A Priority-based Balanced Routing Scheme for Random Broadcasting and Routing in Tori

Chi-Hsiang Yeh, Emmanouel (Manos) A. Varvarigos, and Abdelhamid Eshoul

Abstract

In this paper, we propose a priority-based balanced routing scheme, called the priority STAR routing scheme, which leads to optimal throughput and average delay at the same time for random broadcasting and routing. In particular, the average reception delay for random broadcasting required in $n_1 \times n_2 \times \cdots \times n_d$ tori with $n_i = O(1)$, n-ary d-cubes with $n = O(1)$, or d-dimensional hypercubes is $O(d + \frac{1}{1-\rho})$. We also study the case where multiple communication tasks for random 1-1 routing and/or random broadcasting are executed at the same time. When a constant fraction of the traffic is contributed by broadcast requests, the average delay for random 1-1 routing required in any d-dimensional hypercube, any n-ary d-cube with $n = O(1)$, and most $n_1 \times n_2 \times \cdots \times n_d$ tori with $n_i = O(1)$ are $O(d)$ based on priority STAR. Our simulation results show that the priority-based balanced routing scheme considerably outperform the best previous routing schemes for these networks.

1. Introduction

Meshes, tori, k-ary n-cubes, and hypercubes are among the most popular network topologies for parallel computers, and numerous algorithms and properties have been proposed and investigated for them [2, 4, 10, 11, 13, 18]. In particular, unicast (node-to-node) routing and broadcasting are the most important communication problems, where a unicast routing task sends a packet from a source node to a certain destination and a broadcasting task [5, 7, 14, 15] copies a packet from a source node to all the other nodes in a network. In a *static communication* environment, a single communication task, such as a broadcast, multinode broadcast (MNB) [4], or total exchange (TE) [4, 8], is performed once and for all. All the nodes know which task they execute and are synchronized to start at the same time. The main objective for a static communication algorithm is to complete the corresponding communication task as soon as possible.

Except for static communication tasks, where conditions are relatively favorable in terms of algorithm development, one can envision situations where communication requests are not deterministic, but are generated at random instants. We call such an environment *dynamic*. The execution of asynchronous computation algorithms is one such situation, but it is reasonable to expect that in many systems a dynamic, largely unpredictable communication environment may be the rule and not the exception. Multitasking, time-sharing, run-time generation of communication requests, and difficulty in identifying the communication tasks at compilation time are some other reasons that make the use of precomputed static communication algorithms (schedules) difficult, and motivate us to find dynamic routing schemes that will run continuously and execute the communication requests on-line. The main objectives for dynamic routing schemes include high maximum throughput and low average delay. Dynamic unicast, also called *random 1-1 routing*, which may be generated by writing to a (nonlocal) memory location in some applications, has been intensively studied in the literature for both parallel computers and general computer networks [3, 4, 13].

Direct application of a static algorithm to its dynamic version may lead to low maximum throughput and large delay. For example, as pointed out by Stamoulis and Tsitsiklis [12], broadcasting based on dimension ordering, which is commonly used for static broadcasting in hypercubes, leads to a maximum throughput factor of $2/d$ (see Section 2 for its definition), which is close to 0 when the dimension d is moderate or large. Investigation of the *random broadcasting* problem was initiated by Stamoulis and Tsitsiklis in [12] for hypercubes, and then considered in [11] for 2-D meshes, and in [16, 17] for hypercubes, d-D meshes, folded-cubes, Manhattan street networks and arbitrary network topologies. In particular, Stamoulis and Tsitsiklis [12] proposed a *direct scheme* based on d completely unbalanced spanning trees and an *indirect scheme* based on d edge-disjoint spanning trees for random broadcasting in d-dimensional hypercubes. The direct scheme in [12] is stable when the throughput factor $\rho < 1$ and requires $O(\frac{d}{1-\rho})$ average broadcast delay and reception delay (see Section 2 for their definitions), while the indirect scheme is stable only when $\rho < \frac{2}{3}$ and requires $O(\frac{d}{2-3\rho})$ average broadcast delay and reception delay. Varvarigos and Bertsekas [16] also formulated and proved

Chi-Hsiang Yeh is with the Dept. of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, K7L 3N6, Canada. Phone: +1 613-533-6368, Fax: +1 613-533-6615, E-mails: yeh@ee.queensu.ca

Emmanouel A. Varvarigos is with the Dept. of Electrical Engineering and Informatics, University of Patras, Patras, Greece.

Abdelhamid Eshoul is with the School of Information Technology and Engineering, University of Ottawa, Ontario, Canada.

Proceedings of the 2003 International Conference on Parallel Processing (ICPP'03)
0190-3918/03 $ 17.00 © 2003 IEEE

the *dynamic broadcasting theorem* for random broadcasting based on partial multinode broadcast (PMNB). The dynamic broadcasting algorithm for d-dimensional hypercubes proposed in [16] is stable when $\rho < 1 - O(\lambda_B d)$ and requires $O(\frac{d}{1-\rho})$ average broadcast delay and reception delay. Varvarigos and Banerjee [17] also proposed a *direct broadcasting scheme* and an *indirect broadcasting scheme* for random broadcasting in arbitrary network topologies. An oblivious routing scheme is a routing scheme where each packet decides (upon its generation) which paths to follow, independently of all other packets in the network. In [12], Stamoulis and Tsitsiklis showed that the lower bounds on the average broadcast delay and average reception delay required by any oblivious routing scheme for random broadcasting in a d-dimensional hypercube are $\Omega(d + \frac{1}{1-\rho})$. Although some of these previously proposed algorithms achieve maximum throughput factor close to 1, none of them can achieve asymptotically optimal delay when the throughput factor is large (i.e., they are usually suboptimal by a factor of $\Theta(d)$).

In this paper, we propose the *priority STAR routing scheme* for random routing and random broadcasting in tori and n-ary d-cubes. We show that random broadcasting can be executed in n-ary d-cubes and $n_1 \times n_2 \times \cdots \times n_d$ tori (i.e., meshes with wraparound) with optimal $O(d + \frac{1}{1-\rho})$ average reception delay when $n, n_i = O(1)$ for all i. Note that general tori are important in that they are incrementally scalable, which is of practical importance; most previous algorithms, however, only consider tori with $n_i = n_j$ for all i and j, and the maximum throughput factor ρ decreases when $n_i \neq n_j$. Also, packets with variable lengths can be broadcast efficiently using our routing scheme, which is not the case for several previous routing schemes for random broadcasting. We conduct computer simulations for our proposed scheme and show that priority STAR considerably outperform the best previous routing schemes for tori.

In a dynamic communication environment, it is common that different types of communication requests, such as unicast, broadcast, multicast, and their multinode versions, are present simultaneously. In previous papers [11, 13, 16, 17] proposing and analyzing dynamic communication algorithms, the authors usually assumed that either unicast or broadcast is the only source of traffic. This is, however, not realistic for the workload of many applications. In this paper, we investigate on *heterogeneous communications* by looking at n-ary d-cubes and general tori where both unicast and broadcast requests are generated dynamically. It can be seen that the traffic generated by random unicast routing in general tori (where $n_i \neq n_j$ for some i, j) is not balanced so that the maximum throughput achieved by a routing scheme that deals with random 1-1 routing and random broadcasting separately, as was done previously in the literature and in practice, is not high. For example, in an $n_1 \times n_2 \times \cdots \times n_d$ tori with $n_1 = n_2 = \cdots = n_{d-1} = n_d/2$, previous methods can only achieve a maximum throughput factor of about 0.67 when 50% of the traffic is generated by unicast and the other 50% is generated by broadcast. In this paper, we show that by using routing schemes that are adaptive to the load created by random broadcasting and random 1-1 routing tasks, network traffic can be exactly balanced over all network links for most situations, leading to maximum throughput factor close to 1 and smaller average delay. By using an appropriate priority discipline, the average delay can be made asymptotically optimal for both random 1-1 routing and random broadcasting. In particular, when a constant fraction of the traffic is generated by broadcast requests, the average delay for random 1-1 routing is only $O(d)$ and the average reception delay for random broadcasting is only $O(d + \frac{1}{1-\rho})$ in any n-ary d-cubes with $n = O(1)$ and most general tori with $n_i = O(1)$, in contrast to $O(\frac{d}{1-\rho})$ for both random routing and random broadcasting using previous routing schemes.

In Section 2, we present the definitions of throughput factor, average broadcast delay, and average reception delay, which are the main performance metrics we will use. In Section 3, we present algorithms for performing random broadcasting in tori, illustrate the central idea of the priority STAR broadcast scheme, and provide simulation results for the average broadcast delays and the average reception delays in tori. In Section 4, we propose the priority STAR routing scheme for random routing and random broadcasting in tori. In Section 5, we conclude the paper.

2. Definitions and Preliminary Results

We define the *throughput factor* (also called *load factor*) as the average utilization of all network links when all the communication tasks are executed using a minimum number of transmissions. More precisely, let λ_i be the arrival rate of communication task type i at a network node and T_i be the minimum number of transmissions required to execute the task, then the throughput factor is given by

$$\rho \stackrel{def}{=} \sum_{i=1}^{t} \frac{\lambda_i T_i N}{L},$$

assuming that all the communication requests are served, where N is the network size, L is the total number of links in the network, and t is the total number of communication types. For example,

$$\rho = \sum_{i=1}^{t} \frac{\lambda_i T_i}{d}$$

for d-dimensional hypercubes and

$$\rho = \sum_{i=1}^{t} \frac{\lambda_i T_i}{2d - 2d/n}$$

for d-D $n \times n \times \cdots \times n$ meshes without wraparound. Note that if the average utilization of a network is a, and the communication algorithms used require a number of transmissions that is a factor of b more than the minimum possible, then the throughput factor of the network is a/b, rather than a. One of the most important objectives when designing dynamic routing schemes is to maximize the maximum

throughput factor so that it is as close to 1 as possible. Note that a throughput factor is always upper bounded by 1, while if a routing scheme is not efficient, the maximum throughput achieved by that scheme may be considerably smaller than 1. For example, when dimension ordering is used, the maximum throughput factor achieved for random broadcasting is only $2/d \ll 1$ [12]. If the arrival rates of communication tasks lead to a throughput factor larger than the maximum throughput achievable by the routing scheme in use, the queue lengths of some/all network links will grow unbounded when queues of infinite length are assumed, or grow with time until they overflow when queues of finite length are assumed, so that the average delays are very large or approach ∞ and retransmissions may be required, further worsening the traffic conditions.

When random routing and random broadcasting are the only types of communication tasks present in an N-node network, the throughput factor is given by

$$\rho \stackrel{def}{=} \lambda_B \frac{N-1}{d_{ave}} + \lambda_R \frac{D_{ave}}{d_{ave}},$$

where λ_B (or λ_R) is the rate at which the source packets to be broadcast (or unicast routed, respectively) are generated, D_{ave} is the average (shortest-path) distance of the network for unicast routing traffic, and d_{ave} is the average number of links per node. More precisely, an N-node network will generate $\lambda_R N$ unicast routing requests and $\lambda_B N$ broadcast tasks per unit of time, which require at least $\lambda_R N D_{ave} + \lambda_B N(N-1)$ packet transmissions per unit of time on the average, where the time unit is taken to be the average transmission time of a packet over a link. Since there are Nd_{ave} directed links in the network, the utilization of the most congested network links is at least equal to the throughput factor ρ. Therefore, a necessary condition for the stability of random broadcasting and routing in any network is that the throughput factor $\rho < 1$. Note that the maximum utilization of all network links is equal to ρ if and only if packets in all unicast tasks are routed through shortest paths, copies of the same source packet of a broadcast task are received exactly once by each node, and the packet transmissions are uniformly distributed over all network links. For example, the throughput factor of a d-dimensional hypercube is given by

$$\rho = \lambda_B \frac{2^d - 1}{d} + \lambda_R \left(\frac{1}{2} + \frac{1}{2(2^d - 1)} \right),$$

assuming that the unicast destinations are uniformly distributed over all network nodes. When random broadcasting is the only type of communication tasks taking place, the throughput factor of an $n \times n$ mesh is given by

$$\rho = \lambda_B \frac{n^2 - 1}{4 - 4/n}.$$

When all network nodes have to receive all the broadcast packets, the maximum throughput factor ρ achievable by any routing scheme in meshes is only 0.5, since some nodes only have two incident links.

The average broadcast delay for random broadcasting is defined as the average time that elapses between the generation of a source packet at a node and the time its broadcast to all the other nodes is completed; the average reception delay is defined as the average time that elapses between the generation of a source packet at a node and the time a particular node receives a copy of the packet, averaged over all nodes. The lower bounds on the average broadcast delay and average reception delay required by any oblivious random broadcasting algorithm for a d-dimensional hypercube are $\Omega(d + \frac{1}{1-\rho})$ when the packets to be broadcast are generated according to a Poisson process [12]. The proof given in [12] for hypercubes can be easily extended to tori and n-ary d-cubes to show that a lower bound on the average broadcast delay and average reception delay required by any oblivious random broadcasting algorithm for an $n_1 \times n_2 \times \cdots \times n_d$ torus is $\Omega(d + \frac{1}{1-\rho})$ when $n_i = O(1)$ for all i. Similarly, we can extend the proof given in [12] to show that when random 1-1 routing is the only traffic source, the average delay required in $n_1 \times n_2 \times \cdots \times n_d$ tori and d-dimensional hypercubes is lower bounded by the network diameter plus the queueing delay at destinations when store-and-forward packet-switching is used, which is $\Omega(d + \frac{1}{1-\rho})$ when $n_i = O(1)$ for all i. When traffic generated by random 1-1 routing and random broadcasting are present at the same time, the average delay experienced by unicast packets in a $n_1 \times n_2 \times \cdots \times n_d$ torus and a d-dimensional hypercube is $\Omega(d)$.

In the following sections, we will present an optimal routing scheme which achieves maximum throughput factor close to 1 and optimal average delay for both random 1-1 routing and random broadcasting. The techniques proposed in this paper can also be applied to other communication problems in various network topologies.

3. Priority-based Broadcast in Tori

In this section, we present an oblivious routing scheme for performing random broadcasting in tori, illustrate the central idea of the scheme, and then analyze its performance.

3.1. STAR Broadcast for Tori

For a given *ending* dimension l, an SDC broadcast algorithm for a d-dimensional $n_1 \times n_2 \times \cdots \times n_d$ torus under the *single-dimension communication (SDC) model* [18, 19], where the nodes are allowed to use only links of the same dimension at any given time, can be presented as follows:

- In Phase 1, the source node sends the packet to be broadcast along dimension $l+1$ via virtual channel 1 if $l+1 \leq d$, or along dimension 1 via virtual channel 2 otherwise.
- In each Phase t, $t = 2, 3, \ldots d$, each node that has a packet forwards the packet along dimension $l+t$ via virtual channel 1 if $l+t \leq d$, or along dimension $l+t-d$ via virtual channel 2 otherwise.

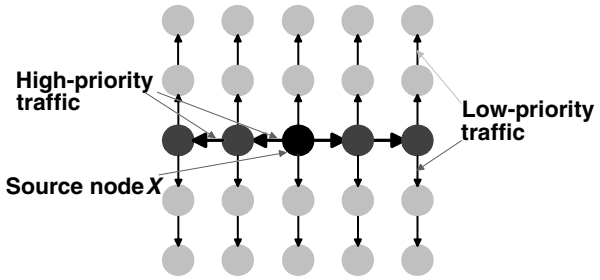

Figure 1. Random broadcasting in a 5×5 torus based on the priority STAR broadcast scheme.

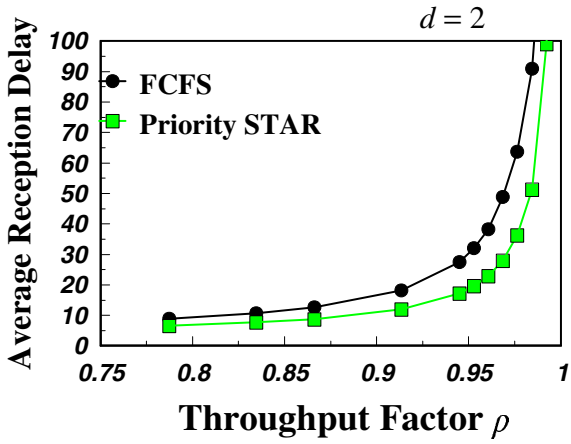

Figure 2. Average reception delays (from simulations) of the priority STAR broadcast scheme and a FCFS generalization of the direct scheme in [12] for random broadcasting in an 8×8 torus with various throughput factors.

The preceding SDC broadcast algorithm is *idling*, in the sense that a link may remain idle even when there is a packet available at its origin end that wants to send it. We can easily modify this algorithm to obtain a *nonidling SDC broadcast algorithm* for random broadcasting under the all-port communication model. More precisely, in the nonidling SDC broadcast algorithm, all the packets are sent along exactly the same path as in the preceding simple broadcast algorithm, but a node forwards all its packets as soon as the associated links are available. For example, the source node will send the packet to all its $2d$ neighbors at time 1 if all its outgoing links are available. Note that there may be other broadcast or 1-1 routing tasks in the network, so some links may be busy. When an associated link is not available, the packet is stored in the associated output queue and waits for service. It can be easily verified that dependency cycles will be formed when there are at least two virtual channels, so the proposed broadcasting scheme is deadlock-free.

The central idea of the STAR broadcast scheme that we propose is to first balance the traffic over all network nodes and links by using an appropriate probability to select each dimension to be the ending dimension, so that throughput is maximized, and then assign an appropriate priority class to each packet so that delay is minimized. Observe that a broadcast task using the preceding nonidling SDC broadcast algorithm generates $a_{l+1,l} = n_{l+1} - 1$ packet transmissions over dimension-$(l+1)$ links, $a_{l+2,l} = (n_{l+2} - 1)n_{l+1}$ packet transmissions over dimension-$(l+2)$ links, and $a_{i,l}$ packet transmissions over dimension-i links for all $i = l+3, l+4, \ldots, d, 1, 2, \ldots, l$, where

$$a_{i,l} = \begin{cases} (n_i - 1)\prod_{j=l+1}^{i-1} n_j = (n_i - 1)n_{i-1}n_{i-2}\cdots n_{l+1} \\ \qquad\qquad\qquad\qquad \text{if} \quad i > l, \\ (n_i - 1)\prod_{j=l+1}^{n} n_j \prod_{j=1}^{j-1} n_j = (n_i - 1)n_{i-1}n_{i-2}\cdots \\ n_1 n_d n_{d-1}\cdots n_{l+1} \qquad\qquad \text{if} \quad i \le l. \end{cases} \tag{1}$$

To balance the traffic, a node needs to select dimension $l = i$ as the ending dimension with certain probability x_i for all $i = 1, 2, \ldots, d$. When there is no traffic other than random broadcasting tasks, the probability vector $(x_1, x_2, \ldots, x_d)$ can be obtained by solving the following system of d linear equations in d unknowns

$$\sum_{j=1}^{d} a_{i,j}x_j = a_{i,1}x_1 + a_{i,2}x_2 + \cdots + a_{i,d}x_d = \frac{N-1}{d} \tag{2}$$

for $i = 1, 2, \ldots, d$, where $N = \prod_{i=1}^{d} n_i$ is the size of the torus. Note that it is guaranteed that the solution to the preceding system of equations satisfies

$$\sum_{i=1}^{d} x_i = x_1 + x_2 + \cdots + x_d = 1$$

since we generate

$$\sum_{i=1}^{d} a_{i,j} = a_{1,j} + a_{2,j} + \cdots + a_{d,j} = N - 1 \tag{3}$$

packets totally for a single broadcast task for any $j = 1, 2, \ldots, d$. ($\sum_{i=1}^{d} x_i = 1$ can be shown by adding all the equations in Eq. (2) together, and then plug Eq. (3) into the resultant equation.) Clearly, if $n_i = n$ for all $i = 1, 2, \ldots, d$ (that is, the torus is an n-ary d-cube), we have $x_j = 1/d$ for all $j = 1, 2, \ldots, d$ since the network is symmetric. A source node that has a packet to broadcast randomly selects dimension $l = i$ with probability x_i as the ending dimension and then use the nonidling SDC broadcast algorithm. If the probabilities x_i's are chosen as the solution to the system Eq. (2), then the expected number of packets to be transmitted on each network link will be the same for all links.

The preceding broadcast scheme for the all-port communication model essentially finds an SDC broadcast algorithm under the SDC model and then rotates the dimensions used by $l = i$ dimensions with probability x_i in order for all broadcast tasks to collectively utilize all dimensions uniformly.

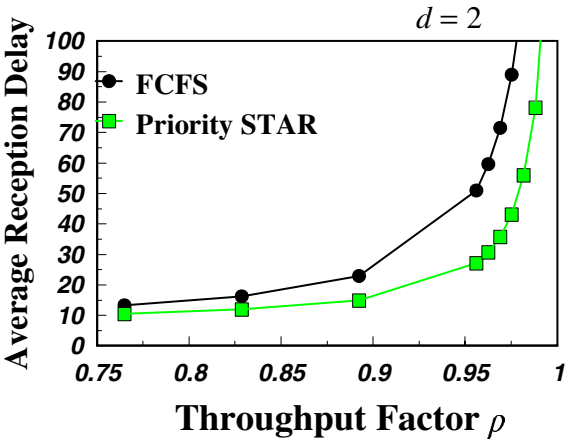

Figure 3. Average reception delays (from simulations) of the priority STAR broadcast scheme and a FCFS generalization of the direct scheme in [12] for random broadcasting in a 16×16 torus with various throughput factors.

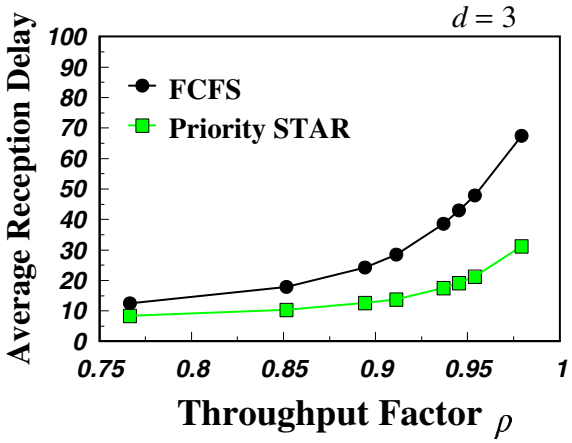

Figure 4. Average reception delays (from simulations) of the priority STAR broadcast scheme and a FCFS generalization of the direct scheme in [12] for random broadcasting in a $8 \times 8 \times 8$ torus with various throughput factors. As compared to Figs. 2 and 3, it can be seen that When the dimension is higher, the superiority of priority STAR is more pronounced.

The resultant broadcast scheme is thus called the *Single-To-All Rotation (STAR) broadcast scheme*. The STAR broadcast scheme can be easily generalized to any product networks to achieve high throughput or maximum throughput (when the traffic can be perfectly balanced over all network links).

3.2. Priority STAR Broadcast for Tori

STAR broadcast can achieve the maximum achievable throughput (i.e., throughput factor $\rho \approx 1$), but the reception and broadcast delays are not optimal. To reduce the delay for broadcasting, we propose to incorporate priority into the STAR broadcast scheme by assigning low priority to the packets that will be forwarded over links of ending dimension l and assigning high priority to the remaining packets. Figure 1 illustrates an example for random broadcasting in a 5-ary 2-cube based on the priority STAR broadcast scheme.

To intuitively illustrate the central idea of our priority STAR broadcast scheme, we first analyze the average reception delay in a torus with $n_i = n$ for all i (i.e., an n-ary d-cube). For simplicity of analysis, we assume that all packets have equal length and require one unit of time for transmission over links in this subsection. Note that the proposed priority STAR broadcast scheme can be applied, without modifications, to general cases where packets may have different lengths. We let ρ_H be the arrival rate of high-priority packets at a node and ρ_L be the arrival rate of low-priority packets. (Since the transmission time of a packet is 1, ρ_H and ρ_L are also the load factors for high-priority packets and low-priority packets, respectively.) We also let V_H and V be the variances of the number of high-priority packets that arrive or are generated at a node during a time slot and that of any packets (i.e., including both low-priority and high-priority packets), respectively. Due to the symmetry of an n-ary d-cube, we can see that the values of ρ_H, ρ_L, V_H, and V are the same at every network node. Also, similar to the analysis given in the following subsection, we can show that $V_H = O(\rho_H)$ and $V = O(\rho)$. Since there are $N/n-1$ high-priority packets and $(1-1/n)N$ low-priority packets generated by a broadcast task, we have $\rho_H < 1/n$ and $\rho = \rho_H + \rho_L < 1$ when the system is stable. Therefore, each of the queues for high-priority packets is a G/D/1 queue [3, 9, 12] with very small arrival rate, and the average waiting time for a high-priority packet is equal to

$$W_H = \frac{V_H}{2\rho_H(1-\rho_H)} - \frac{1}{2} = O\left(\frac{\rho_H}{1-\rho_H}\right) = O(1/n) = o(1).$$

According to the conservation law [9], the average waiting time in a queue will not be affected by assigning different priority classes to packets when the arrival process remains the same and the assignment of priority classes is independent of the service time of the packets. (This is true since the service time is a constant in this analysis example.) Therefore, the average waiting time for packets (including both low-priority and high-priority packets) in our priority STAR broadcast scheme is given by that of a G/D/1 queue with arrival rate ρ and variance V and is equal to

$$W = \frac{V}{2\rho(1-\rho)} - \frac{1}{2} = O\left(\frac{\rho}{1-\rho}\right).$$

Also, we have

$$W = \frac{N-N/n}{N-1} W_L + \frac{N/n-1}{N-1} W_H.$$

Thus, the average waiting time for low-priority packets is

$$W_L \approx W = O\left(\frac{\rho}{1-\rho}\right).$$

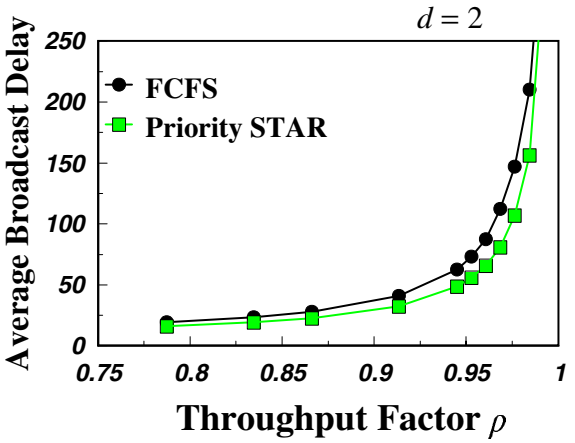

Figure 5. Average broadcast delays (from simulations) of the priority STAR broadcast scheme and a FCFS generalization of the direct scheme in [12] for random broadcasting in an 8×8 torus with various throughput factors.

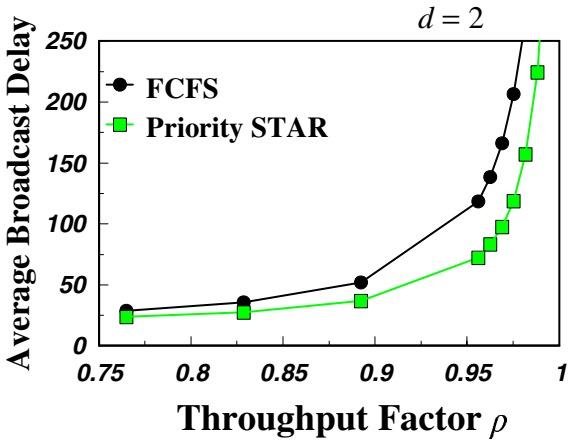

Figure 6. Average broadcast delays (from simulations) of the priority STAR broadcast scheme and a FCFS generalization of the direct scheme in [12] for random broadcasting in a 16×16 torus with various throughput factors.

In the preceding random broadcasting algorithm, a packet is forwarded as a high-priority packet for at most $\lfloor n/2 \rfloor (d-1)$ steps and is forwarded as a low-priority packet for at most $\lfloor n/2 \rfloor$ steps before it is received by a node. Moreover, since only $1/n - 1/N$ out of the total traffic is high-priority traffic, the average waiting time for a high-priority packet is very small $[O(1/n) = o(1)]$. Since the average waiting time for a low-priority packet is $O(\frac{1}{1-\rho})$, the average reception delay is given by

$$O\left(nd + \frac{n}{1-\rho}\right).$$

When the number n of nodes along each dimension of the k-ary n-cube is a constant, the average reception delay is $O(d + \frac{1}{1-\rho})$ and is asymptotically optimal, as can be seen by comparing with the lower bound $\Omega(d + \frac{1}{1-\rho})$ shown in [12] for any oblivious algorithm. As a comparison, by generalizing the broadcast scheme proposed in [12] for random broadcasting in n-ary d-cubes or torus, the average reception delay is $O(\frac{dn}{1-\rho})$ and is suboptimal by a factor of $\Theta(d)$ even when $n = O(1)$. Intuitively, the improvements obtained by our scheme are due to the fact that a broadcast packet traverses most of its path (except for the last few transmissions on the broadcast tree) as a high priority packet with small queueing delay, and only a small part of its path as a low priority packet with high queueing delay. For this to happen, it is also important that high-priority transmissions form a small (or constant) fraction of the total number of transmissions, since there are fewer transmissions on the part of the tree closer to the root than on the part of the tree closer to the leaves. Our priority STAR broadcast scheme also improves on the average reception delay of the random broadcasting algorithm for arbitrary network topology proposed in [17] by a factor of $\Theta(d)$ when the throughput factor is large. The analysis given in this section can be easily generalized to tori with an arbitrary number of nodes along each dimension. Since hypercubes are a special case of tori, the algorithms proposed in this section can also be applied to hypercubes [21].

In Figs. 2-7, we conduct computer simulations to evaluate the performance of the priority STAR broadcast scheme and compare it with the generalization of the broadcast scheme proposed in [12] based on first-come first-serve (FCFS). It can be seen that by simply incorporating priority into broadcasting, the reception and broadcast delays can both be reduced considerably, especially when the throughput is high. Also, the superiority of the proposed STAR broadcast scheme is more pronounced when the dimension of the torus is higher, as predicted in our analysis and comparisons. Another implication of our results is that if we limit the average reception delay and/or the average broadcast delay for an application to be below certain thresholds, then a priority-based broadcast scheme like priority STAR can achieve a higher throughput.

4. Heterogeneous Communications in Tori

In a dynamic communication environment, it is common that different types of communication requests, such as unicast, broadcast, multicast, scatter, gather, accumulation, and their partial or full multinode versions, are present simultaneously. All previous work investigating dynamic communication problems [11, 13, 16, 17], however, assumes that either unicast or broadcast is the only source of traffic. In this section, we investigate on heterogeneous communications in n-ary d-cubes and general tori where both random routing and random broadcasting traffic are present simultaneously (see Fig. 8).

To perform unicast routing in a torus, we send the packet along the shortest path between the source and destination nodes. Let λ_B and λ_R be the arrival rates of source packets

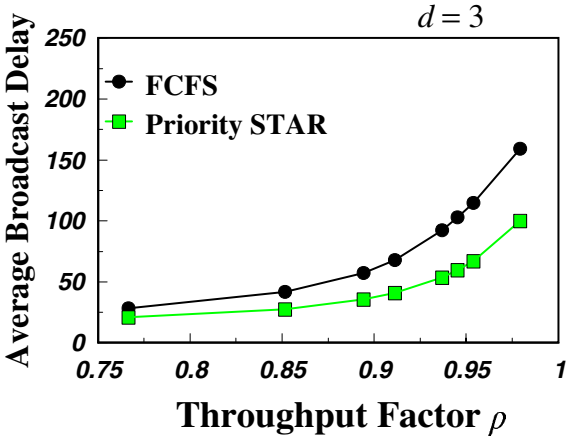

Figure 7. Average broadcast delays (from simulations) of the priority STAR broadcast scheme and a FCFS generalization of the direct scheme in [12] for random broadcasting in an $8 \times 8 \times 8$ torus with various throughput factors. As compared to Figs. 5 and 6, it can be seen that When the dimension is higher, the superiority of priority STAR is more pronounced.

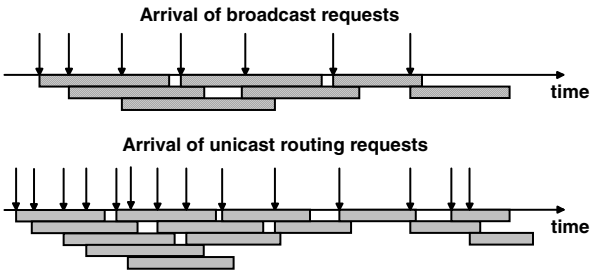

Figure 8. A heterogeneous communication environment where unicast and broadcast requests are generated at each node of a network or parallel computer at random instances. Several broadcast tasks and many unicast tasks may be present simultaneously. For example, if unicast and broadcast requests create comparable amount of network traffic and the throughput factor ρ of the network does not approach zero, then there are an average of $\Theta(d^2n + \frac{dn}{1-\rho})$ broadcast tasks and an average of $\Theta(dN)$ unicast tasks being executed in an n-ary d-cube simultaneously using our routing scheme. (There are an average of $\Theta(\frac{d^2n}{1-\rho})$ broadcast tasks and an average of $\Theta(\frac{dN}{1-\rho})$ unicast tasks being executed in an n-ary d-cube simultaneously using previous routing schemes that do not assign relatively higher priority to packets for unicast.) Note that when the traffic is heavier, the average broadcast delay for random broadcasting is larger due to the increased queueing delay.

at a node for random broadcasting and random 1-1 routing, respectively. Since the average distance of an n_i-node ring is $\lfloor n_i/4 \rfloor$, random 1-1 routing generates $\lambda_R \lfloor n_i/4 \rfloor$ transmissions per unit of time on the average on dimension-i links for all $i = 1,2,...,d$. Note that the traffic is not balanced since the average utilization of a dimension-i link is approximately proportional to the length of the dimension i, and the maximum utilization among all dimension-i links is at least equal to this average no matter what routing algorithm is used. However, if a parallel system is executing random broadcasting tasks simultaneously, then we can balance the traffic over all network nodes and links using the STAR or REDO [20] broadcast algorithm and changing the probability with which each broadcast tree is used.

To balance the traffic when using the STAR broadcast algorithm, a node needs to select the ending dimension as $l = i$ with an appropriate probability x_i, $i = 1,2,...,d$. The probability vector $(x_1,x_2,...,x_d)$ can be obtained by solving the following system of d linear equations in d unknowns

$$\sum_{j=1}^{d} a_{i,j}\lambda_B x_j + \lambda_R \left\lfloor \frac{n_i}{4} \right\rfloor = \lambda_B \frac{N-1}{d} + \lambda_R \frac{\sum_{i=1}^{d} \lfloor n_i/4 \rfloor}{d}, \quad (4)$$

where, $a_{i,l}$ are given by Eq. (1). (The solution to the system of equations Eq. (4) is guaranteed to satisfy $\sum_{i=1}^{d} x_i = 1$.) If $n_i = n$ for all $i = 1,2,...,d$ (that is, the torus is an n-ary d-cube), we have $x_j = 1/d$, $j = 1,2,...,d$. In order to broadcast a packet, its source randomly selects i as the ending dimension with probability x_i and then uses the STAR broadcast algorithm. Then the expected number of packets on each network link generated by the random broadcasting and unicast routing algorithms is the same for all links as long as we can find a legitimate solution to the system of equations (i.e, all the probabilities x_i should be nonnegative numbers no larger than 1). If we obtain an infeasible solution, for example, $x_1 > 1$ and $x_2 < 0$ for a 2-D mesh, we should use probability vector $(1,0)$ instead of (x_1,x_2). Such situations only occur when λ_R is very large (that is, the utilization of some links is very close to 1 for the traffic generated by random 1-1 routing alone) and the value of n_i for certain dimension(s) i is considerably larger than those of other dimensions. In such a case, the maximum utilization of the network links is only slightly increased by the traffic generated by the random broadcasting algorithm. Similar to the discussion given in Section 3, we can also use a REDO broadcast algorithm [20] with a probability vector obtained by solving a different system of linear equations.

By applying the priority STAR routing scheme to random broadcasting and random 1-1 routing in tori, the number of transmissions is minimized (since all the packets are sent along the shortest paths for routing and exactly $N-1$ transmissions are generated by a broadcast task), and the transmissions are uniformly distributed over all network nodes and links, assuming that a feasible solution exists (i.e., $0 \leq x_i \leq 1$ for all i). In such a case, our routing and broadcast algorithms are stable as long as the throughput factor $\rho < 1$, where

$$\rho = \lambda_B \frac{N-1}{2d} + \lambda_R \frac{\sum_{i=1}^{d} \lfloor n_i/4 \rfloor}{2d}.$$

This can be shown by arguing that the queue of any network link will not build up to infinite length.

To reduce both the average delay for random 1-1 routing and the average reception delay for random broadcasting, we can assign high priority to all the unicast packets and all the broadcast packets except for those transmitted along the ending dimension. As a result, when a constant fraction of the traffic is generated by broadcast requests, the average queueing delay at a node for unicast packets (which have high priority) is a small constant so that the average delay for random 1-1 routing is $O(nd)$ in an n-ary d cubes or an $n_1 \times n_2 \times \cdots \times n_d$ torus with $\sum_{i=1}^{d} n_i = O(nd)$. Similar to the analysis given in Section 3, we can show that the resultant average reception delay for random broadcasting is $O\left(nd + \frac{n}{1-\rho}\right)$ in an n-ary d cubes or an $n_1 \times n_2 \times \cdots \times n_d$ torus with $\max n_i = O(n)$. To further reduce the average reception delay for random broadcasting, we can assign medium priority to all the unicast packets, low priority to broadcast packets transmitted along the ending dimension, and high priority to the rest of the broadcast packets.

5. Conclusion

In this paper, we proposed the priority STAR routing scheme for random broadcasting and routing in tori, n-ary d-cubes, and hypercubes. The proposed routing scheme leads to average reception delays that are are optimal within a factor asymptotically equal to 1 when the throughput factor is close to 0 and within a small constant factor for any other throughput factor. The priority STAR broadcast scheme also improves the best previous routing schemes significantly and can achieve optimal average reception delays. Moreover, we showed that by combining random broadcasting and random 1-1 routing, the traffic in general tori can be exactly balanced over all network links for most situations, leading to maximum throughput factor $\rho \approx 1$.

References

[1] Abraham, S. and K. Padmanabhan, "Performance of the direct binary n-cube network for multiprocessors, *IEEE Trans. Computers,* vol. 38, no. 7, Jul. 1989, pp. 1000-1011.

[2] Bertsekas, D.P., C. Ozveren, G.D. Stamoulis, P. Tseng, and J.N. Tsitsiklis, "Optimal communication algorithms for hypercubes," *J. Parallel Distrib. Computing*, vol. 11, no. 4, Apr. 1991, pp. 263-275.

[3] Bertsekas, D.P. and R. Gallager, *Data Networks*, Prentice Hall, Englewood Cliffs, N.J., 1992.

[4] Bertsekas, D.P. and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods,* Athena Scientific, 1997.

[5] Bruck, J., L. De Coster, N. Dewulf, C.-T. Ho, and R. Lauwereins, "On the design and implementation of broadcast and global combine operations using the postal model," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 3, Mar. 1996, pp. 256-265.

[6] Greenberg, A.G. and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Communications,* vol. 40, no. 6, Jun. 1992. pp. 1070-1081.

[7] Ho, C.-T. and M.-Y. Kao, "Optimal broadcast in all-port wormhole-routed hypercubes," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 2, Feb. 1995, pp. 200-204.

[8] Johnson, S.L. and C.-T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Computers*, vol. 38, no. 9, Sep. 1989, pp. 1249-1268.

[9] Kleinrock, L., *Queueing Systems, Vol. II: Computer Applications,* John Wiley & Sons, New York, 1976.

[10] Leighton, F.T., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes,* Morgan-Kaufman, San Mateo, CA, 1992.

[11] Modiano, E. and A. Ephremides, "Efficient algorithms for performing packet broadcasts in a mesh network," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, Aug. 1996, pp. 639-648.

[12] Stamoulis, G.D. and J.N. Tsitsiklis, "Efficient routing schemes for multiple broadcasts in hypercubes," *IEEE Trans. Parallel Distrib. Sys.,* vol. 4, no. 7, Jul. 1993, pp. 725-739.

[13] Stamoulis, G.D. and J.N. Tsitsiklis, "The efficiency of greedy routing in hypercubes and butterflies," *IEEE Trans. Communications,* vol. 42, no. 11, Nov. 1994, pp. 3051-3061.

[14] Tsai, Y.J. and P.K. McKinley, "A broadcast algorithm for all-port wormhole-routed torus networks," *IEEE Trans. Parallel Distrib. Sys.,* vol. 7, no. 8, Aug. 1996, pp. 876-885.

[15] Tseng, Y.C., "A dilated-diagonal-based scheme for broadcast in a wormhole-routed 2D torus," *IEEE Trans. Computers,* vol. 46, no. 8, Aug. 1997, pp. 947-952.

[16] Varvarigos, E.A. and D.P. Bertsekas, "Dynamic broadcasting in parallel computing," *IEEE Trans. Parallel Distrib. Sys.,* vol. 6, no. 2, Feb. 1995, pp. 120-131.

[17] Varvarigos, E.A. and A. Banerjee, "Routing schemes for multiple random broadcasts in arbitrary network topologies," *IEEE Trans. Parallel Distrib. Sys.,* vol. 7, no. 8, Aug. 1996, pp. 886-895.

[18] Yeh, C.-H., "Efficient low-degree interconnection networks for parallel processing: topologies, algorithms, VLSI layouts, and fault tolerance," Ph.D. dissertation, Dept. Electrical & Computer Engineering, Univ. of California, Santa Barbara, Mar. 1998.

[19] Yeh, C.-H. and E.A. Varvarigos, "Macro-star networks: efficient low-degree alternatives to star graphs," *IEEE Trans. Parallel Distrib. Sys.,* vol. 9, no. 10, Oct. 1998, pp. 987-1003.

[20] Yeh, C.-H., E.A. Varvarigos, and H. Lee, "An optimal routing scheme for multiple broadcast," *Proc. Int'l Conf. Parallel and Distributed Systems,* Dec. 1998, pp. 342-349.

[21] Yeh, C.-H., E.A. Varvarigos, and H. Lee, "The priority broadcast scheme for dynamic broadcast in hypercubes and related networks," *Proc. Symp. Frontiers of Massively Parallel Computation,* Feb. 1999, pp. 294-301.