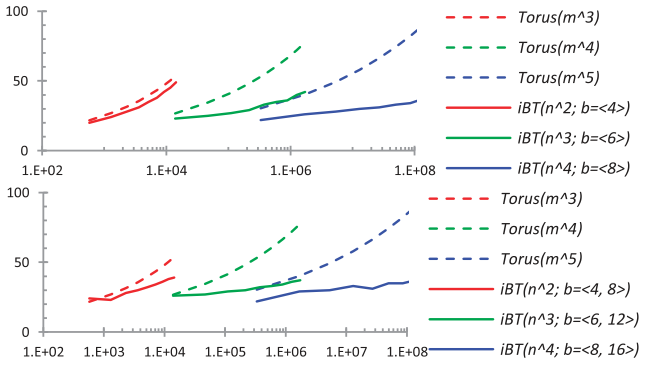

Fig. 11. $T_{xy}(\text{iBT}(n^d;\boldsymbol{b}),0)$ and $T_{xy}(\text{Torus}(m^{d+1}))$ versus N.

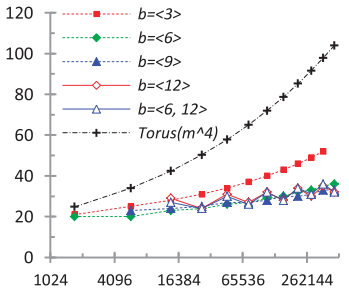

Fig. 12. $T_{all}(\text{iBT}(n^3;\boldsymbol{b}),0)$ versus N.

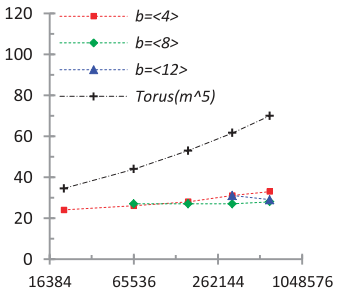

Fig. 13. $T_{all}(\text{iBT}(n^4;\boldsymbol{b}),0)$ versus N.

available in the online supplemental material. Fig. 11 shows: interlacing short-length bypass links, e.g., $\boldsymbol{b}=\langle 2d\rangle$, enables outperforming a torus with the added more dimension.

6 Global Broadcast

6.1 Algorithms

The global broadcast algorithm for general iBT networks can be derived from generalizing the rectangular broadcast algorithm. In a 3D iBT network, three colored groups can be formed to route a message decomposed into three categories: 1-/2-/3-color packets. Starting from the source, 2-colored packets are broadcasted first in y-dimension, then z-dimension, and finally x-dimension. The same procedure can be carried out for two other colored packets until all nodes are filled with the original message. Starting with an assumption: a message is to broadcast from node $\vec{s}$ to all dimensions $(d \geq 2)$, we have the procedures of a global broadcast algorithm.

1. $\vec{s}$ initiates d independent line broadcasts in parallel: broadcast k_i i-color packets in ith dimension.
2. A node $\vec{x}$ broadcasts i-color packets in the jth dimesion $\forall j \in \chi^c(i,k_m(\vec{x},\vec{s}))$ if and only if it satisfies: $i \in \Omega^c(\vec{x},\vec{s})$ and $\chi^c(i,k_m(\vec{x},\vec{s})) \neq \emptyset$.
3. A node $\vec{x}$ forward a newly received i-color packet to all of its neighbors in the dimension the packet comes from if and only if $i \in \Omega^c(\vec{x},\vec{s})$.
4. A node $\vec{x}$ forward a newly received i-color packet to its nearest neighbor $\vec{y}$ if $i \in \Omega(\vec{y},\vec{s})$.

An example of six categories of nodes in a 3D iBT network is illustrated in Fig. 14 of Appendix 5, which is available in the online supplemental material.

6.2 Performance Dependencies

Figs. 12 and 13 present $T_{all}(\text{iBT}(n^d;\boldsymbol{b}))$ for $d \in \{3,4\}$ with various bypass schemes as the network size increases. By fitting these data, we obtain T_{all} as functions of network size N:

$$T_{all}(\text{iBT}(n^2;\boldsymbol{b}=\langle 8,32\rangle)) = 1.75 \cdot N^{0.2} + 16.22, \tag{8.1}$$

$$T_{all}(\text{iBT}(n^3;\boldsymbol{b}=\langle 6\rangle)) = 0.54 \cdot N^{0.28} + 14.71. \tag{8.2}$$

It is difficult at this moment to extra physical meanings of the fitted parameters (coefficients and exponents). It is tempting to argue that these parameters carry first-principle interpretations.

6.3 Performance Analysis

We compare the one-to-all broadcast performance between an iBT network with a fixed bypass scheme and a torus network of either the same network dimensions or the same degree in Figs. 7 and 8. Figs. D and E of Appendix 3, which is available in the online supplemental material, compare the link utilizations of selected 2D/3D iBT networks with 3D/4D torus networks.

Experiments described in Section 6.2 reaffirm that adding bypass links to the torus networks improves the one-to-all broadcasts speed by 80 percent as shown in Fig. 7. The iBT networks constructed this way also outperforms a higher dimensional torus by 60 percent as shown in Fig. 8. Figs. D and E of Appendix 3, which is available in the online supplemental material, also demonstrate that an iBT network can boost the link utilization more efficiently and thus complete the one-to-all broadcast more quickly than a torus.

7 Discussions and Conclusions

We design the line, rectangular, and global broadcast algorithms for iBT networks in a generic way in order to balance the bandwidth utilization and congestion reduction. Performing extensive numerical experiments on these algorithms, we understand their performance dependencies on the network size, dimensionality and bypass scheme.

By fitting the performance metric f in middle-sized iBT networks, we conveniently analyze the scalability of an iBT network with its torus originals in Figs. 5, 6, 7, and 8. For networks with up to 1,000,000 nodes and node degree no more than 10, an iBT network with fixed-length bypass links always outperforms a comparable torus of the same degree. For example, in terms of the diameter and the average node-to-node distance, rectangular and global broadcasts, $\text{iBT}(1000^2;\boldsymbol{b}=\langle 8,32\rangle)$ outperforms the $\text{Torus}(100^3)$ by approximately 80 percent.

The iBT network broadcasts faster than a comparable torus in a regular collective communication pattern. More importantly, the iBT network can broadcast more efficiently than the comparable torus in an irregular pattern.

The maximization of broadcast bandwidth is achieved by the wrap-around connections in torus and by the bypass links in an iBT network. It is, however, difficult to leverage on the wrap-around connections for the irregular broadcast patterns thus broadcast efficiency in a torus is greatly diminished. On the other hand, the iBT network with its short fixed-length bypass links has relaxed such restrictions and improved the performance even for larger networks, as shown in Figs. 7 and 8.

The iBT networks, with strategic addition of bypass links to the underlying torus networks and methodical utilization of such links, not only have a short network diameter and average node-to-node distance but they also provide a natural infrastructure of efficient global communications such as broadcast.

REFERENCES

[1] TOP500. Top 500 Supercomputer Site, http://www.top500.org, 2012.

[2] J. Dongarra et al., "High-Performance Computing: Clusters, Constellations, MPPs, and Future Directions," *Computing in Science and Eng.*, vol. 7, pp. 51-59, 2005.

[3] A. Hoisie et al., "A Performance Comparison through Benchmarking and Modeling of Three Leading Supercomputers: Blue Gene/L, Red Storm, and Purple," *Proc. ACM/IEEE SC Conf. Supercomputing (SC '06)*, p. 3, 2006.

[4] N.R. Adiga et al., "Blue Gene/L Torus Interconnection Network," *IBM J. Research and Development*, vol. 49, Mar./May 2005.

[5] G. Almasi et al., "Design and Implementation of Message-Passing Services for the Blue Gene/L Supercomputer," *IBM J. Research and Development*, vol. 49, pp. 393-406, 2005.

[6] N.R. Adiga et al., "An Overview of the BlueGene/L Supercomputer," *Proc. ACM/IEEE Conf. Supercomputing*, p. 60, 2002.

[7] S.L. Scott and G.M. Thorson, "The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus," *Proc. HOT Interconnect Symp. IV*, Aug. 1996.

[8] E. Chan et al., "Collective Communication on Architectures that Support Simultaneous Communication over Multiple Links," *Proc. 11th ACM SIGPLAN Symp. Principles and Practice of Parallel Programming*, 2006.

[9] W. Liu et al., "Portable and Scalable Algorithm for Irregular All-to-All Communication," *J. Parallel and Distributed Computing*, vol. 62, pp. 1493-1526, 2002.

[10] J. Duato et al., *Interconnection Networks: An Engineering Approach.* Morgan Kaufmann Publishers Inc., 2002.

[11] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks.* Morgan Kaufmann Publishers Inc., 2003.

[12] *Petascale Computing: Algorithms and Applications.* Chapman & Hall/CRC Computational Science, 2008.

[13] P. Zhang, R. Powell, and Y. Deng, "Interlacing Bypass Rings to Torus Networks for More Efficient Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 2, pp. 287-295, Feb. 2011.

[14] A.R. Mamidala et al., "MPI Collectives on Modern Multicore Clusters: Performance Optimizations and Communication Characteristics," *Proc. IEEE Eighth Int'l Symp. Cluster Computing and the Grid*, pp. 130-137, 2008.

[15] G. Almasi et al., "Optimization of MPI Collective Communication on BlueGene/L Systems," *Proc. 19th Ann. Int'l Conf. Supercomputing*, 2005.

[16] A. Faraj and X. Yuan, "Automatic Generation and Tuning of MPI Collective Communication Routines," *Proc. 19th Ann. Int'l Conf. Supercomputing*, 2005.

[17] R. Thakur et al., "Optimization of Collective Communication Operations in MPICH," *Int'l J. High Performance Computing Applications*, vol. 19, pp. 49-66, Feb. 2005.

[18] E.W. Chan et al., "On Optimizing Collective Communication," *Proc. IEEE Int'l Conf. Cluster Computing*, 2004.

[19] A. Faraj et al., "MPI Collective Communications on the Blue Gene/P Supercomputer: Algorithms and Optimizations," *Proc. IEEE 17th Symp. High Performance Interconnects*, pp. 63-72, Aug. 2009.

[20] D.B. Gannon and J.V. Rosendale, "On the Impact of Communication Complexity on the Design of Parallel Numerical Algorithms," *IEEE Trans. Computers*, vol. C-33, no. 12, pp. 1180-1194, Dec. 1984.

[21] R. Rabenseifner, "Automatic Profiling of MPI Applications with Hardware Performance Counters," *Proc. Sixth European Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pp. 22-22, 1999.

[22] J.J. Dongarra et al., "The LINPACK Benchmark: Past, Present and Future," *Concurrency and Computation: Practice and Experience*, vol. 15, pp. 803-820, 2003.

[23] M. Eleftheriou et al., "Performance Measurements of the 3D FFT on the Blue Gene/L Supercomputer," *Proc. 11th Int'l Euro-Par Conf. Parallel Processing*, pp. 795-803, 2005.

[24] P. Mitra et al., "Fast Collective Communication Libraries, Please," *Proc. Intel Supercomputing Users' Group Meeting*, 1995.

[25] J. Pješivac-Grbović et al., "Performance Analysis of MPI Collective Operations," *Cluster Computing*, vol. 10, pp. 127-143, 2007.

[26] S.S. Vadhiyar et al., "Automatically Tuned Collective Communications," *Proc. ACM/IEEE Conf. Supercomputing (CDROM)*, 2000.

[27] R. Thakur and W. Gropp, "Improving the Performance of Collective Operations in MPICH," *Proc. 10th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pp. 257-267, 2003.

[28] L.P. Huse, "Collective Communication on Dedicated Clusters of Workstations," *Proc. sixth European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 1999.

[29] T. Kielmann et al., "Bandwidth-Efficient Collective Communication for Clustered Wide Area Systems," *Proc. 14th Int'l Symp. Parallel and Distributed Processing*, 2000.

[30] Y. Yang and J. Wang, "Near-Optimal All-to-All Broadcast in Multidimensional All-Port Meshes and Tori," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 2, pp. 128-141, Feb. 2002.

[31] A. Faraj et al., "STAR-MPI: Self Tuned Adaptive Routines for MPI Collective Operations," *Proc. 20th Ann. Int'l Conf. Supercomputing*, 2006.

[32] R. Thakur and A.N. Choudhary, "All-to-All Communication on Meshes with Wormhole Routing," *Proc. Eighth Int'l Symp. Parallel Processing*, 1994.

[33] C. Calvin et al., "All-to-All Broadcast in Torus with Wormhole-Like Routing," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 130-137, 1995.

[34] H.H.J. Ben, "Gossiping on Meshes and Tori," *IEEE Trans. Parallel and Distributed Systems*, vol. 9, no. 6, pp. 513-525, June 1998.

[35] U. Meyer and J.F. Sibeyn, "Time-Independent Gossiping on Full-Port Tori," *Max-Planck Institut für Informatik*, Sept. 1998.

[36] M. Barnett et al., "Broadcasting on Meshes with Wormhole Routing," *J. Parallel and Distributed Computing*, vol. 35, pp. 111-122, 1996.

[37] S. Kumar et al., "Optimization of All-to-All Communication on the Blue Gene/L Supercomputer," *Proc. 37th Int'l Conf. Parallel Processing*, pp. 320-329, 2008.

[38] Y. Yang and J. Wang, "Pipelined All-to-All Broadcast in All-Port Meshes and Tori," *IEEE Trans. Computers*, vol. 50, no. 10, pp. 1020-1032, Oct. 2001.

[39] G.E. Fagg et al., "ACCT: Automatic Collective Communications Tuning," *Proc. Seventh European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 2000.

[40] I.-H. Chung et al., "MPI Performance Analysis Tools on Blue Gene/L," *Proc. ACM/IEEE Conf. Supercomputing (SC '06)*, p. 16, 2006.

[41] IBM Blue Gene Team, "Overview of the Blue Gene/P project," *IBM J. Research and Development*, vol. 2, pp. 199-220, 2008.

[42] V. Puente et al., "The Adaptive Bubble Router," *J. Parallel and Distributed Computing*, vol. 61, pp. 1180-1208, 2001.

[43] V. Puente et al., "Adaptive Bubble Router: A Design to Improve Performance in Torus Networks," *Proc. Int'l Conf. Parallel Processing*, 1999.

[44] M. Barnett et al., "Global Combine Algorithms for 2-D Meshes with Wormhole Routing," *J. Parallel and Distributed Computing*, vol. 24, pp. 191-201, 1995.

Peng Zhang received the BS degree in mathematics from Nankai University in 2003, the MS degree in parallel computing from the Nankai Institute of Scientific Computing in 2006, and the PhD degree in applied mathematics from Stony Brook University in 2012. He is working as a postdoctoral research associate of biomedical engineering at Stony Brook University, with the focus of the multiscale computing applications on large-scale parallel computers.

Yuefan Deng received the BS degree in physics from Nankai University in 1983 and MA, M Phil, and PhD degrees from Columbia University, in 1985, 1986, and 1989, respectively. He is working as a professor of applied mathematics at Stony Brook University with 20 years of experience of HPC research. He is a member of the IEEE.