The Efficiency of Greedy Routing in Hypercubes and Butterflies

George D. Stamoulis and John N. Tsitsiklis, *Member, IEEE*

***Abstract*— We analyze the following problem: Each node of the d-dimensional hypercube independently generates packets according to a Poisson process with rate λ. Each of the packets is to be sent to a randomly chosen destination; each of the nodes at Hamming distance k from a packet's origin is assigned an *a priori* probability $p^k(1-p)^{d-k}$. Packets are routed under a simple greedy scheme: each of them is forced to cross the hypercube dimensions required in increasing index-order, with possible queueing at the hypercube nodes. Assuming unit packet length and no other communications taking place, we show that this scheme is stable (in steady-state) if $\rho < 1$, where $\rho \overset{\text{def}}{=} \lambda p$ is the load factor of the network; this is seen to be the broadest possible range for stability. Furthermore, we prove that the average delay T per packet satisfies $T \leq \frac{dp}{1-\rho}$, thus showing that an average delay of $\Theta(d)$ is attainable for any fixed $\rho < 1$. We also establish similar results in the context of the butterfly network. Our analysis is based on a stochastic comparison with a product-form queueing network.**

I. Introduction

A. Problem Definition—Summary of the Results

During the execution of parallel algorithms in a network of processors, it is necessary that processors communicate with each other in order to exchange information. This is accomplished by routing messages through the underlying interconnection network. In the present paper, we consider a problem that arises in this context: the nodes (processors) of a hypercube network generate packets at random time instants; each packet has a single destination, which is selected at random. We discuss a simple greedy scheme for routing these packets and we analyze its steady-state stability and delay properties. The results to be derived extend to the butterfly network.

We consider the d-dimensional *binary hypercube* (or d-cube); e.g., see [2]. This network consists of 2^d nodes, numbered from 0 to $2^d - 1$. Associated with each node z is a binary identity $(z_d, \ldots, z_1)$, which coincides with the binary representation of the number z. For $j \in \{1, \ldots, d\}$, we denote by e_j the node numbered 2^{j-1}; that is, all entries of the binary identity of e_j equal 0 except for the jth one (from the right), which equals 1. For two nodes z and y, we denote by $z \oplus y$ the vector $(z_d \oplus y_d, \ldots, z_1 \oplus y_1)$, where $\oplus$ is the symbol for the XOR operation. The d-cube has $d2^d$ arcs; each arc is directed and connects two nodes whose binary identities differ in a single bit; see Fig. 1(a), where the 3-cube is depicted. That is, arc (z, y) exists if and only if, for some $m \in \{1, \ldots, d\}$, $z_i = y_i$ for $i \neq m$ and $z_m \neq y_m$; this is equivalent to $y = z \oplus e_m$ for some $m \in \{1, \ldots, d\}$. Such an arc is said to be of the mth *type*; the set of arcs of the mth type is called the mth *dimension.* Note that (z, y) stands for a *unidirectional* arc pointing from z to y; of course, if arc (z, y) exists, so does arc (y, z). The *Hamming distance* between two nodes z and y is defined as the number of bits in which their binary identities differ; it is denoted by $H(z, y)$. Any path from z to y contains at least as many arcs as the Hamming distance between z and y. Moreover, there always exist paths that contain exactly that many arcs; these paths are *shortest*. It is easily seen that the *diameter* of the d-cube equals d.

The underlying assumptions for communications are as follows: Each piece of information is transmitted as a packet with unit transmission time. Only one packet can traverse an arc at a time; all transmissions are error-free. Each node may transmit packets through all of its output ports and at the same time receive packets through all of its input ports. Each node has infinite buffer capacity. Finally, for analytical convenience, the time axis is taken to be continuous. (Our results can be easily extended to the slotted case; see [17].)

In many of the routing problems that are discussed in the literature, there is a finite set of packets to be routed to their destinations, and all packets are assumed to be available at time zero; these are *static* routing problems. In contrast, in this paper, we assume that new packets are generated at random times over an infinite time horizon; problems of this type are called *dynamic*. We assume that each node of the d-cube generates packets according to a Poisson process with rate λ; different nodes generate their packets independently of each other. Each packet has a *single* destination, which is selected *randomly* according to the following probability distribution:

$$\Pr[\text{a packet generated by node } x \text{ is destined for node } z] = p^{H(x,z)}(1-p)^{d-H(x,z)}, \qquad (1)$$

where $p \in (0, 1]$; different packets make their selections independently of each other.

Notice that the problem just defined is *invariant under translation*; that is, if each hypercube node is renamed from x to $x \oplus y^*$ (where y^* is a fixed d-bit string), then the statistics of the various random variables are not affected.

It is seen from (1) that for $p = \frac{1}{2}$ the destination distribution is uniform; that is, each node (including its origin) is equally

Paper approved by J. F. Kurose, the Editor for Networking Systems of the IEEE Communications Society. Manuscript received October 26, 1990; revised March 8, 1993. This work was supported by the NSF under Grant ECS-8552419 and by the ARO under Grant DAAL03-86-K0171.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

IEEE Log Number 9401953.

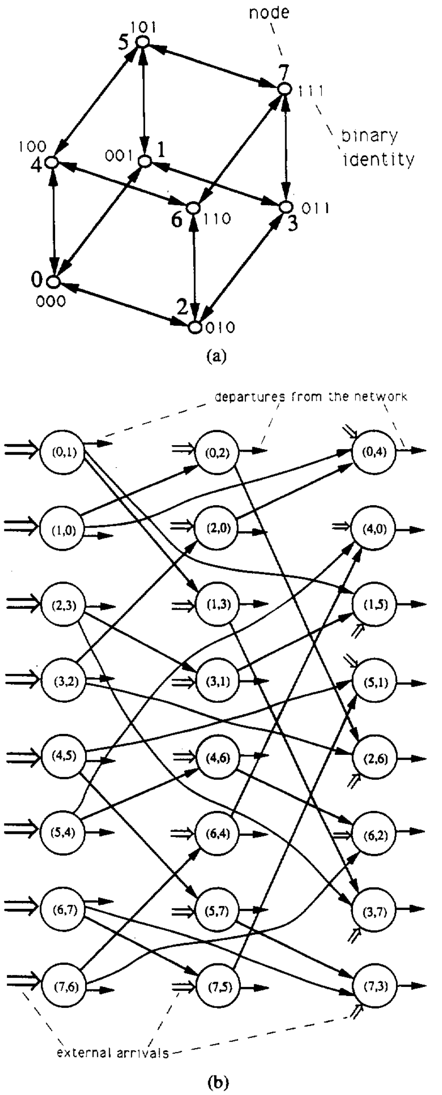


Fig. 1. (a) The three-dimensional hypercube. (b) The equivalent network $\mathcal{Q}$ for the three-dimensional hypercube.

likely to be chosen as a packet's destination. This is the case usually considered in the literature (see Section I–B); in most of the related works, a packet's origin is not a permissible destination; however, it is easily seen that our results (when rescaled appropriately) also apply to this case. Also note that for $p < \frac{1}{2}$ the destination distribution favors nodes at shorter distance from a packet's origin; in this case, packet transmissions tend to be more localized.

As will be proved in Section II–A, the inequality

$$\rho \overset{\text{def}}{=} \lambda p < 1$$

is a *necessary* condition for *stability*; ρ will be called the *load factor* of the system. Therefore, it is of particular interest to devise a routing scheme that is guaranteed to be stable for all $\rho < 1$. Moreover, it is desirable that such a scheme does not introduce excessive *delay*; a reasonable delay objective is to require that for every $\rho < 1$, the average delay is of the order of d. It is plausible that these objectives might be attained if we let each packet choose a shortest path leading to its destination and attempt to traverse this path as fast as possible. However, the performance of such *greedy* schemes has not been analyzed rigorously in the literature. In this paper, we prove that the following greedy scheme has the desired properties: consider a packet originating at node x and destined for node z; this packet will be routed through that shortest path (from x to z) in which the hypercube dimensions are crossed in *increasing* index-order. (Such paths are often referred to as canonical.) For example, a packet travelling from node $(0,0,0,0)$ to node $(1,0,1,1)$ in the 4-cube would follow the path

$$(0,0,0,0) \to (0,0,0,1) \to (0,0,1,1) \to (1,0,1,1).$$

It will be proved that this simple routing scheme is stable for all $\rho < 1$, which is the broadest possible stability region. By the term "stable" it is meant that the time spent by the nth packet in the system converges in distribution (as $n \to \infty$) to a limiting random variable, which is finite with probability 1. Moreover, it will be established that, for $\rho < 1$, the delay T induced by the scheme satisfies

$$dp + p\frac{\rho}{2(1-\rho)} \le T \le \frac{dp}{1-\rho};$$

of particular interest is the upper bound on the delay, which guarantees that, for any fixed ρ, each packet reaches its destination in an average time $\Theta(d)$. Notice also that under heavy traffic (i.e., for $\rho \to 1$) the delay T increases as $\frac{1}{1-\rho}$. It will be established that such a behavior under heavy traffic is optimal for any fixed d; indeed, it will be proved that $\lim_{\rho \to 1}[(1-\rho)T] > 0$ under *any* legitimate routing scheme.

The results above may be easily extended to the d-dimensional *butterfly*, a switching network that can be viewed as an "unfolded" version of the d-cube; see Section IV–A and [2]. In this context, it is assumed that packets are generated at one of the fronts of the butterfly and destined for a randomly chosen node at the opposite front; the destination distribution is identical to that presented in (1), except for the fact that x and z belong to opposite fronts of the butterfly. Notice that crossing the dimensions in increasing index-order is the only legitimate choice of paths for the butterfly. Thus, the scheme simply reduces to greedy routing; this will be seen to be stable for all $\rho < 1$, where ρ is now defined as $\rho \overset{\text{def}}{=} \lambda \max\{p, (1-p)\}$; moreover, for $\rho < 1$, the average delay T satisfies

$$d + p\frac{\lambda p}{2(1-\lambda p)} + (1-p)\frac{\lambda(1-p)}{2[1-\lambda(1-p)]} \le T \le \frac{dp}{1-\lambda p} + \frac{d(1-p)}{1-\lambda(1-p)}.$$

Again, the delay T is $\Theta(d)$ for any fixed $\rho < 1$, which is the optimal order of magnitude; also, the behavior of T under heavy traffic will be seen to be optimal, for any fixed d.

To the best of our knowledge, these results are new. Moreover, our analysis provides the first proof that some routing scheme (on either the d-cube or the butterfly) is stable for all $\rho < 1$ while satisfying the requirement for $\Theta(d)$ average delay; proving that greedy routing has these properties has been a long-standing open question in the routing literature. Also, this is the first routing scheme for which the bounds on the delay are expressed in simple formulae involving the system's parameters ρ and d. Finally, the approach for deriving the aforementioned results is new as well: it is established that the hypercube (resp., the butterfly) behaves as a queueing network with deterministic servers (each corresponding to an arc) and with Markovian routing among the various servers; then, by using sample path arguments, it is shown that the delay induced by this queueing network is dominated by that corresponding to a product-form network. This kind of approach relies on the assumption of Poisson arrivals; nevertheless, we hope that our analysis will be suggestive of the efficient perfomance of greedy routing under more general packet-generating processes; in fact, the conditions for stability derived in our analysis are much more general.

B. Survey of Previous Work

There exists a considerable literature on hypercube routing, especially for static problems; see [18], [19], [15], [7], [20], [11], [2], [4] and the references therein. Reference [7] also contains a scheme for routing continuously batches of permutations, by pipelining. This leads to a scheme that can be applied to the problem studied in this paper; however, such a scheme would be stable only for quite small values of the load factor and would not satisfy our desire to have stability for every ρ less than 1.

The dynamic routing problem of this paper has been dealt with in several articles, which we discuss below; all of them assume that the destination distribution is uniform. Abraham and Padmanabhan [1] have constructed an approximate model for this problem, under various assumptions on the buffer capacity of the nodes. In particular, they assume that packets advance in the respective paths independently of each other; the model involves some parameters, which are determined by solving a system of non-linear equations. Greenberg and Hajek [9] have provided an approximate analysis for the case of deflection routing. Greenberg and Goodman [8] study the case of a square mesh. More recently, Leighton [13] proved that greedy routing in the square mesh has very satisfactory average performance. Bouras *et al.* [3] considered the same problem in the context of Banyan networks; however, we are unable to follow some of the steps in the analysis therein. Mitra and Cieslak [14] and Hajek and Cruz [10] have dealt with similar problems in the context of the extended Omega network; the analysis is again approximate and is based on "Kleinrock's independence assumption." Finally, another dynamic routing problem, was analyzed by Stamoulis and Tsitsiklis in [16], where it was assumed that packets generated at random instants and at random nodes of the hypercube must be broadcast to all nodes.

II. Preliminary Results for the Hypercube

A. The Necessary Condition for Stability

We start with an observation to be used several times in the analysis. Consider a fixed packet $\mathcal{P}$ generated at node x. Let $\mathcal{B}_i$ denote the event that packet $\mathcal{P}$ will choose a destination z such that $z_i \neq x_i$; notice that if event $\mathcal{B}_i$ occurs, then $\mathcal{P}$ will have to cross an arc of the ith dimension in order to reach its destination. It is a straightforward consequence of the definition of the destination distribution [see (1)] that the following is true:

Lemma 1: For any fixed packet $\mathcal{P}$, events $\mathcal{B}_1, \ldots, \mathcal{B}_d$ are mutually independent, with $\Pr[\mathcal{B}_i] = p$ for $i = 1, \ldots, d$. Independence prevails both with and without conditioning on the origin of the packet. □

Lemma 1 essentially implies the following: In order to choose the binary identity of its destination, packet $\mathcal{P}$ flips each of the bits of the identity of its origin x; each bit-flip is performed with probability p, *independently* of the others. Notice also that the average number of bit-flips performed equals dp; therefore, under any routing scheme, each packet will have to traverse at least dp hypercube arcs on the average.

Next, we derive the necessary condition for stability. The average total number of packets generated in the network per unit time equals $\lambda 2^d$. Thus, by the conclusion of the previous paragraph, it is seen that during each time unit an average total demand for at least $\lambda 2^d dp$ packet transmissions is generated in the system. Since at most $d2^d$ packet transmissions may take place per unit time, it follows that the system can be stable only if $\lambda 2^d dp \leq d2^d$. Thus, we obtain the following necessary condition for stability under any routing scheme:

$$\rho \overset{\text{def}}{=} \lambda p \leq 1, \tag{2}$$

where ρ will be called the *load factor* of the system. This terminology is appropriate, because when $\rho \approx 1$ all hypercube arcs are almost always busy, even if no redundant packet transmissions take place. Notice that (2) is a necessary condition for stability under more general arrival processes. Furthermore, this condition can be strengthened to $\rho < 1$, unless all arrival processes are deterministic.

B. Lower Bounds on the Delay

First, we establish a *universal* lower bound on the steady-state average delay T per packet; that is, a bound that applies to *any* routing scheme. Recall that T is defined as the stationary average of the time elapsing between the moment a packet is generated until it reaches its destination.

Proposition 2: The average delay T per packet induced by any routing scheme satisfies

$$\begin{aligned} T &\geq max\{dp, p\mathcal{D}(2^d;\rho)\} \\ &= \Omega\left(dp + p\frac{\rho}{2^d(1-\rho)}\right), \qquad \forall \rho < 1, \end{aligned}$$

where $\mathcal{D}(2^d;\rho)$ is the average delay for the $M/D/2^d$ queue with unit service time and arrival rate $2^d\rho$. □

Proof: Consider a fixed packet $\mathcal{P}$ generated at node x; if its random destination satisfies $z_1 \neq x_1$ (that is, if event $\mathcal{B}_1$ occurs for $\mathcal{P}$), then $\mathcal{P}$ will not reach its destination until it traverses at least one arc of the 1st dimension. Let W be the average time until a packet crosses the 1st dimension, with the convention that packets that never do so contribute zero to this average; clearly, $T \geq W$. It is straightforward to see that the value of W can only decrease if we introduce the following conditions:

(a) Each packet for which event $\mathcal{B}_1$ has not occured never crosses the 1st dimension.

(b) Each packet for which event $\mathcal{B}_1$ has occured is available upon its generation at all nodes; moreover, such a packet will only cross the first available arc of type 1.

Under these assumptions, the 2^d arcs of the first dimension operate as an $M/D/2^d$ queue. The input stream of this queue consists of all packets for which event $\mathcal{B}_1$ occurs; by Lemma 1, this stream is Poisson with rate $\lambda 2^d p = 2^d\rho$. The average delay induced by this queue equals $\mathcal{D}(2^d;\rho)$; since only a fraction p of the packets "joins" this $M/D/2^d$ queue, we have

$$W \geq p\mathcal{D}(2^d;\rho). \tag{3}$$

Recall now that $T \geq W$ and $T \geq dp$ (see Sectiom II-A); these facts together with (3) imply that

$$T \geq \max\{dp, p\mathcal{D}(2^d;\rho)\}. \tag{4}$$

Furthermore, it is known [6] that

$$\mathcal{D}(2^d;\rho) \geq 1 + \frac{\rho}{2^{d+1}(1-\rho)};$$

combining this with (4), it follows that

$$\begin{aligned} T &= \Omega\left(\max\left\{dp, p + p\frac{\rho}{2^{d+1}(1-\rho)}\right\}\right) \\ &= \Omega\left(dp + p\frac{\rho}{2^d(1-\rho)}\right), \end{aligned}$$

where we have also used the inequality $\max\{\alpha_1,\alpha_2\} \geq \frac{1}{2}(\alpha_1+\alpha_2)$. The proof of the result is now complete. *Q.E.D.*

The universal lower bound of Proposition 2 shows that $\lim_{\rho\to 1}[(1-\rho)T] > 0$, for any fixed d, under any routing scheme. As far as asymptotics with respect to d are concerned, the bound appears to be loose, due to the presence of the factor $\frac{1}{2^d}$. Below, we establish a sharper lower bound applying to a restricted but fairly broad class of routing schemes.

As suggested by the proof of Proposition 2, a scheme that comes close to attaining the universal lower bound for the delay T (if there exists such a scheme) would schedule transmissions *adaptively*. This claim is further supported by Proposition 3, which establishes a lower bound on T under *oblivious* schemes. Under an oblivious scheme, each packet selects its path, possibly using randomization, independently of the existing traffic and insists on traversing the selected path (see [5]); we also assume that all rules for path selection are time-independent.

Proposition 3: The average delay T per packet induced by any *oblivious* routing scheme satisfies

$$T = \Omega\left(dp + p\frac{\rho}{1-\rho}\right). \quad □$$

Proof: This proof is similar to that of Proposition 2. We consider a node x and an arc $(y, y\oplus e_1)$; under any oblivious scheme, the following is true: for each packet generated at x, the event that arc $(y, y\oplus e_1)$ is the *first* arc of type 1 to be crossed by such a packet is independent of any events involving other packets. Let $q_{x,y}$ be the probability of the event that was just described. Then,

$$\sum_{y=0}^{2^d-1} q_{x,y} \geq p, \qquad \forall x \in \{0,\ldots,2^d-1\}, \tag{5}$$

because it is with probability p that some packet generated by node x will necessarily cross an arc of the 1st dimension. Let W be the average time until a packet crosses the 1st dimension for the first time, with the convention that packets that never do so contribute zero to this average; clearly, $T \geq W$. For any oblivious routing scheme, the value of W can only decrease if we introduce the following conditions:

(a) Each packet to cross the 1st dimension is only delayed at the first time it does so.

(b) Each packet to cross arc $(y, y\oplus e_1)$ is available at node y upon its generation.

Under these conditions, each arc $(y, y\oplus e_1)$ is fed by a group of 2^d Poisson streams. We denote by r_y the total arrival rate of the compound Poisson stream; obviously, we have

$$r_y = \lambda\sum_{x=0}^{2^d-1} q_{x,y}, \qquad \forall y \in \{0,\ldots,2^d-1\}. \tag{6}$$

Clearly, arc $(y, y\oplus e_1)$ behaves as an $M/D/1$ queue with unit service time. Therefore (see [12]), the average delay W_y per packet joining this queue is given as follows:

$$W_y = 1 + \frac{r_y}{2(1-r_y)}.$$

Using this, we obtain

$$W \geq \frac{1}{\lambda 2^d}\sum_{y=0}^{2^d-1} r_y W_y = \frac{1}{\lambda 2^d}\sum_{y=0}^{2^d-1} r_y\left[1 + \frac{r_y}{2(1-r_y)}\right]. \tag{7}$$

Combining (5) and (6), we have

$$\sum_{y=0}^{2^d-1} r_y \geq \lambda 2^d p. \tag{8}$$

Notice now that $r[1 + \frac{r}{2(1-r)}]$ is a convex and increasing function of r; therefore, in light of (8), the right-hand quantity in (7) is minimized when $r_y = \lambda p$ for all $y \in \{0,\ldots,2^d-1\}$. Thus, it follows that

$$W \geq \frac{1}{\lambda 2^d}\sum_{y=0}^{2^d-1} \lambda p\left[1 + \frac{\lambda p}{2(1-\lambda p)}\right] = p\left[1 + \frac{\rho}{2(1-\rho)}\right].$$

This together with the facts $T \geq W$ and $T \geq dp$ proves that

$$T \geq \max\left\{dp, p\left[1 + \frac{\rho}{2(1-\rho)}\right]\right\},$$

and the result follows. Q.E.D.

III. The Main Results for the Hypercube

In this section, we analyze an efficient greedy routing scheme for the hypercube network. As already mentioned in Section I-A., the scheme is as follows: Each packet proceeds towards its destination by crossing the dimensions required in increasing index-order. To clarify matters, consider a packet $\mathcal{P}$ generated at node x and destined for node z; let $i_1, \ldots, i_k$ be the entries in which the binary identities of x and z differ, with $i_1 < i_2 < \cdots < i_k$; then, packet $\mathcal{P}$ follows the path

$$x \to x \oplus e_{i_1} \to x \oplus e_{i_1} \oplus e_{i_2} \to \cdots \to x \oplus e_{i_1} \oplus \cdots \oplus e_{i_k} = z.$$

Whenever several packets present at a node y wish to traverse the same arc, then priority is given to the one that arrived at y the first. Note that this scheme is oblivious.

It will be seen in Section III-A that, under this scheme, the hypercube is equivalent to a queueing network with certain useful properties. The analysis in Sections III-B and -C deals with the performance of this equivalent queueing network.

A. The Equivalent Queueing Network

It is straightforward that, under our routing scheme, the d-cube may be viewed as a queueing network, with $d2^d$ deterministic FIFO "servers"; each "server" has unit service duration and corresponds to a hypercube *arc*. This equivalent queueing network (to be referred to as $\mathcal{Q}$) has the following properties (see Fig. 1(b) for an illustration):

Property A: The external arrival stream at any arc $(x, x \oplus e_i)$ is Poisson with rate $\lambda p(1-p)^{i-1}$; streams corresponding to different arcs are mutually independent.

To see this, consider a packet $\mathcal{P}$ generated at node x of the d-cube; with probability $p(1-p)^{i-1}$ the destination of $\mathcal{P}$ satisfies $z_1 = x_1, \ldots, z_{i-1} = x_{i-1}$ and $z_i \neq x_i$ (see Lemma 1). Since packets cross the hypercube dimensions in increasing index-order, it follows that each of the packets generated by node x will join the queue for arc $(x, x \oplus e_i)$ with probability $p(1-p)^{i-1}$.

Property B: After crossing arc $(y, y \oplus e_i)$, a packet will *never* traverse again an arc $(z, z \oplus e_j)$ with $j \in \{1, \ldots, i\}$. Thus, the equivalent network $\mathcal{Q}$ is a *layered* network; that is, its "servers" are organized in d levels, with the ith level comprising all arcs $(y, y \oplus e_i)$ for $y \in \{0, \ldots, 2^d - 1\}$, i.e. all arcs of the ith dimension. Upon "service completion" at a certain level, a packet either joins a queue at a higher level or it departs from the network.

Property C: Routing is *Markovian.* In particular, upon crossing arc $(y, y \oplus e_i)$, a packet takes one of the following actions: either it joins the queue at arc $(y \oplus e_i, y \oplus e_i \oplus e_j)$ with probability $p(1-p)^{j-i-1}$ for $j = i+1, \ldots, d$; or it departs from the network with probability $(1-p)^{d-i}$. After crossing arc $(y, y \oplus e_d)$, a packet departs from the network with probability 1. Different packets take their routing decisions independently of each other.

The validity of property C can be easily seen if we visualize a packet's propagation through the network as follows. Upon generation, the packet decides whether or not to cross dimension 1; the probability that it decides positively equals p. If it does so, then it takes its step on this dimension and *then* it decides whether or not to cross dimension 2; if it does not decide to cross dimension 1, then it considers crossing dimension 2, etc.

B. Stability for $\rho < 1$

In the previous subsection, we established that, under the routing scheme analyzed, the hypercube is equivalent to a queueing network $\mathcal{Q}$ with Markovian routing. In this subsection, we derive a sufficient condition for stability of the routing scheme. First, we prove the following result:

Lemma 4: The total arrival rate at any arc of the d-cube equals $\lambda p = \rho$. □

Proof: By symmetry among the hypercube nodes, all arcs belonging to the same dimension j have the same total arrival rate θ_j. Furthermore, the total arrival rate for the jth dimension equals $2^d \lambda p$, because each of the packets generated within the d-cube crosses the jth dimension for an expected number of p times. Hence, we have $2^d \theta_j = 2^d \lambda p$, which gives $\theta_j = \lambda p = \rho$ for all $j \in \{1, \ldots, d\}$. *Q.E.D.*

Notice now that the equivalent network $\mathcal{Q}$ has the following properties:

(a) $\mathcal{Q}$ is *acyclic* (Property B).

(b) Each "server" is fed externally by a Poisson process. Arrival processes corresponding to different "servers" are independent. (Property A.)

(c) Service times corresponding to different packets and/or different "servers" are (trivially) independent.

(d) Routing is Markovian (Property C).

These properties allow us to apply a result on the stability of acyclic networks; see [21, p. 246]. It thus follows that network $\mathcal{Q}$ is *stable* if the total arrival rate for each "server" is less than unity. By stability it is meant that the time spent by the nth packet in the system converges in distribution (as $n \to \infty$) to a limiting random variable, which is finite with probability 1 and independent of the initial state. Recalling the equivalence of $\mathcal{Q}$ with the hypercube (under our greedy routing scheme) and using Lemma 4, we reach the following conclusion:

Proposition 5: The greedy routing scheme under analysis is stable for all $\rho < 1$. □

In light of the necessary condition for stability $\rho < 1$ (see Section II-A), it is seen that the routing scheme under analysis has *optimal* stability properties. In fact, the stability result of [21] applies to more general arrival processes and so does Proposition 5.

C. Delay Bounds

In this subsection, we establish upper and lower bounds for the average delay T induced by the routing scheme under analysis. Starting with the upper bound (which is the most interesting result), we will show that $T \leq \frac{dp}{1-\rho}$ for all $\rho < 1$. The basic idea for proving this result is as follows:

If the service discipline at the "servers" of the equivalent network $\mathcal{Q}$ is *changed* from FIFO to *Processor Sharing* (PS), then the average delay per packet *increases*; under the PS discipline, $\mathcal{Q}$ becomes a *product-form* network, and its delay is easily computed.

Recall that under the PS discipline all customers present at a server receive an equal proportion of service *simultaneously*; see [21, p. 354]. For example, consider a deterministic PS server, with unit service rate; assume that it has two customers to serve, with the first customer arriving at time 0 and the second at time $\frac{1}{4}$; upon arrival of the second customer, the first one has $\frac{3}{4}$ units of service remaining; however, due to the presence of the second customer, she will be served at rate $\frac{1}{2}$; thus, she will depart at time $\frac{1}{4} + 2\frac{3}{4} = \frac{7}{4}$; similarly, it can be seen that the second customer will depart at time 2. Notice that we are using the term "service rate" for a PS server (rather than the term "service duration"), because the time duration for which a customer receives service depends on previous and future arrivals.

The proof of the upper bound on the delay T makes use of several lemmas that establish sample path results; these we present next. Throughout, we assume that the network starts empty at time 0.

Lemma 6: Consider a deterministic FIFO server with unit service duration. For a fixed sequence $t_1, t_2, \ldots$ of arrival times, let $D_1, D_2, \ldots$ denote the corresponding sequence of departure times. Similarly, let $\tilde{D}_1, \tilde{D}_2, \ldots$ be the departure times for a deterministic PS server, with unit service rate, fed by the same input stream. There holds

$$D_i \leq \tilde{D}_i, \text{ for } i = 1, \ldots \qquad \square$$

Proof: Clearly, we have $D_1 = t_1 + 1$. In the context of the PS server, the 1st customer will depart at time t_1+1 only if no other customers arrive until that time; otherwise, the server will be slowed down, and the 1st customer will depart later than $t_1 + 1$. It follows that

$$\tilde{D}_1 \geq t_1 + 1 = D_1. \tag{9}$$

It is well-known that the PS discipline is *work-conserving*; see [21, pp. 353–354. That is, the unfinished work $W(t)$ at time t is the *same* for both the FIFO and the PS servers considered. By definition of $W(t)$, we have

$$D_i = t_i + W(t_i-) + 1, \text{ for } i = 1, \ldots \tag{10}$$

We now consider the ith arrival at the PS server, where $i \geq 2$. If $W(t_i-) = 0$, then reasoning similarly as in proving (9), it follows that $\tilde{D}_i \geq t_i+1 = D_i$. Assume now that $W(t_i-) \neq 0$; it is straightforward that customers depart from a deterministic PS server in the order they arrive; hence, the ith customer may depart only after an amount $W(t_i-) + 1$ of work has been finished by the server. Therefore, we have

$$\tilde{D}_i \geq t_i + W(t_i-) + 1 = D_i,$$

where we have also used (10). The proof of the lemma is now complete. *Q.E.D.*

Let there be two streams of events, one occurring at times $\tau_1, \tau_2, \ldots$ and the other at times $\tau'_1, \tau'_2, \ldots$ If $\tau_i \leq \tau'_i$ for $i = 1, \ldots$, then the latter stream of events will be said to be a *delayed version* of the former. For example, as implied by Lemma 6, for any fixed arrival stream, the departing stream of a deterministic PS server is a delayed version of the one of the corresponding FIFO server.

Lemma 7: Let there be a deterministic FIFO server with unit service duration. Let $D_1, D_2, \ldots$ (resp. $D'_1, D'_2, \ldots$) be the sequence of departure times corresponding to a fixed sequence $t_1, t_2, \ldots$ (resp. $t'_1, t'_2, \ldots$) of arrival times. If $t_i \leq t'_i$ for $i = 1, \ldots$, then

$$D_i \leq D'_i, \quad \text{for } i = 1, \ldots \qquad \square$$

Proof: There holds

$$D_1 = t_1 + 1 \text{ and } D_i = \max\{D_{i-1}, t_i\} + 1 \text{ for } i = 2, \ldots;$$

similarly,

$$D'_1 = t'_1 + 1 \text{ and } D'_i = \max\{D'_{i-1}, t'_i\} + 1 \text{ for } i = 2, \ldots$$

Using these facts and the assumption $t_i \leq t'_i$ for $i = 1, \ldots$, the result follows by a straightforward inductive argument. Q.E.D.

The result to be established next is based on Lemmas 6 and 7; generalizing this result will lead to the upper bound on the delay induced by our greedy routing scheme. We consider the queueing network $\mathcal{G}$ depicted in Fig. 2(a). This consists of three deterministic FIFO servers with unit service duration, denoted by S_1, S_2 and S_3. Customers completing service at S_1 or S_2 either depart from the network or they join the queue at S_3; routing decisions are Markovian. Obviously, $\mathcal{G}$ is a *layered* network (see Section III–A). We define a *sample path* ω of $\mathcal{G}$ as the following collection of information:

(a) The *external* arrival times at servers S_1, S_2 and S_3.
(b) The routing decision taken by the ith customer upon service completion at S_1 (resp. S_2) for $i = 1, \ldots$

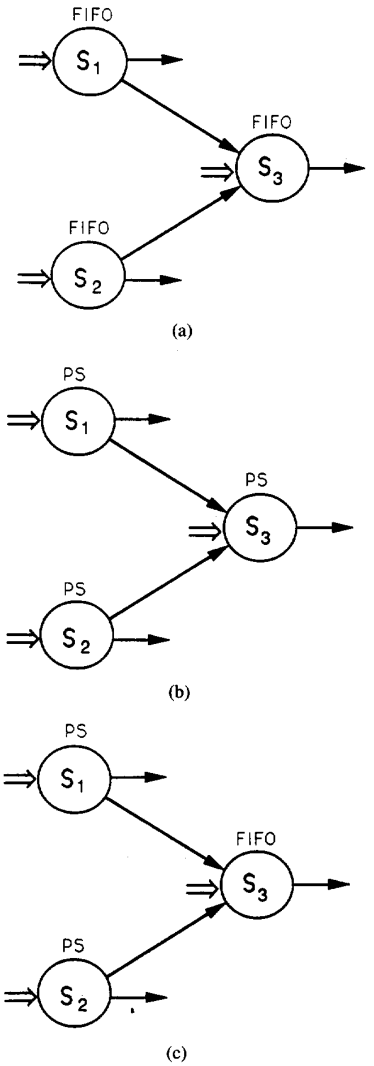
Clearly, given a sample path ω, network $\mathcal{G}$ evolves in a *deterministic* fashion. The result to be proved is as follows.

Lemma 8: Let $\tilde{\mathcal{G}}$ be a network identical to $\mathcal{G}$ except for the fact that PS service discipline applies for the servers of $\tilde{\mathcal{G}}$ (instead of FIFO); see Fig. 2(b). For a particular sample path ω, let $B(t)$ [resp. $\tilde{B}(t)$] denote the number of customers departing from $\mathcal{G}$ (resp. $\tilde{\mathcal{G}}$) during the interval $[0, t]$; there holds

$$B(t) \geq \tilde{B}(t), \qquad \forall t \geq 0. \qquad \square$$

Proof: First, we consider a network $\mathcal{G}'$ obtained from $\mathcal{G}$ by changing the service discipline *only* at S_1 and S_2 (from FIFO to PS); see Fig. 2(c).

We define as the *output stream* of a server the stream of customers completing service therein, including those that do not depart from the network. Notice that server S_1 is not affected at all by the presence of the other two servers; the same statement applies for server S_2. Therefore, applying Lemma 6, it is seen that the output stream of server S_1 in $\mathcal{G}'$ is a delayed version of that corresponding to S_1 of $\mathcal{G}$. Recalling also that the routing decisions of customers completing service are the same for networks $\mathcal{G}$ and $\mathcal{G}'$, it follows that the substream of customers departing from $\mathcal{G}'$ at



(a)

(b)

(c)

Fig. 2. (a) Network $\mathcal{G}$. (b) Network $\tilde{\mathcal{G}}$. (c) Network $\mathcal{G}'$.

S_1 is a delayed version of the corresponding substream in $\mathcal{G}$. Similar statements apply for the streams stemming from S_2.

Next, we consider the stream feeding S_3 in $\mathcal{G}'$; this stream is a delayed version of that feeding S_3 in $\mathcal{G}$, because each arrival at S_3 of $\mathcal{G}'$ corresponds to an arrival at S_3 of $\mathcal{G}$ that occurs *no later*. [Recall the aforementioned "comparison" of the output streams of S_1 (resp. S_2) in the two networks and the coupling of the routing decisions.] Therefore, applying Lemma 7, the output stream from S_3 of $\mathcal{G}'$ is a delayed version of that corresponding to S_3 of $\mathcal{G}$. The former output stream is delayed *further* when the service discipline at S_3 of $\mathcal{G}'$ is changed from FIFO to PS. This modification (which yields network $\tilde{\mathcal{G}}$) does not affect the streams of customers departing from the 1st level. Therefore, for each of the servers of $\tilde{\mathcal{G}}$, its departing stream is a delayed version of that of the corresponding server of $\mathcal{G}$; this proves the result in question.

Its should be noted that customers joining S_3 may get out of order when changing the service discipline; thus, a *particular* customer may depart earlier from $\tilde{\mathcal{G}}$ than from $\mathcal{G}$. Nevertheless, this does not affect the validity of the lemma. Q.E.D.

Next, we generalize Lemma 8. In the context of the network $\mathcal{Q}$, a sample path ω is defined as the collection of information comprising all external arrival times and all routing decisions. Notice that routing decisions at each "server" are identified by the *order* they are taken, not by the identity of the packets deciding; e.g., "the 1st packet to cross arc $(e_1 \oplus e_2, e_1)$ will advance to $(e_1, e_1 \oplus e_3)$, the second such packet will depart," etc. Such an identification of the routing decisions is legitimate due to the fact that routing in $\mathcal{Q}$ is Markovian. Similarly as in Lemma 8, we denote as $\tilde{\mathcal{Q}}$ the network obtained from $\mathcal{Q}$ after changing the service discipline of all "servers" from FIFO to PS.

Lemma 9: For a particular sample path ω, let $B(t)$[resp. $\tilde{B}(t)$] denote the number of packets that have departed from $\mathcal{Q}$ (resp. $\tilde{\mathcal{Q}}$) during the interval $[0, t]$; there holds

$$B(t) \geq \tilde{B}(t), \qquad \forall t \geq 0. \qquad \square$$

Outline of the Proof: This proof is done by extending the argument used in proving Lemma 8. In particular, one has to replace the FIFO "servers" by PS ones, on a level-by-level basis, starting from the 1st level and moving one level at a time. At the jth step of this process, all streams stemming from levels $1, \ldots, j-1$ remain the same, while all streams stemming from levels $j, \ldots, d$ are delayed. The only subtle point of this proof lies on the fact that packets may get out of order at certain steps; see also the proof of Lemma 8. Nevertheless, this creates no difficulty, due to the assumed coupling of routing decisions. If one insists on tracing the path followed by a particular packet [say the first to arrive at "server" $(0, e_1)$] it may occur that this *changes* at some step of the process described above; this is of no importance, because the "comparison" of the various streams still applies, even though the streams may consist of different packets at each step. Q.E.D.

Now that we have established Lemma 9, we can easily prove the following result:

Proposition 10: Let $N(t)$ [resp. $\tilde{N}(t)$] denote the (random) total number of packets present in network $\mathcal{Q}$ (resp. $\tilde{\mathcal{Q}}$) at time t. There holds

$$N(t) \leq_{\text{st}} \tilde{N}(t), \qquad \forall t \geq 0. \qquad \square$$

Proof: On a sample path basis, there holds $N(t) = B(t) - A(t)$, where $A(t)$ [resp. $B(t)$] is the number of arrivals at (resp. departures from) network $\mathcal{Q}$ during $[0, t]$; a similar relation holds for network $\tilde{\mathcal{Q}}$. Using Lemma 9, we have $N(t) \leq \tilde{N}(t)$ on a sample path basis. Relaxing the coupling of the arrival processes and the routing decisions in the two networks, we obtain the stochastic inequality in question. *Q.E.D.*

Notice that Proposition 10 (and Lemma 9) applies for *all* layered networks with Markovian routing and deterministic FIFO servers (possibly with different service times); in particular, if the FIFO discipline is changed to PS, then the total number of customers in such a network increases in the stochastic sense.

Next, we present the main result of this subsection.

Proposition 11: The delay T of the greedy routing scheme under analysis satisfies

$$T \leq \frac{dp}{1-\rho}, \qquad \forall \rho < 1. \qquad \square$$

Proof: As established in [21, pp. 93–94], network $\tilde{\mathcal{Q}}$ is of the *product form*, provided that it is stable. Since the total arrival rate for each "server" equals ρ (as was the case under the FIFO discipline), the steady-state probability that a particular "server" of $\tilde{\mathcal{Q}}$ hosts n packets equals $(1-\rho)\rho^n$. Therefore, the steady-state average total number $\tilde{N}$ of packets present in $\tilde{\mathcal{Q}}$ equals $\tilde{N} = d2^d\frac{\rho}{1-\rho}$. This together with Proposition 10 implies that the average total number N of packets present in network $\mathcal{Q}$ (in steady-state) satisfies

$$N \leq d2^d \frac{\rho}{1-\rho}. \tag{11}$$

Recall now the equivalence of network $\mathcal{Q}$ with the d-cube under the greedy routing scheme analyzed. By Little's law, the average delay T induced by this scheme satisfies

$$T = \frac{N}{\lambda 2^d} = \frac{Np}{\rho 2^d}. \tag{12}$$

This together with (11) proves the result. Q.E.D.

It should be noted that the steady-state average number of packets N is guaranteed to exist. This is because, for $\rho < 1$, network $\tilde{\mathcal{Q}}$ empties infinitely often for each sample path; this also applies to network $\mathcal{Q}$, because of Proposition 10 (which holds on a sample-path basis as well). When $\mathcal{Q}$ empties it *regenerates* and this property suffices for our purposes; the technical details are omitted.

Next, we comment on the number of packets stored per hypercube *node*. The steady-state average number of packets per node equals $\frac{N}{2^d}$; this satisfies $\frac{N}{2^d} \leq d\frac{\rho}{1-\rho}$. Thus, it is seen that, for any fixed ρ, the average size of the queue built at each node is $O(d)$. In fact, one can show that the total number of packets within the d-cube is $O(d2^d)$ with high probability. Indeed, by Proposition 10 and the product-form property of $\tilde{\mathcal{Q}}$, the random variable $\lim_{t\to\infty} N(t)$ is stochastically dominated (in steady-state) by the sum of $d2^d$ independent geometrically distributed random variables with expected value $\frac{\rho}{1-\rho}$. Using the Chernoff bound, it follows that, for $t \to \infty$, $N(t) \leq d2^d\frac{\rho}{1-\rho}(1+\epsilon)$ with high probability, for any $\epsilon > 0$.

As a final result, we present a lower bound on the delay T which is a little sharper than the lower bound of Proposition 2 (at most by a factor of 2).

Proposition 12: The delay T of the greedy routing scheme under analysis satisfies

$$T \geq dp + p\frac{\rho}{2(1-\rho)}, \qquad \forall \rho < 1. \qquad \square$$

Proof: Let N_j denote the stationary average number of packets in the queue for an arc of the jth dimension. Since each dimension comprises 2^d arcs, there holds

$$N = \sum_{j=1}^{d} 2^d N_j. \tag{13}$$

Each arc of the 1st dimension is only fed by a Poisson stream with rate $\rho < 1$; using the expression for the average size of an $M/D/1$ queue (see [12]), it follows that

$$N_1 = \rho + \frac{\rho^2}{2(1-\rho)}. \tag{14}$$

Recall now that, for $j \geq 2$, arc $(x, x \oplus e_j)$ has a total arrival rate of ρ; since each packet stays at an arc for at least one time unit, we have $N_j \geq \rho$ for $j = 2, \ldots, d$. Combining this with (13) and (14), we obtain

$$N \geq d2^d\rho + 2^d\frac{\rho^2}{2(1-\rho)};$$

this together with (12) proves the result. Q.E.D.

Next, notice that, by Propositions 11 and 12, we have (for fixed p)

$$\frac{p}{2} \leq \lim_{\rho\to 1}[(1-\rho)T] \leq dp.$$

It is an interesting open problem to close the gap in the above inequality. It is conjectured that for all $p \in (0,1)$, the upper bound is tight (within a factor independent of d). This conjecture is based on the fact that, for $p \in (0,1)$, each packet $\mathcal{P}$ faces *additional* contention for each dimension it crosses; that is, $\mathcal{P}$ contends with packets that had not entered the path of $\mathcal{P}$ up to this point. On the other hand, it is easily seen that the lower bound is tight for $p = 1$. Indeed, in this case, each packet generated at node x is destined for node $\bar{x}$, where each entry of the binary identity of $\bar{x}$ is the complement of the corresponding entry of x; thus, by crossing the hypercube dimensions in increasing index-order, packets generated at different nodes follow *disjoint* paths; this easily gives that $T = d + \frac{\rho}{2(1-\rho)}$.

IV. Greedy Routing on the Butterfly Network

In this section, we extend the results derived for the hypercube to the butterfly network. First, we briefly describe the basic properties of this network.

A. The Butterfly Network

The d-dimensional butterfly is an "unfolded" version of the d-cube. It consists of $(d+1)2^d$ nodes, organized in $d+1$ levels, with each level having 2^d nodes. In particular, for $j \in \{1, \ldots, d+1\}$, the nodes of the jth level are denoted by $[x; j]$ where $x \in \{0, \ldots, 2^d - 1\}$. For $j \neq d+1$, each node $[x; j]$ is connected to two nodes, namely $[x; j+1]$ and $[x \oplus e_j; j+1]$; see Fig. 3(a), where the 2-butterfly is depicted. Therefore, there exist two types of arcs:

a) Arcs of the form $[x; j] \to [x; j+1]$, which are referred to as *straight* arcs; for notational convenience, arc $[x; j] \to [x; j+1]$ will be denoted by $(x; j; \mathrm{s})$.
b) Arcs of the form $[x; j] \to [x \oplus e_j; j+1]$, which are referred to as *vertical* arcs; for notational convenience, arc $[x; j] \to [x \oplus e_j; j+1]$ will be denoted by $(x; j; \mathrm{v})$.

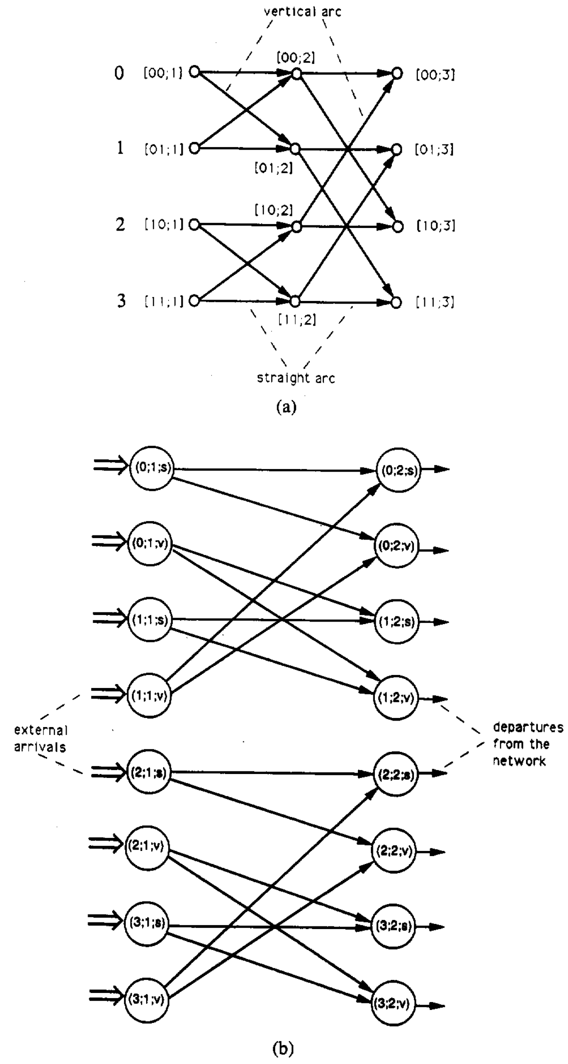


Fig. 3. (a) The two-dimensional butterfly. (b) The equivalent network $\mathcal{R}$ for the three-dimensional butterfly.

We view the butterfly as a switching network: packets are generated at the 1st level and destined for the $(d+1)$st level. It is easily seen that for each origin-destination pair $[x;1]$ and $[z;d+1]$ there corresponds a *unique* path, which consists of d arcs. In particular, let $i_1,\ldots,i_k$ be the entries in which the binary identities of x and z differ, with $i_1 < i_2 < \cdots < i_k$. Then, the path from $[x;1]$ to $[z;d+1]$ contains exactly k vertical arcs, namely

$$(x;i_1;\mathrm{v}),(x\oplus e_1;i_2;\mathrm{v}),\ldots,(x\oplus e_1\cdots\oplus e_{i_{k-1}};i_k;\mathrm{v});$$

the remaining $d-k$ arcs of the path are straight arcs. Notice that these k vertical arcs correspond to the arcs traversed by a packet travelling from x to z in the d-cube, when dimensions are crossed in increasing index-order.

B. Preliminary Results

The dynamic routing problem to be analyzed is essentially the same as that in the context of the d-cube. That is, each node of the 1st level independently generates packets according to a Poisson process with rate λ; all packets have unit transmission time. Each packet has a single destination in the $(d+1)$st level; this destination is selected randomly, according to the following rule:

$$\Pr\big[\text{a packet generated by node } [x;1] \text{ is destined for node } [z;d+1]\big] = p^{H(x,z)}(1-p)^{d-H(x,z)},$$

where $p \in [0,1]$; recall that $H(x,z)$ denotes the Hamming distance of the binary representations of x and z. Again, different packets make their selections independently of each other. Notice that, for $p=\frac{1}{2}$, the destination distribution is uniform over the nodes of the $(d+1)$st level; that is, each such node is equally likely to be chosen as a packet's destination.

First, we note that a result analogous to Lemma 1 applies; however, in the present context, $\mathcal{B}_j$ corresponds to the event that a packet has to traverse a vertical arc stemming from the jth level. Furthermore, notice that arcs $(x;1;\mathrm{s})$ and $(x;1;\mathrm{v})$ may only be traversed by packets generated by node x. Therefore, packets to traverse arc $(x;1;\mathrm{v})$ form a Poisson stream with rate λp; similarly, packets to traverse arc $(x;1;\mathrm{s})$ form a Poisson stream with rate $\lambda(1-p)$. Recalling that all packets have unit transmission time, it follows that the inequalities $\lambda p < 1$ and $\lambda(1-p) < 1$ are both *necessary* conditions for stability of any routing scheme. Combining these conditions, we obtain the following result: Stability may prevail only if

$$\rho \overset{\mathrm{def}}{=} \lambda \max\{p, 1-p\} < 1. \tag{15}$$

Notice that, for given λ, the maximum value of ρ occurs for $p=\frac{1}{2}$. For $p > \frac{1}{2}$, the vertical arcs become the bottleneck of the system; for $p < \frac{1}{2}$, the straight arcs become the bottleneck of the system (cf. Lemma 14 below).

Next, we present a *universal* lower bound on the average delay T per packet.

Proposition 13: Under *any* routing scheme, there holds

$$T \geq d + p\frac{\lambda p}{2(1-\lambda p)} + (1-p)\frac{\lambda(1-p)}{2[1-\lambda(1-p)]}. \qquad \square$$

Proof: When no idling occurs, the value W_{v} (resp. W_{s}) of the average delay induced by arc $(x;1;\mathrm{v})$ [resp. $(x;1;\mathrm{s})$] equals that of an $M/D/1$ queue with arrival rate λp [resp. $\lambda(1-p)$] and unit service duration; when idling occurs, these delay values are larger. Thus, we have [12]

$$W_{\mathrm{v}} \geq 1 + \frac{\lambda p}{2(1-\lambda p)} \quad \text{and}$$

$$W_{\mathrm{s}} \geq 1 + \frac{\lambda(1-p)}{2[1-\lambda(1-p)]}. \tag{16}$$

Note that after a packet arrives at the second level, it requires at least $d-1$ more time units until it reaches its destination; thus, it is seen that, under any routing scheme, the average delay T per packet satisfies

$$T \geq d - 1 + pW_{\mathrm{v}} + (1-p)W_{\mathrm{s}}.$$

This together with (16) proves the result. Q.E.D.

Equation (15) as well as Proposition 13 demonstrate the limitations applying to the performance of any routing scheme. The scheme to be analyzed below is the simplest possible:

Packets are routed in a *greedy* fashion; that is, each packet advances at its respective path as fast as possible. When several packets contend for the same arc, then priority is allotted on a FIFO basis.

In fact, given that there is only one path per origin-destination pair, greedy routing is the most natural scheme arising in the context of the butterfly. It will be shown in Section IV–C that this simple scheme is very efficient.

C. Performance Analysis of Greedy Routing

Similarly with the hypercube (see § 3.1), under greedy routing, the butterfly may be viewed as a queueing network $\mathcal{R}$ with $d2^{d+1}$ deterministic FIFO "servers"; each of them has unit service duration and corresponds to an *arc*. Furthermore, $\mathcal{R}$ is acyclic and routing is Markovian. In Fig. 3(b), we present the network $\mathcal{R}$ corresponding to the two-dimensional butterfly.

Next, we investigate the stability properties of our greedy routing scheme; for this purpose, we need the following simple result, whose proof is omitted.

Lemma 14: The total arrival rate at each arc $(x; j; \mathrm{s})$ equals $\theta_{\mathrm{s}} = \lambda(1-p)$. Also, the total arrival rate at each arc $(x; j; \mathrm{v})$ equals $\theta_{\mathrm{v}} = \lambda p$. □

Similarly with Section III-B, the sufficient condition for stability of the equivalent network $\mathcal{R}$ (and of the greedy routing scheme) is obtained by applying the result of [21, p. 246]; this condition is as follows:

Proposition 15: Greedy routing on the butterfly is stable if

$$\lambda p < 1 \qquad \text{and} \qquad \lambda(1-p) < 1,$$

or equivalently $\rho \overset{def}{=} \lambda \max\{p, 1-p\} < 1$. □

In light of the necessary condition for stability in (15), it is seen that greedy routing in the butterfly has *optimal* stability properties. We now establish the upper bound for the average delay T per packet induced by greedy routing.

Proposition 16: There holds

$$T \leq \frac{dp}{1-\lambda p} + \frac{d(1-p)}{1-\lambda(1-p)}, \qquad \forall \rho < 1. \quad \square$$

Proof: By Little's law, we have

$$T = \frac{N}{\lambda 2^d}, \tag{17}$$

where N is the average total number of packets present in the equivalent network $\mathcal{R}$ in steady-state. We now consider the network $\tilde{\mathcal{R}}$, which is identical to $\mathcal{R}$ except for the fact that all of its "servers" operate under a PS discipline; let $\tilde{N}$ be the corresponding average total number of packets. Since $\mathcal{R}$ is a layered network with Markovian routing, we can apply Proposition 10; see also the comment on the generality of that result, following its proof. Therefore, we have

$$N \leq \tilde{N}. \tag{18}$$

In the stable case (i.e., for $\rho < 1$), network $\tilde{\mathcal{R}}$ is of the product form [21, pp. 93–94]. Recalling also Lemma 14, it follows that the stationary probability that a particular "server" $(x; j; \mathrm{v})$ [resp. $(x; j; \mathrm{s})$] of $\tilde{\mathcal{R}}$ hosts n packets equals $(1-\lambda p)(\lambda p)^n$ (resp. $[1-\lambda(1-p)][\lambda(1-p)]^n$). Since there exist $d2^d$ "servers" of each of the two types, it follows that

$$\tilde{N} = d2^d \frac{\lambda p}{1-\lambda p} + d2^d \frac{\lambda(1-p)}{1-\lambda(1-p)}.$$

This together with (17) and (18) proves the result. Q.E.D.

Next, we comment on the number of packets stored per node of the butterfly; first, notice that only the nodes of levels $1, \ldots, d$ have to store packets. An overall estimate of the expected number of packets per node is provided by the quantity $\frac{N}{d2^d}$, which satisfies

$$\frac{N}{d2^d} \leq \frac{\lambda p}{1-\lambda p} + \frac{\lambda(1-p)}{1-\lambda(1-p)} \overset{\text{def}}{=} q_\rho.$$

This estimate is quite favorable because it suggests that the "overall" average queue-size per node is $O(1)$ for any fixed ρ. However, it is not guaranteed that this bound holds for the average number of packets stored by the nodes of each individual level. It is conjectured that this is actually the case; the following result provides strong evidence for this claim: for any $j \in \{1, \ldots, d\}$, the total number of packets stored by the nodes of levels $1, \ldots, j$ does not exceed $j2^d q_\rho(1+\epsilon)$ with high probability, for any $\epsilon > 0$. This result may be proved by applying stochastic domination between the first j levels of networks $\mathcal{R}$ and $\tilde{\mathcal{R}}$, and using the product-form property of $\tilde{\mathcal{R}}$.

Using Propositions 13 and 16 and the definition of ρ it follows that

$$\frac{1}{2}\max\{p, 1-p\} \leq \lim_{\rho \to 1}[(1-\rho)T] \leq d \max\{p, 1-p\}.$$

It is an interesting open problem to close the gap in the above inequality. As in the case of hypercubes (see the end of § 3.3), it is conjectured that the upper bound is tight for all $p \in (0,1)$; for $p = 0$ and for $p = 1$, the lower bound is tight, because packets originating at different nodes follow disjoint paths.

V. Concluding Remarks

In this paper, we analyzed a problem where the nodes of the hypercube network generate packets at random time instants, according to independent Poisson processes. Each packet has unit transmission time and is destined for a randomly selected node; in a special case, the destination distribution is uniform. We considered a simple greedy routing scheme, where each packet crosses the hypercube dimensions required in increasing index-order. We proved that this scheme has optimal stability properties and, when stable, it induces an average delay $T = \Theta(d)$ per packet; the bounds on the average delay were given in simple closed-form expressions. Our analysis was based on a new approach, which relates the behavior of the hypercube (under the routing scheme considered) to that of a queueing network with Markovian routing. Using the same idea, we extended the results to the butterfly network, thus proving the efficiency of greedy routing in this context.

It would be of interest to analyze the problem under an arbitrary destination distribution. For this case, it may be profitable to "mix" the packets by first sending each of them to a random intermediate node, as is done for the permutation task in [18] and [19]. Such a "mixing" may result in improved delay properties under medium traffic, at the expense of reducing the maximum traffic that may be sustained by the system.

In an even more general version of the problem analyzed, it may be assumed that each packet is destined for a different subset of nodes; it may also be assumed that the packets received by a node influence the packet-generating process of this node as well as the lengths and destinations of the new packets. This situation arises in the distributed execution of iterative algorithms. Analyzing this general problem seems to be a rather challenging and interesting direction for further research.

Acknowledgment

The authors are grateful to Dr. T. Ott for mentioning the possibility of comparing the FIFO and PS disciplines.

References

[1] S. Abraham and K. Padmanabhan, "Performance of the direct binary n-cube network for multiprocessors," in *Proc. 1986 Int. Conf. Parallel Processing*, 1986.

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

[3] C. Bouras, J. Garofalkis, P. Spirakis, and V. Triantafillou, "Queueing delays in buffered multistage interconnection networks," Dep. Comput. Sci., New York University, Tech. Rep. 289, 1987.

[4] D. P. Bertsekas, C. Ozveren, G. D. Stamoulis, P. Tseng, and J. N. Tsitsiklis, "Optimal communication algorithms for hypercubes," *J. Parallel Distrib. Comput.*, vol. 11, pp. 263–275, 1991.

[5] A. Borodin and J. E. Hopcroft, "Routing, merging and sorting on parallel models of computation," in *Proc. 14th Annu. ACM Symp. Theory of Comput.*, pp. 338–344, 1982.

[6] S. L. Brumelle, "Some inequalities for parallel-server queues," *Oper. Res.*, vol. 19, pp. 402–413, 1971.

[7] Y. Chang and J. Simon, "Continuous routing and batch routing on the hypercube," in *Proc. 5th ACM Symp. Principles of Distrib. Comput.*, pp. 272–281, 1986.

[8] A. G. Greenberg and J. Goodman, "Sharp approximate models of adaptive routing in mesh networks," preprint, 1986.

[9] A. G. Greenberg and B. Hajek, "Deflection routing in hypercube networks," preprint, 1989.

[10] B. Hajek and R. L. Cruz, "Delay and routing in interconnection networks," in *Flow Control of Congested Networks*, A. R. Odoni, L. Bianco, and G. Szago, Eds. New York: Springer-Verlag, 1987.

[11] S. L. Johnsson and C. -T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Comput.*, vol. 38, pp. 1249–1267, 1989.

[12] L. Kleinrock, *Queueing Systems, Vol. I: Theory.* New York: Wiley, 1975.

[13] F. T. Leighton, "Average case of greedy routing algorithms on arrays," preprint, 1990.

[14] D. Mitra and R. A. Cieslak, "Randomized parallel communications on an extension of the Omega network," *J. ACM*, vol. 34, pp. 802–824, 1987.

[15] Y. Saad and M. H. Schultz, "Data communication in hypercubes," Dep. Comput. Sci., Yale University, Res. Rep. YALEU/DCS/RR-428, 1985.

[16] G. D. Stamoulis and J. N. Tsitsiklis, "Efficient routing schemes for multiple broadcasts in hypercubes," in *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, pp. 725–739, 1993.

[17] G. D. Stamoulis, "Routing and performance evaluation in interconnection networks," Ph.D. dissertation, Dep. of EECS, M.I.T, June 1991.

[18] L. G. Valiant and G. J. Brebner, "Universal schemes for parallel communication," in *Proc. 13th Annu. ACM Symp. Theory of Comput.*, pp. 263–277, 1981.

[19] L. G. Valiant, "A scheme for fast parallel communication," *SIAM J. Comput.*, vol. 11, pp. 350–361, 1982.

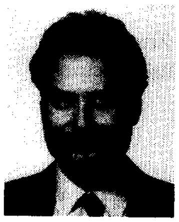
[20] L. G. Valiant, "General purpose parallel architectures," Aiken Computation Lab., Harvard Univ., Rep. TR-07-1988, 1988.

[21] J. Walrand, *An Introduction to Queueing Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

George D. Stamoulis was born in Athens, Greece, in 1964. He received the diploma in electrical engineering in 1987 (with highest honors) from the National Technical University of Athens, Athens, Greece, and the M.S. degree in 1988, and the Ph.D. degree in 1991 in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

He is currently serving in the Hellenic Navy as a lecturer also in the Hellenic Naval Academy. He is also a Research Associate with the communication networks group of the Department of Electrical and Computer Engineering, National Technical University of Athens; he is participating in RACE projects. His research interests are in the areas of routing and performance evaluation of multiprocessing systems and communication networks, and queueing theory.

Dr. Stamoulis was among the winners of the Greek Mathematics Olympiad in both 1981 and 1982. He also participated in the 23rd International Mathematic Olympiad, in Budapest, in July 1982. He is a member of the Technical Chamber of Greece and Sigma Xi.



John N. Tsitsiklis (S'80–M'83) was born in Thessaloniki, Greece, in 1958. He received the B.S., M.S. and Ph.D. degrees in electrical engineering, all from the Massachusetts Institute of Technology, Cambridge, in 1980, 1981, and 1984, respectively.

During the academic year 1983–1984, he was an acting professor of electrical engineering at Stanford University, Stanford, CA, USA. Since 1984, he has been with the Massachusetts Institute of Technology, where he is currently a Professor of Electrical Engineering. His research interests are in the areas of systems and control theory, parallel and distributed computation, and operations research in which he has co-authored more than 50 papers. He is co-author (with D. Bertsekas) of *Parallel and Distributed Computation: Numerical Methods* (1989). He has been a recipient of an IBM Faculty Development Award (1983), an NSF Presidential Young Investigator Award (1986), Outstanding Paper Award by the IEEE Control Systems Society (for a paper co-authored with M. Athans, 1986), and of the Edgerton Faculty Achievement Award by M.I.T. (1989). He was a plenary speaker at the 1992 IEEE Conference on Decision and Control. Finally, he is an Associate Editor of *Automatica* and *Applied Mathematics Letters*, and has been an Associate Editor of the IEEE Transactions on Automatic Control.