Routing Schemes for Multiple Random Broadcasts in Arbitrary Network Topologies

Emmanouel A. Varvarigos and Ayan Banerjee

Abstract—We consider the problem where packets are generated at each node of a network according to a Poisson process with rate λ, and each of them has to be broadcast to all the other nodes. The network topology is assumed to be an arbitrary bidirectional graph. We derive upper bounds on the maximum achievable broadcast throughput, and lower bounds on the average time required to complete a broadcast. These bounds apply to any network topology, independently of the scheme used to perform the broadcasts. We also propose two dynamic broadcasting schemes, called the indirect and the direct broadcasting scheme, that can be used in a general topology, and we evaluate analytically their throughput and average delay. The throughput achieved by the proposed schemes is equal to the maximum possible, if a half-duplex link model is assumed, and is at least equal to one half of the maximum possible, if a full-duplex model is assumed. The average delay of both schemes is of the order of the diameter of the trees used to perform the broadcasts. The analytical results obtained do not use any approximating assumptions.

Index Terms—General graphs, edge-disjoint trees, dynamic broadcasting, queuing systems.

1 Introduction

Broadcasting is the operation where a packet is copied from a node to all the other nodes of a network. In this paper, we consider the *dynamic broadcasting* problem, where broadcast requests are generated at random time instants at each node of a multiprocessor network that has an arbitrary topology. In particular, we assume that packets are generated at each node according to a Poisson process with rate λ, and that each of them has to be broadcast to all other nodes. We are interested in finding efficient routing schemes to perform the broadcasts in a general topology, and in evaluating their performance. The assumption of Poisson arrivals is made only because the mathematics of the analysis require it, and it is inessential for the implementation of the schemes that we propose. The dynamic broadcasting problem arises, for example, in iterations of the form

$$x = f(x_1, \ldots, x_n), \tag{1}$$

where x is an n-dimensional vector. Here we assume that iteration (1) is executed asynchronously, with processor i storing and updating the component x_i, and broadcasting the new value of x_i when it changes appreciably. The dynamic broadcasting problem arises in many other situations, and we believe it deserves a position among the generic network routing problems. Schemes that run continuously, and execute on-line the broadcast requests should be part of the communication primitives of any multiprocessor network.

Previous works (see, e.g., [2], [3], [10], [17]) have mainly dealt with finding optimal schedules to execute prototype (and usually well-structured) broadcast communication tasks in certain regular topologies. This is different from the model that we adopt in this paper where packets are generated continuously, over an infinite time horizon. The dynamic broadcasting problem was first addressed by Stamoulis and Tsitsiklis in [14] for hypercubes, and Varvarigos and Bertsekas in [20] and [18] for hypercubes and d-dimensional meshes, respectively. The routing schemes and analytical results that we develop in the present paper are considerably more general, since they apply to arbitrary network topologies, that may or may not have any symmetry properties. Also the upper bounds on the throughput and the lower bounds on the delay obtained for general topologies use techniques that are considerably more general and complicated than those used for hypercube or mesh networks. The throughput and delay for unicast (one-to-one) communication have been examined extensively in the literature for a number of topologies (see, for example, [6], [7], [8], [11], [19]), with the analysis being approximate, except for the results given by Stamoulis and Tsitsiklis in [15] for hypercubes and butterflies, which did not use any approximating assumptions. We believe that the throughput and the delay of a network for broadcast (one-to-many) communication are equally important criteria in evaluating network performance. The analysis to be presented on the dynamic broadcasting problem will not use any independence or other approximating assumptions. We consider this particularly important, since the analysis of problems that involve networks of queues is in general extremely difficult. For example, no accurate analysis exists for the throughput and delay of the corresponding unicast communication problem in a general network topology, despite the efforts of many researchers.

We will use three criteria in order to evaluate the performance of a dynamic broadcasting scheme in a given network topology. The first criterion is the *maximum broadcast throughput*, which is the maximum generation rate λ per node that can be accommodated by a broadcasting

• *The authors are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106-9560. Email: manos@ece.ucsb.edu.*

Manuscript received Dec. 12, 1994; revised Feb. 27, 1996.
For information on obtaining reprints of this article, please send e-mail to: transpds@computer.org, and reference IEEECS Log Number D95218.

1045-9219/96$05.00 ©1996 IEEE

scheme with the queueing delays being finite. The second criterion is the *average broadcast delay* $\mathcal{B}$, which is the average time that elapses between the generation of a packet at a node and the time its broadcast to all the other nodes is completed. The third criterion is the *average reception delay* $\mathcal{R}$, which is the average time that elapses between the generation of a packet at a node and the time a particular node s receives a copy of the packet, averaged over all nodes s of the network. Since for the broadcast of a packet to be completed all nodes must receive a copy of it, we have $\mathcal{R} \leq \mathcal{B}$, for any broadcasting scheme. The average reception delay is important in the case where a processor can start processing a packet as soon as it receives it, without having to wait for the packet to be delivered to all the other processors. For example, if the iteration of (1) converges under the totally asynchronous model (see [2, chapter 6]), then processors can be allowed to compute faster and execute more iterations than others, without waiting at predetermined points for messages from other processors to arrive. This happens, for example, when the function f in (1) is a contraction mapping (as is the case when $f(x) = Ax + b$ with the spectral radius of $|A|$ satisfying $\rho(|A|) < 1$, or when $f(x) = \min_{\mu \in M}(c(\mu) + aP(\mu)x)$, corresponding to a Markovian decision problem), or a monotone mapping (as is the case when $f_i(x) = \min_{j \in A(i)}(a_{ij} + x_{ij})$, corresponding to a Bellman-Ford algorithm, or for certain network flow problems; see [2] for more examples).

We assume that store-and-forward switching is used for the transfer of data. All packets have equal length and they require one unit of time (or slot) in order to be transmitted over a link. We consider two communication models. In the first model, called Multiple Link Availability Full Duplex model (or F-D model, for brevity), a node is capable of transmitting and receiving messages along all its incident links concurrently, and each link can be used for transmission along both directions simultaneously. In the second model, called Multiple Link Availability Half Duplex model (or H-D model, for brevity), a node can use all its incident links simultaneously, but a link can be used for transmission along only one direction at a time.

We show that a necessary stability condition under the F-D model is $\lambda < \min\{2k_{\max}/N, d_{\min}/(N-1)\}$, where $k_{\max}$ is the maximum number of edge-disjoint spanning trees, $d_{\min}$ is the minimum node degree, and N is the number of nodes of the network. When the H-D model is assumed, the corresponding bound on the throughput is $\lambda < k_{\max}/N$. We also give lower bounds on the average broadcast delay $\mathcal{B}$ in terms of the parameters $k_{\max}$ and $d_{\min}$. The bounds derived hold for *any* network, and for *any* scheme that can be used to perform the broadcasts.

We introduce two new routing schemes, called the *indirect broadcasting* and the *direct broadcasting* schemes, which can be used to perform the broadcasts in an arbitrary bidirectional graph G. We evaluate the throughput of the schemes under the F-D and the H-D model, and we compare it to the corresponding upper bounds. We also obtain analytical expressions for the average broadcast delay and the average reception delay of the schemes.

The indirect broadcasting scheme is proved to be stable for $\lambda < k_{\max}/N$, under both the F-D and the H-D model. Therefore, the maximum throughput achieved by the indirect scheme for a given network under the H-D model is equal to the maximum possible for that model. When the F-D model is assumed, the throughput of the indirect scheme is between 0.5 and 1 of the maximum possible (given by the upper bound). We also show that, under the F-D model, this is the best stability region that we could expect for an algorithm that works for *general* topologies (that is, for some—but not all—topologies, $\lambda < k_{\max}/N$ is also a necessary stability condition under the F-D model). In the course of developing the indirect broadcasting scheme, we introduce a new communication task, called the *generalized multinode broadcast*, and propose efficient algorithms to execute it in a general topology. The generalized multinode broadcast, in addition to being an important component of the direct broadcasting scheme, it is also important on its own merit, since it arises in a number of other applications. We also evaluate the average broadcast delay of the indirect broadcasting scheme, and show that it is of the order of the average diameter of the spanning trees used by the scheme (or of the order of the maximum diameter of the spanning trees used, if a simpler but less efficient GMNB algorithm is used), for any load in the stability region.

The second broadcasting scheme that we consider is the direct broadcasting scheme, which is a particularly simple scheme to implement. We analyze its throughput under the F-D model, and prove that the scheme is stable for $\lambda < k_{\max}/N$. We also obtain an expression for the average reception delay of the scheme and show that it is of the order of the average mean internodal distance of the spanning trees used, for any load in the stability region.

The remainder of the paper is organized as follows. In Section 2, we describe the notation that we will use, and give some preliminary results. In Section 3, we derive upper bounds on the throughput, and lower bounds on the average broadcast delay that apply to any broadcasting scheme in a given network. In Section 4, we describe and analyze the proposed dynamic broadcasting schemes. In particular, Subsection 4.1 deals with the indirect broadcasting scheme, while Subsection 4.2 deals with the direct broadcasting scheme. Finally, in Section 5, we conclude the paper.

2 Notation and Preliminary Results

In this section we introduce the notation, and present some preliminary results.

The network topology is given by a general (bidirectional) graph $G = (N, \mathcal{E})$. A *graph* $G = (\mathcal{N}, \mathcal{E}\})$ is defined as a set $\mathcal{N}$ of nodes, viewed as the processors of the network, and a collection $\mathcal{E}$ of pairs of distinct nodes in $\mathcal{N}$. Each pair $e = [s, t]$ of $\mathcal{E}$ is called an *edge*, and corresponds to a bidirectional communication link between processors s and t. The cardinality of the sets $\mathcal{N}$ and $\mathcal{E}$ is denoted by N and E, respectively. An edge $[s, t]$ of G consists of the two unidirectional links (s, t) and (t, s) to be referred to as *arcs*. Given an arc (s, t), we refer to node s as the start of the arc and to node t as the tail of the arc. The *diameter* δ of a graph is defined as the maximum shortest distance between any two nodes of the network. A *span-*

ning tree is a subgraph of G that includes all the vertices of G, and is a tree. A set of spanning trees are *edge-disjoint*, if no two of them have any edges in common. If a given node is considered as the root of a tree, the *depth* of the tree is defined as the maximum distance between the root and any other node of the tree.

A communication task that will be useful in the description of our algorithms is the *generalized multinode broadcast* (abbreviated GMNB), where some arbitrary nodes of a network G have a total of M packets to broadcast. During a GMNB, a node may have more than one packets to broadcast (as opposed to the partial multinode broadcast task defined in [18], where each node has at most one packet to broadcast). Nodes that have at least one packet to broadcast are called *active* nodes. The following lemma deals with the GMNB task in a tree.

LEMMA 1. *The GMNB in a tree network, under the F-D or the H-D model, requires at most $K + L - 1$ slots, where L is the diameter of the tree, and K is the total number of packets that have to be broadcast.*

PROOF (abbreviated). Consider the GMNB algorithm where each node transmits the packets originated at that node on all its incident links, one after the other. A (nonleaf) node transmits each packet that it receives from its neighbors on all its incident links, except for the link over which it received it. No link remains idle if there is a packet that wants to use it, and conflicts over the use of the links are resolved according to a FIFO discipline (although any other discipline would also work). In the case of the H-D model we have the additional constraint that a link can be used for transmission only in one direction at a time. If two neighboring nodes u and v have packets that wants to use link $[u, v]$, one of them is arbitrarily selected for transmission, and the others are queued (we assume that the two end nodes of a link have a way to decide which of them will transmit first).

It takes $l_{s,t}$ transmissions for a packet to move from node t to node s. Since there are K packets in the system, and the paths followed by them are nonoverpassing (in the sense that if two paths share some links and then split, they will never meet again), a packet may be delayed for a total of at most $K - 1$ time units. Therefore, node s will receive a packet from node t after at most $l_{s,t} + K - 1$ slots. Hence, the broadcast of all packets is completed after at most $K - 1 + \max_{s,t} l_{s,t} = K - 1 + L$ slots. Note that the worst case completion time for both the F-D and the H-D model arises when all the K packets are initially located at a leaf of the tree. □

Given a network of diameter δ, it is always possible to find a spanning tree $\mathcal{T}$ of depth δ rooted at some node of the network. Using the notion of the *postorder traversal* on $\mathcal{T}$, we define an ordering of the network nodes of $\mathcal{T}$. In the postorder traversal, the subtrees connected to the root are visited first in the order from left to right, followed by the root, with this order of traversal carried recursively on each subtree. The ordering of the network nodes defined by postorder traversal on $\mathcal{T}$ will be denoted by "<".

Given a graph G, a spanning tree $\mathcal{T}$, and a set of active nodes that have a total of M packets to broadcast, we define the rank r_s of node s as $r_s = \sum_{t<s} x_t$, where "<" is the order defined by $\mathcal{T}$, and x_t is the number of packets that node t has to broadcast. The ranks of all nodes can be computed in 2δ steps by performing a parallel prefix operation on $\mathcal{T}$ (see [9, pp. 37-44]). During the parallel prefix operation each node also learns the total number of packets M that have to be broadcast.

LEMMA 2. *Consider a graph G that has k edge-disjoint spanning trees $T_1, \ldots, T_k$ with diameters $L_1 \le \cdots \le L_k = L_{\max}$, respectively. The GMNB task in G, under the F-D or the H-D model, can be performed in at most*

$$T_{GMNB} \le \frac{M}{k} + L_{\max} + 2\delta$$

slots, where M is the total number of packets that have to be broadcast, and δ is the diameter of G.

PROOF. We first perform a parallel prefix operation to compute the rank r_s of each node s, and the total number of packets M. This requires 2δ steps. Based on the values of r_s and M, each node s can decide (in a way to be discussed shortly) the tree on which each of the packets originated at s will be broadcast. Assume that a total of n_j packets are to be broadcast along tree T_j, $j = 1, \ldots, k$, where $\sum_{j=1}^{k} n_j = M$. The completion time of all transmissions on tree T_j is (by Lemma 1) at most equal to $n_j + L_j - 1$. Thus, the total time required to perform the GMNB in G satisfies

$$T_{GMNB} \le \max_{j=1,\ldots,k} \left(n_j + L_j\right) - 1 + 2\delta. \qquad (2)$$

One way to decide which packets are broadcast on each tree is the following. The x_s packets originated at each node s are given distinct sequense numbers in the set $\{r_s, r_s + 1, \ldots, r_s + x_s - 1\}$. Node s then transmits the packet with sequence number $r_s + j - 1$, along tree $T_{(r_s+j) \bmod k}$. In this way, packets are divided equally among the trees, so that at most $\lceil M/k \rceil$ packets are broadcast on each tree. In that case, we have $n_j \le \lceil \frac{M}{k} \rceil$ for all j, and the total time required to perform the GMNB satisfies

$$T_{GMNB} \le \left\lceil \frac{M}{k} \right\rceil + \max_{j} L_j + 2\delta - 1. \qquad \square$$

The GMNB algorithm described in Lemma 2 is particularly easy to implement. In the following lemma we present an algorithm that requires more computation at the nodes, but results in a smaller number of communication steps (slots).

LEMMA 3. *Consider a graph G that has k edge-disjoint spanning trees $T_1, \ldots, T_k$ with diameters $L_1 \le \cdots \le L_k$, respectively. The GMNB task in G, under the F-D or the H-D model, can be performed in at most*

$$T_{GMNB} \le \frac{M}{k} + \bar{L} + 2\delta$$

slots, where

$$\bar{L} = \frac{\sum_{i=1}^{k} L_i}{k}$$

is the average diameter of the spanning trees, M is the total number of packets that have to be broadcast, and δ is the diameter of G.

PROOF. The idea is to use the algorithm of Lemma 2, but assign more packets to trees of small diameters so as to minimize the $\max_j(n_j + L_j)$ in (2). This can be achieved by running the following "water-mark" algorithm.

Initialize: $n_j = 0$ for all j.
For $i = 1$ to M: {Let $J = \arg\min_{j=1,2,\ldots,k}(L_j + n_j)$. Assign packet i to tree T_J, and increment n_J.}

The watermark algorithm may assign no packets to trees with large diameters. Let T_m, $m \le k$, be the highest numbered tree that is assigned any packets. Equation (2) gives

$$T_{GMNB} \le \max_{j=1,\ldots,m}\left(n_j + L_j\right) + 2\delta - 1 \le$$

$$2\delta - 1 + \left\lceil \frac{M + \sum_{i=1}^{m} L_i}{m} \right\rceil \le 2\delta + \frac{M + \sum_{i=1}^{m} L_i}{m}. \quad (3)$$

If $m = k$, the lemma follows immediately from (3). If $m \ne k$, then, since T_m is the last tree that is assigned any packets and L_i is increasing with i, we have

$$M + \sum_{i=1}^{m} L_i / m \le L_{m+1} \le \sum_{i=m+1}^{k} L_i / (k - m),$$

which gives, after some algebraic manipulation,

$$\frac{M + \sum_{i=1}^{m} L_i}{m} \le \frac{M + \sum_{i=1}^{k} L_i}{k}.$$

The last inequality together with (3) completes the proof. □

The algorithms described in Lemmas 2 and 3 are both distributed; an active node s only needs to know its rank r_s and the total number of packets M, both of which are available at s after the parallel prefix operation. For the algorithm of Lemma 3, each node also needs to know the diameters L_i, $i = 1, 2, \ldots, k$, of the edge-disjoint spanning tree, so that it can locally execute the watermark algorithm. Since all nodes run the same watermark algorithm (ties are resolved in the same way) with the same data, they all arrive at consistent results regarding the trees on which the packets are broadcast. The time complexity given in Lemma 3 takes into account only the number of communication steps (slots), and does not include computations at the nodes.

3 Universal Upper Bounds on the Throughput and the Delay

In this section, we derive upper bounds on the maximum arrival rate λ per node that can be accommodated by a given network. The bounds apply to any network topology, and they hold for any scheme that can be used to execute the broadcasts in that topology. The analysis in both the current and the next section will assume the F-D model for the network links; whenever the H-D model is used, this will be stated explicitly.

Given a general graph $G = (\mathcal{N}, \mathcal{E}\})$, a *partition* P of the vertices of G is a collection of nonempty disjoint subsets of $\mathcal{N}$ whose union is $\mathcal{N}$. We let $E_P(G)$ be the set of edges of G that connect nodes belonging to different constituent sets of P. We define the graph $G_P = (P, E_P(G))$ as the graph obtained by shrinking each member of P to a single vertex, and keeping only the edges connecting nodes that belong to different sets of P (note that G_P may have more than one edges connecting a given pair of vertices in P). It is known ([12, Theorem 1]) that a graph G has k edge-disjoint spanning trees if and only if

$$k \le \frac{|E_P(G)|}{|P| - 1}$$

for every partition P of $\mathcal{N}$, where | | denotes the cardinality of a set. Therefore, the maximum number of edge-disjoint spanning trees is given by

$$k_{\max} = \min_{P} \frac{|E_P(G)|}{|P| - 1}. \quad (4)$$

Edge-disjoint spanning trees can be constructed using the Matroid Partition Algorithm (see [16] and [5, pp. 85-87]), which returns in polynomial time k edge-disjoint spanning trees, if they exist, and a negative answer if they do not exist.

The following lemma gives an upper bound on the maximum broadcast throughput per node.

LEMMA 4. *A necessary condition for stability for any broadcasting scheme is* $\lambda \le \frac{2k_{\max}}{N}$.

PROOF. Consider a particular partition P. Each packet generated has to undergo at least $|P| - 1$ transmissions on links of the set $E_P(G)$. Since the total rate at which broadcasts are generated in the network is equal to $N\lambda$, and there is a total of $2|E_P(G)|$ unidirectional links corresponding to the edges in $E_P(G)$, a necessary condition for stability is

$$\lambda N(|P| - 1) \le 2|E_P(G)|$$

for every partition P, or equivalently,

$$\lambda \le \frac{2}{N} \min_{P} \frac{|E_P(G)|}{|P| - 1} = \frac{2k_{\max}}{N}, \quad (5)$$

where we have used (4). □

The following lemma gives an upper bound on the maximum throughput in terms of the minimum degree $d_{\min}$ of the nodes.

LEMMA 5. *A necessary condition for stability for any broadcasting scheme is* $\lambda \le \frac{d_{\min}}{N-1}$.

PROOF. Consider a node s of the network with indegree $d_{\min}$. Since a total of $\lambda(N - 1)$ packets will have to cross the incoming links of s per unit of time, the lemma follows. □

Lemma 5 is also valid for directed (not necessarily bidirectional) graphs with $d_{\min}$ being the minimum indegree of the nodes. Equation (5) is a necessary stability condition for any network and any broadcasting scheme in that network. It should be noted, however, that there exist networks for which more stringent stability conditions hold. In particular, it is easy to construct networks for which $k_{\max} = d_{\min}$ (such networks include d-dimensional meshes without wraparound and trees). For such networks, the necessary stability condition becomes $\lambda \leq \frac{k_{\max}}{N-1}$. This implies that we cannot hope to find a broadcasting scheme that will work for *all* topologies and will be stable for $\lambda \leq \frac{2k_{\max}}{N}$ (since such a scheme would not work for networks for which $k_{\max} = d_{\min}$). There are, however, networks for which stability can be guaranteed for any $\lambda < 2k_{\max}/N$ (an example of such a network is hypercube, as shown in [20], asymptotically, as the number of nodes N becomes large).

In what follows we obtain lower bounds on the average broadcast delay, which is defined as the average time that elapses between the generation of a packet at node and the time its broadcast to all the other nodes is completed. The bounds to be derived hold for any scheme that executes the broadcasts in a given network. The proofs of the lemmas use techniques similar to those developed in [14] for hypercube networks. Unless stated otherwise, the model assumed is the F-D model.

LEMMA 6. *For any graph G, the average broadcast delay $\mathcal{B}$ satisfies*

$$\mathcal{B} = \Omega\left(\frac{1}{1 - \frac{\lambda N}{2k_{\max}}}\right),$$

where N is the number of nodes, λ is the rate at which broadcasts are generated at each node, and $k_{\max}$ is the maximum number of edge-disjoint spanning trees of the network.

PROOF. Consider any partition $P = \{S_1, \ldots, S_{|P|}\}$ of the nodes of the graph, and let $E_P(G)$ be the set of edges connecting nodes in different sets of the partition. To obtain a lower bound on the delay, we assume that upon the generation or reception of a packet at a node in S_i, it is immediately available at all other nodes in S_i. This favorable assumption clearly underestimates the average broadcast delay. For a broadcast to be completed, each packet has to undergo at least $|P| - 1$ transmissions on links of the set $E_P(G)$. If we focus on the set of links $E_P(G)$ and view them as servers, then we obtain an $M/D/m$ system with arrival rate equal to $N\lambda$ (which is the total arrival rate to the network), service time equal to $|P| - 1$, and number of servers $m = 2|E_P(G)|$. Therefore, the average broadcast delay $\mathcal{B}$ of any broadcasting scheme satisfies $\mathcal{B} \geq T_{M/D/m}$, where $T_{M/D/m}$ is the average delay of the $M/D/m$ system defined above. Using the lower bound on the delay of $M/D/m$ systems given in [4] we obtain

$$\mathcal{B} = \Omega\left(\frac{1}{1 - \frac{\lambda N(|P| - 1)}{2|E_P(G)|}}\right). \tag{6}$$

Since (6) holds for any partition P, the lemma follows from (4). □

The following lemma gives an lower bound on the average broadcast delay in terms of the minimum degree $d_{\min}$ of the network nodes. Its proof is similar to a proof given in [14].

LEMMA 7. *Let $d_{\min}$ be the minimum degree of the nodes of a network. Then*

$$\mathcal{B} = \Omega\left(\frac{1}{1 - \frac{\lambda(N-1)}{d_{\min}}}\right) \tag{7}$$

for any broadcasting scheme.

Lemma 7 also gives a lower bound on the average reception delay $\mathcal{R}$ for networks whose nodes have equal degrees. It also applies to directed graphs with $d_{\min}$ being the minimum of the indegrees of the nodes. The lower bound of Lemma 6 is valid for any graph, independently of the scheme used to execute the broadcasts. However, as the following corollary shows, there exist networks for which tighter bounds on the average broadcast delay hold.

COROLLARY 1. *There exists a network for which the average broadcast delay satisfies*

$$\mathcal{B} = \Omega\left(\frac{1}{1 - \frac{\lambda(N-1)}{k_{\max}}}\right)$$

for any broadcasting scheme.

PROOF. Consider a network that has $k_{\max} = d_{\min}$, and use Lemma 7. □

The bounds on the throughput and the average broadcast delay given in Lemmas 4 and 6 assume that information can be transmitted on a link in both directions at the same time (F-D model). When a link can be used only in one direction at a time (H-D model), the bounds can be modified as follows.

LEMMA 8 (half-duplex model). *A necessary stability condition under the H-D model is $\lambda \leq k_{\max}/N$. Also the average broadcast delay $\mathcal{B}$ of any broadcasting scheme satisfies*

$$\mathcal{B} = \Omega\left(\frac{1}{1 - \frac{\lambda N}{k_{\max}}}\right).$$

Note that the bounds of Lemmas 5 and 7 also hold for the H-D model; since, however, we always have $k_{\max} \leq d_{\min}$, they are not as strict (for the H-D model) as those given by Lemma 8.

4 Broadcasting Algorithms

In this section we propose and analyze two dynamic broadcasting schemes for arbitrary network topologies. Both schemes use edge-disjoint spanning trees as distribution trees on which the packets are broadcast. In the first scheme, called the *indirect broadcasting scheme,* the broadcasts are performed by executing successive GMNB algorithms. In the second scheme, called the *direct broadcasting scheme,* each packet generated selects randomly one of the edge-disjoint spanning trees and is broadcast on it.

4.1 Indirect Broadcasting Scheme

One way to execute the broadcasts is by successively executing generalized multinode broadcast (GMNB) algorithms, each starting when the previous one has finished. If the graph has k edge-disjoint spanning trees with maximum diameter $L_{\max}$ and the algorithm described in Lemma 2 (where the packets are equally split among the trees) is used, the GMNB can be performed in time

$$T_{GMNB} \le \frac{M}{k} + L_{\max} + 2\delta .$$

If the GMNB algorithm of Lemma 3 (where packets are assigned to edge-disjoint spanning trees so that the delays over all trees are nearly equal) is used, the GMNB can be performed in time

$$T^{*}_{GMNB} \le \frac{M}{k} + \overline{L} + 2\delta ,$$

where $\overline{L}$ is the average diameter of the edge-disjoint spanning trees. Note that both algorithms complete the GMNB task in at most $MX + V$ slots, where M is the number of packets and X and V are some known scalars.

In the indirect broadcasting scheme, the time axis is divided into GMNB periods, each starting when the previous one has finished. Each GMNB period can (conceptually) be divided into two parts. The first part is called the *notification interval,* and its duration can be upper bounded by a known constant V that depends only on the size of the network and is independent of the number of packets M (in particular, if the GMNB algorithm of Lemma 2 or 3 is used, $V = L_{\max} + 2\delta$ or $V = \overline{L} + 2\delta$, respectively). During the notification interval, each active node s can be viewed as informing the other nodes that it intends to broadcast its x_s packets (this is done by merely participating in the parallel prefix operation). The second part of a GMNB period is called the *broadcast interval,* and its duration is equal to XM. The broadcast interval is empty if there are no packets to broadcast ($M = 0$). Even though the duration of each GMNB period is random (because packet arrivals are random), it is known to all the nodes of the network, because each node learns during the broadcast interval the total number of packets M, and, from there, the duration of the following broadcast interval. Therefore, if all nodes initiate the dynamic broadcast scheme at the same time, and accurate local clocks are available, no further synchronization is needed. If the local clocks are not accurate and nodes do not start the parallel prefix phase at the same time, this results in an increase in the effective duration 2δ of the parallel prefix phase (or, alternatively, in some nodes missing the opportunity to transmit packets during a period). The end of the parallel prefix phase can then be used to resynchronize nodes.

In order to analyze the performance of the indirect broadcasting scheme the following auxiliary queueing system, called *gated vacation system,* will be useful. Consider a queuing system where customers arrive at a rate of λN customers per unit of time (slot) and require $X = 1/k$ time units each in order to be served. In addition to serving customers, the server occasionally takes a vacation in order to perform some organizational work. In particular, the time axis at the server is divided into service intervals, where customers are served, and vacation intervals. When the server returns from a vacation it serves all the customers that have arrived prior to the beginning of the preceding vacation period. When all eligible customers have been served, the system takes a vacation of duration V. The gated vacation system has been analyzed in [1], where the following theorem was proved.

Theorem 1. *Let the arrival process of customers at the gated vacation system be a Poisson process with rate λN, and the customer service times and vacation durations be constant and equal to X and V, respectively. Then the mean waiting time in queue for this system is*

$$W = \frac{\rho X}{2(1-\rho)} + \frac{V}{2} + \frac{V}{1-\rho}, \qquad (8)$$

where $\rho = \lambda N X$.

The next theorem gives the average broadcast delay and the stability region of the indirect broadcasting scheme in an arbitrary network topology.

Theorem 2. *Assume that for a given N-processor network there exists an algorithm that performs the GMNB communication task in time $XM + V$, where M is the number of packets that have to be broadcast and X, V are scalars that are independent of M (they may depend on the network under consideration). Assume also that during the GMNB algorithm each node learns the value of M. Then the indirect broadcasting scheme that uses this GMNB algorithm has the following performance characteristics. If the packets to be broadcast are generated at each node of the network according to a Poisson process with rate λ, independently of the other nodes, the average broadcast delay $\mathcal{B}$ satisfies*

$$\mathcal{B} \le W + X + \min(W - V, \rho W), \qquad (9)$$

where $\rho = \lambda N X$ and W is given by (8).

Proof. Each active node participates in a GMNB period with all the packets generated at that node prior to the beginning of the GMNB period (that is, prior to the beginning of the vacation interval of that period). The duration of the broadcast interval of a GMNB period is at most MX time units, where M is the number of eligible packets at the start of a period.

We will refer to the indirect broadcasting scheme as system "*b*" (for "broadcast"), and to the gated vacation system as system "*a*" (for "auxiliary"). Let the vacation and customer service times of system "*a*" be constant and equal to V and X, respectively. Consider the following analogy between systems "a" and "b." Let a service interval of system "a" correspond to a broadcast interval of system "b," and an arrival of a customer in system "a" correspond to the generation of a broadcast request in system "b." Note the similarities between the two systems. During a service interval of system "a" (or broadcast interval of system "b") all customers (or broadcast requests, respectively) can be served, provided that they arrived prior to the beginning of the current period. It is easy to see that the probability distributions of the length of the vacation intervals (which are fixed), the length of the service (or broadcast) intervals, and the number of customers (or broadcast requests) served in a service interval are identical for both systems. In particular, the duration of a vacation interval of system "a" and a notification interval of system "b" are both equal to V by construction. The duration of a broadcast interval of system "b" is equal to MX, where M is the number of packets present in the system at the beginning of (the parallel prefix operation of) the notification interval. Similarly, the duration of a service interval of system "a" is equal to MX, where M is the number of packets in the customers at the beginning of the preceding vacation interval. The only difference between the two systems is that in system "b" a broadcast is completed at the end of a GMNB period, while in system "a" customers complete service at times jX, $j = 1, 2, \ldots, M$, from the beginning of the service interval.

The waiting time W_a in queue for a packet of the auxiliary system is given by Theorem 1. Let U_1 be the average time between the beginning of a service interval of system "a" and the time that a customer served in this interval starts service (see Fig. 1). Similarly, let U_2 be the average time between the completion of service of a customer of system "a" and the end of the corresponding service interval. It can be seen that $U_1 = U_2 \le W_a - V$. We denote by K the number of customers found in the system by an arriving customer. We then have

$$U_1 = U_2 \le E(K)X = \lambda N W_a X = \rho W_a,$$

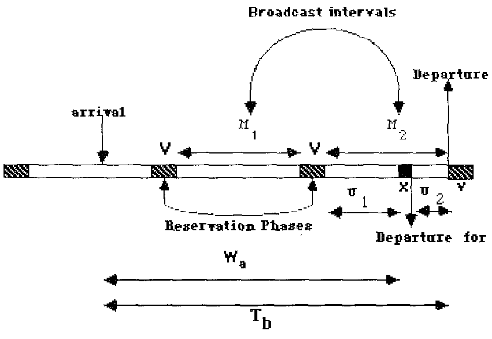


Fig. 1. Notification (or vacation) and broadcast (or service) intervals for the network broadcasting scheme (or the auxiliary queuing system, respectively).

where we have used Little's theorem $E(K) = \lambda N W_a$. The average broadcast delay $\mathcal{B}$ satisfies

$$\mathcal{B} = W_a + X + U_2 \le W_a + X + \min(W_a - V, \rho W_a),$$

which completes the proof. □

The following theorem is the main result of this subsection.

THEOREM 3. *Let G be a (bidirectional) network that has $k_{\max}$ edge-disjoint spanning trees. Let $L_{\max}$ (or $\overline{L}$) be the maximum diameter (or the average diameter, respectively) of the edge-disjoint spanning trees, and δ be the diameter of G. The indirect broadcasting scheme that uses the GMNB algorithm of Lemma 2 is stable for*

$$\lambda < \frac{k_{\max}}{N},$$

and has average broadcast delay $\mathcal{B}$ that satisfies

$$\mathcal{B} \le \frac{\rho^2 - \rho + 2}{2k_{\max}(1 - \rho)} + \frac{(L_{\max} + 2\delta)(1 + \rho)(3 - \rho)}{2(1 - \rho)}, \quad (10)$$

where $\rho = \lambda N / k_{\max}$. Moreover, the average broadcast delay for light load satisfies

$$\mathcal{B} \le 1.5 L_{\max} + 3\delta + \frac{1}{k_{\max}}, \quad \rho \approx 0.$$

If the GMNB algorithm of Lemma 3 is used, then $L_{\max}$ in the above expressions should be replaced by $\overline{L}$.

PROOF. If we use the GMNB algorithm described in Lemma 2 (or Lemma 3), then the average broadcast delay $\mathcal{B}$ of the indirect broadcasting scheme can be obtained from Theorem 2 by substituting $X = 1/k_{\max}$ and $V = L_{\max} + 2\delta$ (or $V = \overline{L} + 2\delta$, respectively), which after some algebraic manipulation gives (10). The indirect broadcasting scheme is, therefore, stable for $\rho = \lambda N X = \frac{\lambda N}{k_{\max}} < 1$. □

The stability region of the indirect broadcasting scheme is half of that given by the universal upper bound of (5). This is the best that we could hope for a general network, since the necessary condition of (5) is not tight for all networks. Indeed, as discussed in Section 3, a necessary stability condition for a network with $d_{\min} = k_{\max}$ is $\lambda \le \frac{k_{\max}}{N}$ (such networks include d-dimensional meshes without wraparound, and trees).

For any fixed load in the stability region, the average broadcast delay $\mathcal{B}$ is $O(L_{\max} + \delta)$ or $O(\overline{L} + \delta)$, depending on the GMNB algorithm that we use. For topologies where the edge-disjoint spanning trees can be chosen so that the maximum diameter $L_{\max}$ or the average diameter $\overline{L}$ is $O(\delta)$, the average broadcast delay of the indirect broadcasting scheme is of the optimal order (networks where this is possible include trees, and meshes of arbitrary dimension with or without wraparound). This is because, under our communication model, the network diameter is a lower bound on any broadcasting task. Since $\overline{L} \le L_{\max}$, the indirect broadcasting scheme that uses the algorithm of Lemma 3 always performs better than the one that uses the algorithm of Lemma 2; this improvement in the performance, however, comes at the expense of a more complicated implementation and more computations at the nodes. Note also that there is a trade-off between stability region and average broadcast delay. If we use the maximum possible number of edge-disjoint spanning trees (in order to have the best stability region), we may have to select trees of large diameter (increasing the delay for light load). A related interesting problem is to consider whether edge-disjoint spanning trees with diameter less than, say, some constant B can be found in polynomial time (computing edge-disjoint spanning trees without any constraint on the diameter can be solved in polynomial time, as we mentioned earlier).

4.2 Direct Broadcasting Scheme

In this subsection, we propose and analyze an alternative dynamic broadcasting scheme, which we call the *direct broadcasting scheme*. We assume that the graph G under consideration has k edge-disjoint spanning trees, denoted by T_j, $j = 1, \ldots, k$. In the direct broadcasting scheme, each packet selects upon its generation one of the k disjoint trees, and is broadcast on it. A packet that is broadcast on tree T_j will be referred to as a packet of class j. At every slot, every node v considers each of its incident links $(v, w) \in T_j$. If v has received a packet of class j that it has neither sent already to w nor it has yet received from w, then v sends such a packet on link (v, w). If v does not have such a packet it sends nothing on (v, w). When more than one packets are eligible for transmission on link (v, w), one of them is transmitted and the remaining are queued.

The direct broadcasting scheme is a particularly simple scheme to implement. The only information carried by a packet is its class j, and each node v only has to know which incident links are associated with each spanning tree. In what follows, we evaluate the broadcast throughput and the average reception delay of the direct broadcasting scheme for an arbitrary network topology. The following lemma, proved by Stamoulis and Tsitsiklis [14], will be useful in our analysis.

LEMMA 9. *Consider a tree $\tilde{T}$. Let s_0 be the root, $s_1, s_2, \ldots, s_n$ be the nonroot nodes of $\tilde{T}$, and d_i, $i = 1, \ldots, n$, be the distance from node s_i to s_0. We assume that packets are generated at each nonroot node s_i, according to a Poisson process with rate λ, and each of them is destined for node s_0. All packets require one slot for transmission over a link, and the root node s_0 can remove at most one packet per slot. The average delay D between the arrival of a packet at a node, and the time it is removed by the root is*

$$D = \frac{1}{2} + \frac{\lambda n}{2(1 - \lambda n)} + \frac{1}{n}\sum_{i=1}^{n} d_i . \tag{11}$$

Moreover, the system is stable if and only if $\lambda n < 1$.

Consider now the direct broadcasting scheme in a network that has k edge-disjoint spanning trees, T_j, $j = 1, 2, \ldots, k$. We assume the F-D model, where a node can transmit or receive packets over all its incident links simultaneously, and a link can be used for transmission in both directions at the same time. We let p_j be the probability with which a packet selects T_j as the spanning tree on which it will be broadcast. We also let l^j_{st} be the distance between nodes s and t using only links of tree T_j. We will initially focus on a particular node s and spanning tree T_j, and calculate the average reception delay $\mathcal{R}_j(s)$ for packets received at s over tree T_j (the average reception delay can then be calculated by averaging over all nodes s and trees T_j). In order to evaluate $\mathcal{R}_j(s)$, it is useful to view node s as the root of T_j and let $T_{j,1}(s), T_{j,2}(s), \ldots, T_{j,m}(s)$ be the subtrees in which T_j is partitioned when s is removed (see Fig. 2). We denote by $n_{j,q}(s)$ the number of nodes of tree $T_{j,q}(s)$ (therefore, $\sum_{q=1}^{m} n_{j,q}(s) = N - 1$). Using Lemma 9, it can be seen that the average reception delay $\mathcal{R}_{j,q}(s)$ for packets received at node s over the subtree $T_{j,q}$ is given by

$$\mathcal{R}_{j,q}(s) = \frac{1}{2} + \frac{\lambda p_j n_{j,q}(s)}{2\left(1 - \lambda p_j n_{j,q}(s)\right)} + \frac{\sum_{t \in T_{j,q}} l^j_{st}}{n_{j,q}(s)} . \tag{12}$$

The average reception delay $\mathcal{R}_j(s)$ for packets received at node s over tree T_j is given by

$$\mathcal{R}_j(s) = \frac{\sum_{q=1}^{m} n_{j,q}(s)\mathcal{R}_{j,q}(s)}{N-1} = \frac{1}{2} + l^j_s + \frac{\lambda p_j}{2(N-1)} \sum_{q=1}^{m} \frac{\left[n_{j,q}(s)\right]^2}{1 - \lambda p_j n_{j,q}(s)} , \tag{13}$$

where

$$l^j_s = \frac{\sum_{t \in T_j} l^j_{st}}{N-1} \tag{14}$$

is the average internodal distance from every node of the network to node s using only links of T_j.

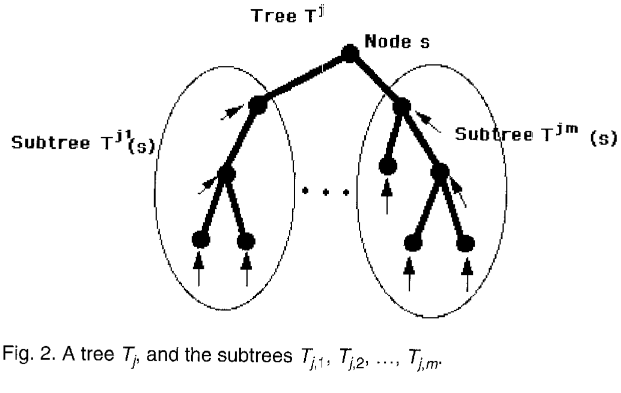


Fig. 2. A tree T_j, and the subtrees $T_{j,1}, T_{j,2}, \ldots, T_{j,m}$.

The average reception delay at node s (averaged over all trees T_j, $j = 1, 2, \ldots, k$) is

$$\mathcal{R}(s) = \sum_{j=1}^{k} p_j R_j(s) = \frac{1}{2} + \sum_{j=1}^{k} p_j l^j_s + \frac{\lambda}{2(N-1)} \sum_{j=1}^{k}\sum_{q=1}^{m} \frac{p_j^2\left[n_{j,q}(s)\right]^2}{1 - \lambda p_j n_{j,q}(s)} . \tag{15}$$

The average reception delay $\mathcal{R}$ can be found by averaging (15) over all nodes s, that is,

$$\mathcal{R} = \frac{1}{2} + \sum_{j=1}^{k} p_j l^j + \frac{\lambda}{2N(N-1)} \sum_{s=1}^{N} \sum_{j=1}^{k} \sum_{q=1}^{m} \frac{p_j^2 n_{j,q}(s)^2}{1 - \lambda p_j n_{j,q}(s)}, \quad (16)$$

where

$$l^j = \sum_{s=1}^{N} \frac{l_s^j}{N}$$

is the average internodal distance of tree T_j. The direct broadcasting scheme is stable if and only if

$$\lambda < \frac{1}{p_j n_{j,q}(s)}, \quad \text{for all } s, j, q. \quad (17)$$

Let $\hat{s}$ be a leaf node of the spanning tree T_j. When $\hat{s}$ is viewed as the root of T_j (equivalently, when considering packets received at node $\hat{s}$ over tree T_j), there is only one subtree $T_{j,1}$ connected to $\hat{s}$, having $n_{j,1}(\hat{s}) = N - 1$ nodes. Thus, the condition of (17) is equivalent to $\lambda < 1/(p_j(N-1))$ for all j. The maximum stability region is achieved when the number of edge-disjoint spanning trees used is equal to the maximum possible, and each packet selects with equal probability the tree on which it is broadcast (that is, when $k = k_{\max}$, and $p_j = 1/k_{\max}$ for all j). In that case, the direct broadcasting scheme is stable if and only if

$$\lambda < \frac{k_{\max}}{N-1}. \quad (18)$$

Therefore, the stability region of the direct broadcasting scheme is one half of the universal upper bound given in (5) for the F-D model, and (in view of the discussion in Section 3) it is the best that a general broadcasting scheme could achieve. In that case, the average reception delay at node s is

$$\mathcal{R}(s) = \frac{1}{2} + l_s + \frac{\lambda}{2k_{\max}(N-1)} \sum_{j=1}^{k_{\max}} \sum_{q=1}^{m} \frac{n_{j,q}(s)^2}{1 - n_{j,q}(s)\,\lambda/k_{\max}}, \quad (19)$$

where $l = \sum_{j=1}^{k_{\max}} l_s^j / k_{\max}$. The average reception delay $\mathcal{R}$ can similarly be found as

$$\mathcal{R} = \frac{1}{2} + l + \frac{\lambda}{2k_{\max}N(N-1)} \sum_{s=1}^{N} \sum_{j=1}^{k_{\max}} \sum_{q=1}^{m} \frac{n_{j,q}(s)^2}{1 - n_{j,q}(s)\,\lambda/k_{\max}}, \quad (20)$$

where $l = \sum_{s=1}^{N} l_s / N$ is the mean internodal distance of the spanning trees, averaged over all nodes in a tree and over all trees. Since $n_{j,q}(s) \leq N - 1$ for all j, q, and s, the average reception delay is $O(l)$ for any load in the stability region. In the usual case where the network under consideration is symmetric, (20) may be considerably simplified. When the network is operating at a load that is considerably smaller than $k_{\max}/N$, maximizing the stability region may not be the primary concern. In such a case one should choose the probabilities p_j so as to minimize $\mathcal{R}$ for a given load λ. For example, when $\lambda \approx 0$, the best strategy is to broadcast all packets on the tree that has the smallest mean internodal distance.

5 Conclusions

We have proposed dynamic broadcasting schemes for a general network topology. We have obtained analytic expressions for the average broadcast delay, the average reception delay, and the stability region of the schemes without using any simplifying approximating assumptions. The performance results obtained were compared with corresponding universal bounds that were also derived. The broadcasting schemes are efficient, simple to implement, and do not make any assumption about the underlying network topology. The approach taken in this paper may be useful in dealing with other problems that do not necessarily involve broadcasts (for example, Theorem 2 essentially gives a general relationship between static and dynamic problems, which could be useful in analyzing other routing problems). Another line of future research may be to apply the results obtained to particular topologies of interest, by finding edge-disjoint spanning trees with small diameter in these topologies (see [2]).

Acknowledgment

This research was supported by the National Science Foundation under Grant NSF-RIA-08930554.

References

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

[2] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, N.J.: Prentice Hall, 1989.

[3] D.P. Bertsekas, C. Ozveren, G.D. Stamoulis, P. Tseng, and J.N. Tsitsiklis, "Optimal Communication Algorithms for Hypercubes," *J. Parallel and Distributed Computing*, vol. 11, pp. 263-275, 1991.

[4] S.L. Brumelle, "Some Inequalities for Parallel-Server Queues," *Operations Research*, vol. 19, pp. 402-413, 1971.

[5] C.J. Colbourn, *The Combinatorics of Network Reliability*, Oxford Univ. Press, 1987.

[6] A.G. Greenberg and J. Goodman, "Sharp Approximate Models of Adaptive Routing in Mesh Networks," *Teletraffic Analysis and Computer Performance Evaluation*, pp. 255-270, Elsevier, 1986.

[7] A.G. Greenberg and B. Hajek, "Deflection Routing in Hypercube Networks," *IEEE Trans. Comm.*, vol. 35, no. 6, pp. 1,070-1,081, June 1992.

[8] A. Krishna, "Communication with Few Buffers: Analysis and Design," PhD thesis, Dept. of Electrical and Computer Eng., Univ. of Illinois at Urbana-Champaign, Dec. 1990.

[9] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays—Trees—Hypercubes*. San Mateo, Calif.: Morgan Kaufmann, 1992.

[10] Y. Lan, A.-H. Esfahanian, and L. Ni, "Multicast in Hypercube Multiprocessors," *J. Parallel and Distributed Computing*, pp. 30-41, 1990.

[11] N.F. Maxemchuk, "Comparison of Deflection and Store-and-Forward Techniques in the Manhattan Street and Shuffle-Exchange Networks," *Proc. INFOCOM '89*, vol. 3, pp. 800-809, Apr. 1989.

[12] C.J.A. Nash-Williams, "Edge-Disjoint Spanning Trees of Finite Graphs," *J. London Math. Soc.*, vol. 36, 1961.

[13] Y. Shiloach, "Edge-Disjoint Branching in Directed Multigraphs," *Information Processing Letters*, 1979.

[14] G.D. Stamoulis and J.N. Tsitsiklis, "Efficient Routing Schemes for Multiple Broadcasts in Hypercubes," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 7, pp. 725-439, July 1993.

[15] G.D. Stamoulis and J.N. Tsitsiklis, "Greedy Routing in Hupercubes and Butterflies," *IEEE Trans. Comm.*, vol. 44, no. 11, pp. 3,051-3,061, 1994.

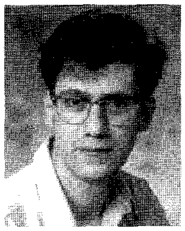
[16] R.E. Tarjan, "A Good Algorithm for Edge-Disjoint Branching," *Information Processing Letters*, 1974.

[17] E.A. Varvarigos and D.P. Bertsekas, "Multinode Broadcast in Hypercubes and Rings with Randomly Distributed Length of Packets," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, pp. 144-154, 1993.

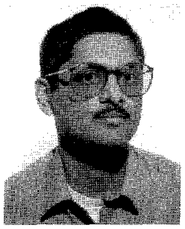
[18] E.A. Varvarigos and D.P. Bertsekas, "Partial Multinode Broadcast and Partial Exchange in d-Dimensional Meshes," *J. Parallel and Distributed Computing*, vol. 23, pp. 177-189, 1994.

[19] E.A. Varvarigos and D.P. Bertsekas, "Performance of Hypercube Routing Schemes With or Without Buffering," *IEEE/ACM Trans. Networking*, vol. 2, no. 3, pp. 299-311, June 1994.

[20] E.A. Varvarigos and D.P. Bertsekas, "Dynamic Broadcasting in Parallel Computing" *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 2, pp. 120-131, Feb. 1995.



Emmanouel A. (Manos) Varvarigos received a Diploma (1988) in electrical engineering from the National Technical University of Athens, Greece, and the MS (1990), Electrical Engineer (1991), and PhD (1992) degrees in electrical engineering and computer science from the Massachusetts Institute of Technology. In 1990, he worked as a researcher at Bell Communications Research, Morristown, New Jersey. Since 1992, he has been an assistant professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. His research interests are in the areas of parallel and distributed computation, protocols for multigigabit networks, performance evaluation, and protocols for mobile communications. Dr. Varvarigos received the first panhellenic prize in the Greek Mathematic Olympiad in 1982, and the Technical Chamber of Greece award four times (1984-1988). He is a member of the Technical Chamber of Greece.



Ayan Banerjee received the BTech (Hons) degree in electronics and electrical communication engineering in 1992 from the Indian Institute of Technology, Kharagpur, India. He received his MS in electrical and computer engineering in 1993 from the University of California, Santa Barbara, and is currently a PhD candidate there. His research interests include parallel computer architecture, with emphasis on routing algorithms and communication protocols, data networks, and wireless communications.