

Enhanced Hypercubes

Nian-Feng Tzeng, *Member, IEEE*, and Sizheng Wei, *Student Member, IEEE*

Abstract—A node in real hypercube systems often has links left unused, because each node is manufactured with a fixed number of links designed for a maximum sized configuration. In this paper, we investigate the hypercube with extra connections added between pairs of nodes through otherwise unused links. Those extra connections are made in a way to maximize the improvement of the performance measure of interest under various traffic distributions. The resulting hypercube, called the enhanced hypercube, requires a simple routing algorithm and is guaranteed not to create any traffic-congested point or link. The enhanced hypercube achieves noticeable improvement in diameter, mean internode distance, and traffic density, and also is more cost-effective when compared to a regular hypercube. An efficient broadcast algorithm which can considerably speed up the broadcast process in enhanced hypercubes is provided as well.

Index Terms—Hypercubes, multicomputer networks, performance analysis, routing algorithms, traffic distributions.

I. INTRODUCTION

HYPERCUBE structures have become the most popular message-passing architecture, and several multicomputer configurations based on the hypercube topology have been designed or even marketed [1]–[4]. There are quite a few hypercube systems currently in use, and with a few scores of nodes, they can deliver performance as high as typical supercomputers on a wide variety of computing problems [5], [6].

Messages in a hypercube may pass through a number of intermediate nodes before reaching their destinations. At each intermediate node, message-processing overhead and queuing delay contribute to the actual communication latency experienced by a message. When executing a job, poor performance may result from an excessive communication time, which could dominate the total run time. In order to maintain high performance for various computing problems, it is desirable to reduce the number of intermediate nodes visited by every message so that the total communication time is kept relatively less than the total computation time. This may be attained by reducing diameter and/or mean internode distance.

In this paper, we study the hypercube where extra connections are established between pairs of cube nodes through otherwise unused links left at each node, which when fabricated, is equipped with a fixed number of links permitting the construction of the maximum allowable size. A hypercube with extra connections, called *skips*, is referred to as an *enhanced hypercube*. The enhanced hypercube achieves a considerable reduction in diameter, and a noticeable improvement in mean internode distance and traffic density. The improvement may

Manuscript received November 7, 1988; revised September 13, 1990. This work was supported in part by the NSF under Grant MIP-8807761 and by the Louisiana Board of Regents' R&D Program Component under Grant 86-USL(2)-127-03.

N.-F. Tzeng is with the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, LA 70504.

S. Wei is with the Department of Computer Science, Rutgers University, New Brunswick, NJ 08903.

IEEE Log Number 9042292.

be translated into a respectable communication time reduction, yielding better performance. As an enhanced hypercube is found to be more cost-effective than a regular hypercube and to suffer from no practical implementation difficulty, it is also beneficial for any hypercube system with no available unused links to augment each node so as to accommodate an additional link, provided that the building block is not pin limited and is allowed to do so.

Unlike the alternative hypercube structure introduced by Hsu *et al.* [7] and the modified hypercube structure addressed by Esfahanian *et al.* [16], the proposed hypercube requires a routing algorithm essentially as simple as that for a regular hypercube. This allows us to modify not only hypercube systems to be manufactured, but also hypercube systems currently in use at a low cost, as will be seen in the subsection concerning implementation issues. Furthermore, the routing algorithm assures that a message would traverse *at most* one skip connection, avoiding any communication bottleneck.

The main objective of this work is to identify the best way of making skips under various traffic distributions and to evaluate the performance improvement of the resulting hypercube. After thorough analysis, it is found that making a skip from a node to its farthest (or second farthest) node for an even (or odd) dimensional hypercube gives rise to optimal or suboptimal performance improvement, provided that the locality of reference is not extremely heavy. This paper is organized as follows: Section II states necessary notation and background. In Section III, a routing procedure for the enhanced hypercube is presented first, followed by the expressions for three fundamental performance measures. Section IV illustrates such resulting hypercubes that attain the maximum improvement in the performance measure of interest under various traffic distributions. An effective broadcast algorithm which can speed up the broadcast process considerably in the proposed hypercubes is introduced in Section V. Section VI discusses the cost-effectiveness measure and implementation issues of the enhanced hypercubes. Finally, concluding remarks appear in Section VII.

II. PRELIMINARIES

An n -dimensional binary cube consists of 2^n nodes, and each node has n links connected directly to n nearest neighbors. A node in the cube can be addressed by a binary form $(x_{n-1} \cdots x_i \cdots x_0)$. The addresses of a node and any one of its nearest neighbors differ in exactly one bit. The maximum number of link traversals a message may make between any two nodes is termed the *diameter*, and the mean number of traversals of a message is referred to as *mean internode distance*.

A. Node-to-Node Communication

The algorithm to route messages from one node to another in hypercubes is simple. Each issued message carries the *relative address* of its source node and its destination node as the routing tag (the relative address of two nodes, say x and y , is the bitwise

Exclusive-OR of their addresses, $x \oplus y$). To facilitate subsequent explanation, a link is assumed to have *link number* i if it connects two nodes whose addresses differ only in the i th bit position (starting with the least significant bit as bit 0). For example, the link between nodes (1101) and (0101) is referred to as link 3. The routing algorithm is formally described below as a procedure, which is executed by every node [9].

Algorithm A: (send or forward a message from node src to node $dest$ with $tag \leftarrow src \oplus dest$ in a regular hypercube):

```

if ( $tag = 0$ )
  { send the message to local processor. }
else
  { starting with the most significant bit of  $tag$ :
    let  $i$  be the bit number of the first 1 in  $tag$ .
    send the message on link  $i$  and set bit  $i$  in  $tag$  to zero. }

```

A message issued at node (0000) with tag 13, for example, is routed sequentially through (1000) and (1100) before reaching its destination (1101). This routing algorithm always finds the shortest paths for messages. In addition, the algorithm consistently scans tags from left to right, avoiding deadlock because higher dimension links are always demanded before lower dimension ones.

Suppose mean internode distance is denoted as D_{mean} . An equation for D_{mean} in general interconnection structures is given [8] by

$$D_{mean} = \sum_{l=1}^n l \times \phi(l) \quad (1)$$

where $\phi(l)$ is the probability of an arbitrary message traversing l links. It is obvious that different $\phi(l)$'s lead to different traffic distributions. Generally, traffic distributions can be uniform or nonuniform. There are many types of nonuniform traffic distributions [8], [10], reflecting distinct types of reference locality. Here, we consider only an interesting nonuniform traffic distribution which characterizes locality as follows: the probability of sending a message from a node to another node decreases as their distance increases. This traffic pattern seems very natural [8] because it reflects (to some extent) the reference locality of many parallel programs, like image processing applications. Other nonuniform traffic distributions can be studied along the same line.

A node is said to be l hops away from another node if the distance between these two nodes is l . Let p_l be the probability of a message (issued at a given node) destined for a node l (> 0) hops away; and p_l equal gp_{l+1} , where $g > 1$ is the decreasing rate of a traffic distribution. The larger the value g is, the higher degree of locality the traffic pattern has. In fact, the case of $g = 1$ models the situation where there is no reference locality and represents a uniform traffic distribution.

Consider a given node in a hypercube. Let all nodes which are l hops away from the given node constitute a *class*, H_l , $1 \leq l \leq n$. Class H_l contains $\binom{n}{l} = \frac{n!}{l!(n-l)!}$ nodes. A message issued at a given node has to traverse l links before terminating at a node in H_l . The probability of an issued message traversing l links is thus expressed by

$$\phi(l) = \binom{n}{l} \times p_l$$

where p_l is $\alpha_0 g^{-l}$ (derived from the relationship of $p_l = gp_{l+1}$), with α_0 a constant chosen to normalize the probabilities ϕ such that their summation is equal to one. Consequently, $\phi(l)$ becomes

$$\phi(l) = \binom{n}{l} \times \alpha_0 g^{-l} \quad (2)$$

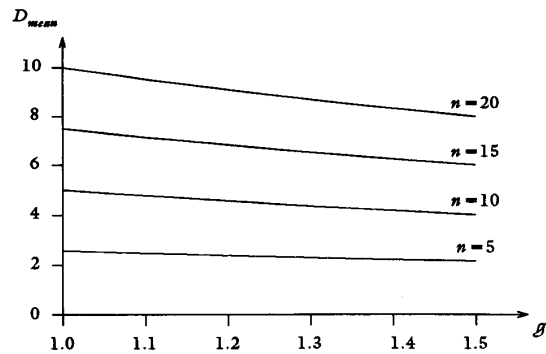


Fig. 1. Mean internode distance (D_{mean}) versus the degree of reference locality (g) for various regular hypercubes.

This equation is applicable to uniform and nonuniform traffic distributions alike. By making use of this equation, we can calculate the probability of an issued message terminating at a particular node, since the summation of all $\phi(l)$'s is 1. It should be noted that α_0 is a function of only n and g ; so adding extra connections to a hypercube does not change α_0 . (Here, we assume that the way of mapping an algorithm to nodes in the enhanced hypercube is identical to that in the regular hypercube.) As long as n and g remain unchanged, an α_0 computed from (2) is also applicable to the hypercube with extra connections and, thus, will be used later. For the case of $g = 1$, the probability for a given message to visit an arbitrary node is found from (2) to be $\frac{1}{2^n - 1}$, which indeed reflects a uniform traffic distribution. By (1) and (2), we are able to obtain D_{mean} of a hypercube. D_{mean} of several hypercubes with various g 's is shown in Fig. 1. It approaches $(n/2)$ as n increases, in the case of uniform traffic distributions. When g increases, as expected, D_{mean} decreases in the same hypercube. It should be noticed that $g = 1.5$ is a case with severe reference locality, as the probability of a message destined for one hop away is then about 38 times as high as for ten hops away.

B. Broadcast Communication

Each node may be able to send a message through at most one outgoing link at a time, or to send/receive messages through all its outgoing/incoming links simultaneously, depending upon the node communication capability. We assume here that every link of nodes has a dedicated pair of transmitters and receivers so that messages can be sent/received through *all* links of a node at the same time. To broadcast a message from one node to all the other nodes in a hypercube of size N can then be achieved by sending the message in $\log_2 N$ steps and sending the message nonredundantly, i.e., each broadcast message is sent to every node exactly once. The following broadcast algorithm is by Sullivan and Bashkow [9]. It is presented in a form of procedures and is executed by every node in a hypercube.

Algorithm B: (originate or forward a broadcast message in a regular hypercube):

```

if (this node is the originating node)
  { set  $weight$  to  $\log_2 N$ . }
{ for each link  $l$  from this node with  $l < weight$ :
  send the message on link  $l$  with  $weight = l$ . }

```

This algorithm is best illustrated by an example. A message broadcast from node (001) in a hypercube with eight nodes is

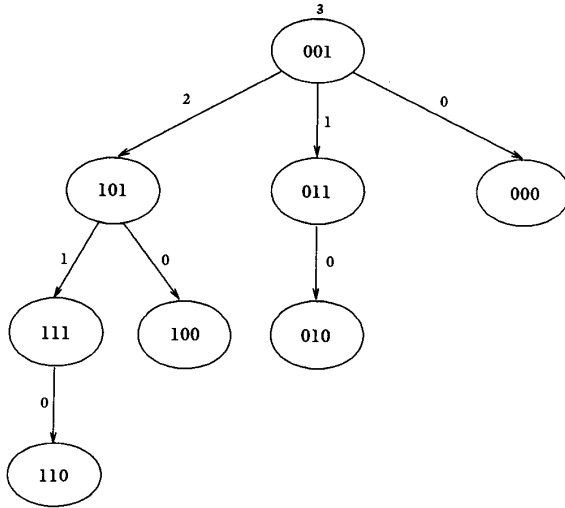


Fig. 2. Broadcast from node (001) in a hypercube with eight nodes.

depicted in Fig. 2, where each arc is associated with a number reflecting its weight. When the message is generated at node (001), the weight is set to 3 and the message is sent through every arc to nodes (000), (011), and (101) with weights being 0, 1, and 2, respectively. After receiving the message and a weight, a node would forward the message to its immediate neighbors through every link l which is less than the weight. For instance, node (101) would forward the message through link 0 and link 1 to node (100) and node (111), respectively. The process repeats and every node would receive exactly one copy of the message after 3 steps.

III. ENHANCED HYPERCUBES

The general measures for an interconnection structure include diameter, mean internode distance, traffic density over a link, the number of connection links per node, routing control complexity, and visit ratios, among others [8], [10], [11]. Several of them are interrelated, e.g., a smaller diameter tends to give a shorter mean internode distance under the same traffic distribution; a shorter mean internode distance leads to lighter traffic density and smaller visit ratios, given the same amount of links and nodes; etc.

Suppose a skip is established between every pair of nodes, $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_i \cdots x_0)$ and $(x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_i \cdots \bar{x}_0)$, with $0 \leq k \leq n-2$. The three-dimensional enhanced hypercubes with $k=1$ and 0 are sketched in Fig. 3. Different k 's lead to different enhanced hypercubes and would yield different degrees of performance improvement. Now, two issues associated with enhanced hypercubes are yet to be solved:

- 1) What is a proper routing algorithm?
- 2) What are the results of performance measures of enhanced hypercubes?

The second issue is important because the results make it possible for us to find out the best way of establishing skips such that the performance improvement of the resulting hypercubes (with respect to a certain measure) is maximized. They also enable us to compare enhanced hypercubes to their regular counterparts. In the following, we address these two issues separately.

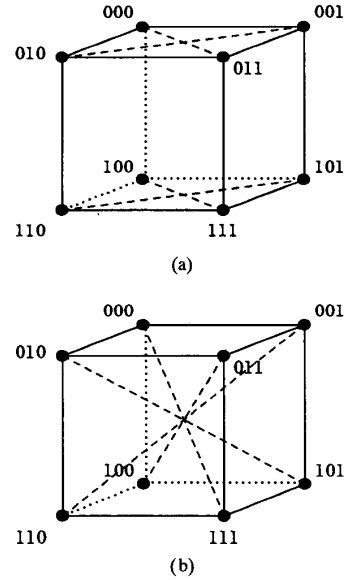


Fig. 3. The three-dimensional enhanced hypercubes with (a) $k=1$ and (b) $k=0$.

A. Node-to-Node Routing Algorithm

To route messages in an enhanced hypercube is essentially the same as in a regular hypercube described previously. However, the routing algorithm has to take advantage of skips as much as possible and avoid creating any potential traffic bottleneck. Whenever a message is initially generated at a node, its tag is checked to decide if the message should traverse the additional link to reduce traversals (the tag of a message is again the bitwise Exclusive-OR of its source and destination addresses). The decision is made according to the total number of nonzero bits in the least significant $(n-k)$ bits of the tag: if there are more than $\lceil (n-k)/2 \rceil$ nonzero bits, the message is delivered through the skip; otherwise, the message takes one of the regular links. When a message is sent over a skip, the lower $(n-k)$ bits of the tag are complemented. From then on, the routing process is exactly identical to Algorithm A given earlier. A node thus treats pass-through messages in the same way as a node in a regular hypercube. It is this property that enables us to modify hypercubes easily, as will be discussed in Section VI (regarding implementation issues). The routing algorithm is formally described as follows.

Algorithm C: (send or forward a message from node src to node $dest$ with $tag \leftarrow src \oplus dest$ in an enhanced hypercube):

```

if ( $src$  is not the originating node)
  { the same as Algorithm A. }
else
  { count the number of nonzero bits in the least significant
    ( $n-k$ ) bits of  $tag$ .
    if (the number of nonzero bits  $> \lceil (n-k)/2 \rceil$ )
      { send the message through the skip and
        complement the lower  $(n-k)$  bits of  $tag$ . }
    else
      { the same as Algorithm A. }
  }

```

Like Algorithm A, this algorithm is executed by each node and can readily be shown deadlock-free because it selects skips

first whenever necessary, and then scans tags along a consistent order; the ordering ensures that no deadlocks can occur. A message routed by Algorithm C takes *at most one skip* in an enhanced hypercube. However, messages can be guaranteed to follow shortest paths in any case, as described below (a proof of this theorem is given in the Appendix).

Theorem 1: A message routed by Algorithm C always follows a shortest path in any enhanced hypercube.

Since a newly generated message in an enhanced hypercube may or may not traverse a skip, and the total number of skips is smaller than that of total regular links, it can easily be verified from Algorithm C that the traffic amount over regular links and the amount over skips are not equivalent. As a consequence, an enhanced hypercube no longer possesses link symmetry and its skips should be utilized efficiently. From the above theorem, it is sure that there is no need to route a message along skips repeatedly, avoiding heavy traffic over skips. In fact, traffic density is shown to be no higher on skips than on regular links in the next section. No skip would possibly become a system bottleneck. A simple yet effective broadcast algorithm that can speed up the broadcasting process in enhanced hypercubes will be presented in Section V.

B. Results of Performance Measures

Among general measures of an interconnection structure, diameter, mean internode distance, and traffic density are of particular interest. The results of the three measures of an enhanced hypercube are derived below.

Diameter: For an enhanced hypercube of size 2^n in which there is a skip between every pair of nodes, $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ and $(x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_0)$, we have the following theorem (a proof is provided in the Appendix). It can be seen that the reduction in diameters depends entirely on k , with a maximum of about 50% reduction (when k equals 0).

Theorem 2: The diameter of an n -dimensional enhanced hypercube with a given k is $k + \lceil (n-k)/2 \rceil$.

Mean Internode Distance: To derive D_{mean} of an enhanced hypercube is involved because added connections may change the distance between a given node and a node in Class H_1 . Consider the distance between node (000) and nodes in $H_2 = \{(011), (101), (110)\}$, as illustrated in Fig. 3(a), where a skip exists between (000) and (011). The added connection makes the distance between (000) and (011) change from 2 to 1, while the distance between (000) and any other node in H_2 remains 2. Although (1) can still be applied to calculate D_{mean} , we are no longer able to derive a simple expression for $\phi(l)$ as given in (2).

Suppose that a skip is established between each pair of nodes, $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ and $(x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_0)$, $0 \leq k \leq n-2$. Without loss of generality, we may consider merely the distance between an arbitrary node and node (00...00) in such an enhanced hypercube. Let us examine the distance between a node $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ in Class H_{n-k+j} , $0 \leq j \leq k$, and the node (00...00) first. Assume i out of the $n-k$ bits of $x_{n-k-1} \cdots x_0$ are "0"; then, $i+j$ of the k bits of $x_{n-1} \cdots x_{n-k}$ must be "1" (because totally $n-k+j$ bits are "1"). For a message being transmitted between the two nodes, the message requires $j+2^*i$ additional hops to reach its destination if it ever traverses the skip. This is because after the traversal, the message is at a node whose address has $j+2^*i$ corresponding bits differing from its tag. Thus, the distance between the two nodes equals $\min(1+j+2^*i, n-k+j)$, where $\min(a, b)$ is

the minimum of a and b . Among the total $\binom{n}{n-k+j}$ nodes in H_{n-k+j} , $\binom{n-k}{i+j} \binom{k}{i}$ of them have this type of address. Since i can be 0, 1, ..., or $\min(k-j, n-k)$ and the probability of a message issued by node (00...00) destined for any node in H_{n-k+j} is $\alpha_0 g^{-(n-k+j)}$ (where α_0 is the same as in a regular hypercube and is obtained as described earlier), we have

$$D_{\text{mean}}^{n-k+j} = \alpha_0 g^{-(n-k+j)} \sum_{i=0}^{\min(k-j, n-k)} \binom{k}{i+j} \binom{n-k}{i} \times \min(1+j+2^*i, n-k+j) \quad (3)$$

where D_{mean}^z denotes the mean distance of a message between node (00...00) and H_z , given that the message is directed to a node in H_z . It is interesting to see that $\sum_{i=0}^{\min(k-j, n-k)} \binom{k}{i+j} \binom{n-k}{i}$ is equal to $\binom{n}{n-k+j}$, and every node in H_{n-k+j} is thus considered in (3).

Likewise, the distance between a node $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ in Class H_{n-k-j} , $1 \leq j < n-k$, and the node (00...00) can be examined. Suppose i out of the k bits of $x_{n-1} \cdots x_{n-k}$ are "1"; then, $i+j$ of the $n-k$ bits of $x_{n-k-1} \cdots x_0$ must be "0". The distance between these two nodes is $\min(1+j+2^*i, n-k-j)$. Therefore, we have

$$D_{\text{mean}}^{n-k-j} = \alpha_0 g^{-(n-k-j)} \sum_{i=0}^{\min(k, n-k-j)} \binom{k}{i} \binom{n-k}{i+j} \times \min(1+j+2^*i, n-k-j). \quad (4)$$

Similarly, we can observe that every node in H_{n-k-j} is included in (4). With (3) and (4), the mean internode distance, D_{mean} , immediately follows

$$D_{\text{mean}} = \sum_{j=0}^k D_{\text{mean}}^{n-k+j} + \sum_{j=1}^{n-k} D_{\text{mean}}^{n-k-j}. \quad (5)$$

Traffic Density: Traffic density is an important structure measure because higher traffic density tends to cause longer delay, potentially yielding poor performance. It is measured in terms of messages per link per unit time, given that each node issues one random message per unit time. In an enhanced hypercube, traffic density on a regular link (denoted by TD_r) is different from that on a skip (denoted by TD_s). TD_r and TD_s are derived below.

According to our routing algorithm, an issued message takes a skip immediately after it is generated only if the least significant $(n-k)$ bits of its source and destination addresses differ in l bits such that $\lceil (n-k)/2 \rceil < l \leq n-k$, whereas the most significant k bits can be arbitrary. Since each skip is shared by two nodes, we have

$$TD_s = 2 \times \sum_{l=\lceil (n-k)/2 \rceil + 1}^{n-k} \binom{n-k}{l} \times \left(\sum_{j=0}^k \binom{k}{j} \times p_{l+j} \right) \quad (6)$$

where $p_i = \alpha_0 g^{-i}$ with α_0 obtained from (2).

Every node generates a random message and every message behaves statistically identically; so we can obtain TD_r by using D_{mean} and TD_s . It is easy to see that the total number of traversals made by all the generated 2^n messages is $2^n \times D_{\text{mean}}$, which is shared by $n2^{n-1}$ regular links and 2^{n-1} skips. The number of traversals over the 2^{n-1} skips is $2^{n-1} \times TD_s$ (because a message takes only one skip traversal), so the amount of traversals over the $n2^{n-1}$ regular links is $2^{n-1}(2 \times D_{\text{mean}} -$

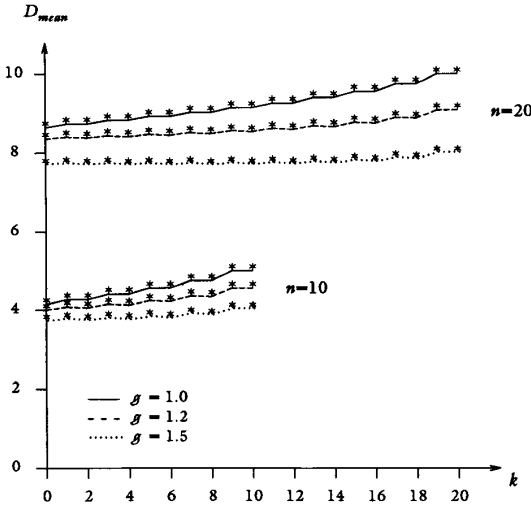


Fig. 4. Mean internode distance (D_{mean}) versus k under various traffic distributions (system size = 2^n).

TD_s). This amount of traversals is shared equally by regular links, yielding

$$TD_r = \frac{2 \times D_{mean} - TD_s}{n} \quad (7)$$

Notice that (6) and (7) are valid for enhanced hypercubes under uniform and nonuniform traffic distributions alike.

Unlike diameter, mean internode distance and traffic density depend not only on k but also on g , the degree of reference locality. Our goal is to determine k so as to minimize one of the performance measures under consideration. With k determined, the other two performance measures can be obtained by utilizing the above expressions.

IV. OPTIMAL ENHANCED HYPERCUBES

An enhanced hypercube is said to be *optimal* with respect to a performance measure if it has the maximum improvement in that performance measure. Optimal enhanced hypercubes with respect to mean internode distance are discussed first.

A. With Respect to Mean Internode Distance

This type of optimal enhanced hypercube can be obtained by determining k such that (5) is minimized. To this end, we consider uniform and nonuniform traffic distributions separately.

Uniform Traffic Distributions: Mean internode distance (D_{mean}) versus k for $n = 10$ and 20 with $g = 1$ is shown in Fig. 4. Note that D_{mean} is identical to the mean internode distance of a regular hypercube given in Fig. 1 when $k = n - 1$ or $k = n$. D_{mean} is found to be minimum as $k = 0$ for all system sizes. This means that a skip should be established between $(x_{n-1} \cdots x_1 \cdots x_0)$ and $(\bar{x}_{n-1} \cdots \bar{x}_1 \cdots \bar{x}_0)$ so as to achieve the maximum improvement in D_{mean} .

The result is somewhat expected because skips are then employed to connect pairs of nodes which lie farthest apart. As a result, D_{mean} of the enhanced hypercube is reduced by 0.85 (with respect to that of the regular hypercube) for $n = 10$, or by 1.35 for $n = 20$ (see the cases of $g = 1.0$ in Fig. 5). The reduced amount grows as n increases and is always noticeable.

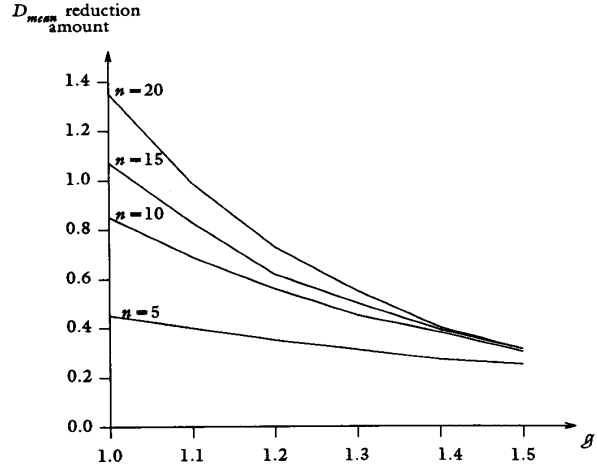


Fig. 5. D_{mean} reduction amount versus g in enhanced hypercubes.

For example, D_{mean} in the enhanced hypercube of $n = 20$ is about equal to D_{mean} in the regular hypercube of $n = 17$ (inferred from Fig. 1), a system whose size is only $1/8$ of 2^{20} .

The diameter of an enhanced hypercube with $k = 0$ is reduced by roughly 50% for all sized systems. Traffic density over a link in enhanced hypercubes can be evaluated by making use of (6) and (7). Table I illustrates TD_r and TD_s for a variety of system sizes. Since TD_s is consistently no larger than TD_r , added skips *never become communication bottlenecks* for all sized systems. Traffic density in a regular hypercube (denoted by TD) can be computed by (7) with $TD_s = 0$. The ratio of TD_r to TD for various system sizes is included in Table I. It can be seen that the improvement rate of traffic density over regular links is noticeable for $g = 1.0$, e.g., TD_r is reduced by as much as 25% for $n \leq 10$, and by as much as 18% for n up to 20. The improvement is due to the reduced mean internode distance as well as an increased number of total links.

Nonuniform Traffic Distributions: From (3), (4), and (5), we are able to obtain D_{mean} of enhanced hypercubes for a given set of n , k , and g . D_{mean} as a function of k for $n = 10$ and 20 under $g = 1.2$ and 1.5 is depicted in Fig. 4. Under $g = 1.5$, it is minimum as $k = 0$ for $n = 10$; and as $k = 6$ for $n = 20$. To improve D_{mean} the most, skips should be created between pairs of nodes determined by n and g . In a system with $n = 20$, for example, node $(x_{19} \cdots x_{14} x_{13} \cdots x_0)$ is connected to node $(x_{19} \cdots x_{14} \bar{x}_{13} \cdots \bar{x}_0)$ to achieve the maximum improvement in D_{mean} , when g is 1.5. The minimum D_{mean} versus n for $g = 1.2$ and 1.5 is shown in Fig. 6, where the number next to each point indicates the corresponding k that leads to the minimum D_{mean} , and is termed k_{opt} . Under moderately nonuniform traffic distributions (i.e., $g = 1.2$), k_{opt} is no greater than 1, suggesting that the skip should be made to the farthest node (for an even-dimensional hypercube) or to the second farthest node (for an odd-dimensional hypercube). For hypercubes with heavily nonuniform traffic distributions (i.e., $g = 1.5$), k_{opt} starts to grow linearly when n exceeds 15.

In Fig. 5, the amount of D_{mean} reduction under various g 's is shown. It can be observed from this figure that the improvement amount decreases for a given system as g increases. This is because when the degree of reference locality becomes higher, skips diminish their impact on shortening message traversals accordingly, as messages then are highly unlikely to be directed

TABLE I
TRAFFIC DENSITY ON REGULAR LINK (TD_r) AND ON SKIPS (TD_s) UNDER VARIOUS g 's

Locality (g)	Traffic density	n (system size = 2^n)			
		5	10	15	20
1.0	TD_r	0.77	0.75	0.82	0.82
	TD_s	0.39	0.75	0.61	0.82
	TD_s/TD_r	0.75	0.75	0.82	0.82
1.2	TD_r	0.71	0.75	0.79	0.81
	TD_s	0.52	0.54	0.54	0.53
	TD_s/TD_r	0.75	0.82	0.87	0.89
1.5	TD_r	0.69	0.71	0.74	0.75
	TD_s	0.39	0.33	0.30	0.30
	TD_r/TD_s	0.79	0.89	0.93	0.94

(Note: TD_s is less than TD_r in an odd-dimensional hypercube, because from Eq. (6), the biggest term would happen at $l = \lceil (n-k)/2 \rceil$, but it is excluded.)

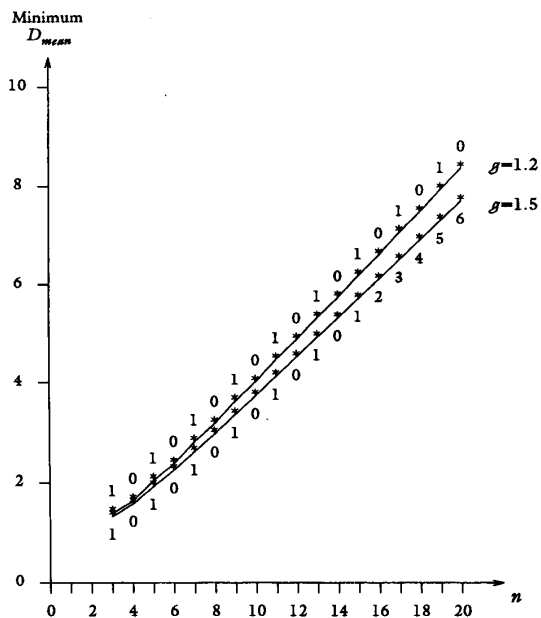


Fig. 6. Minimum mean internode distance versus system size ($= 2^n$) under nonuniform traffic distributions. (k_{opt} is shown for each given g and n .)

to far remote nodes. When g grows, an increasing number of messages terminate at closer nodes.

For a given enhanced hypercube, the ratio of traffic density improvement decreases for a larger g , as shown in Table I. This is because the improvement rate of D_{mean} declines as the degree of reference locality increases (see Fig. 5). From Table I, one can observe that a system with higher reference locality has a smaller TD_s . This is due to the fact that a diminishing amount of messages would traverse skips as g is raised. It is interesting to find from Table I that when g grows, traffic density over regular links also decreases, but not as significantly as over skips.

B. With Respect to Other Performance Measures

From Theorem 2, it is clear that $k = 0$ results in an optimal enhanced hypercube with respect to diameter under all traffic distributions, because diameter is independent of g 's.

We may arrive at optimal enhanced hypercubes with respect

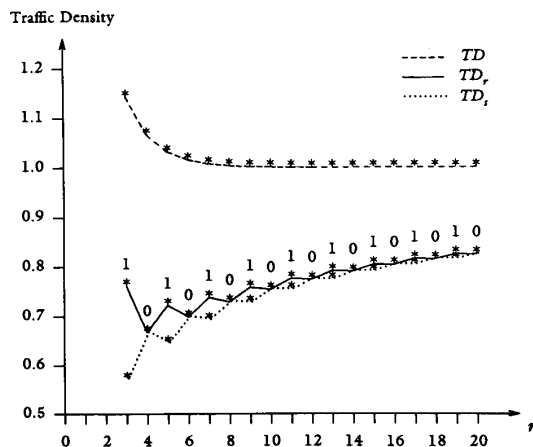


Fig. 7. Minimum traffic density (on regular links) versus system size ($= 2^n$) under uniform traffic distributions. (TD_s , TD , and k_{opt} for each n are also shown.)

to traffic density from (6) and (7). It can be shown that TD_s is less than or equal to TD_r for an arbitrary n , k , and g . If we attempt to keep the highest traffic density (which is TD_r) as low as possible, (7) is minimized to find out appropriate k 's.

In Figs. 7 and 8, minimum TD_r as a function of system size is sketched. Also given in the figures are TD_s , TD (traffic density over links in regular hypercubes), and the optimal k for each set of g and n . When reference locality is light to moderate (e.g., $g \leq 1.2$), k_{opt} is 0 for an even n and 1 for an odd n , for n up to 20. When $g = 1.5$, k_{opt} starts to grow linearly, as n is beyond 11. It can be seen that the improvement in traffic density is always considerable. The diameter and mean internode distance of such an optimal enhanced hypercube can be evaluated easily.

V. BROADCAST ALGORITHM FOR ENHANCED HYPERCUBES

The global broadcast operation is one of the most important and frequently used operations in hypercubes [18]. For example, it can be employed to efficiently initialize vectors, to implement distributed agreement/election [19] and clock synchronization [20], etc. In an enhanced hypercube, the broadcast operation can be done much faster by making use of the added skips. The considerable speedup in broadcast could lead to noticeable performance improvement. A node, as before, is assumed to be

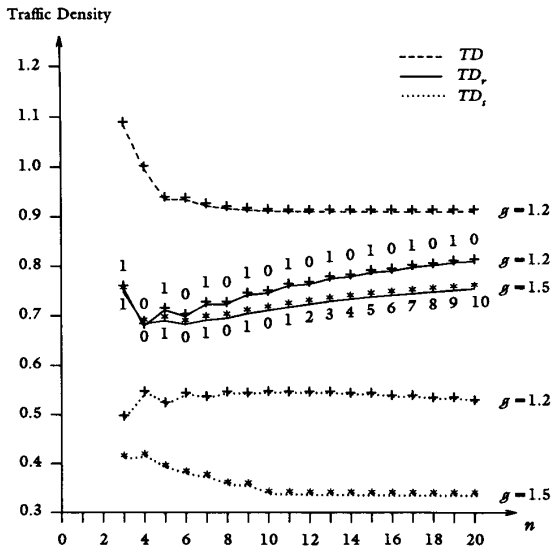


Fig. 8. Minimum traffic density (on regular links) versus system size ($= 2^n$) under nonuniform traffic distributions. (TD_s , TD , and k_{opt} for each n and g are also shown.)

capable of forwarding a broadcast message to all its outgoing links at a time.

The basic idea is that messages are now broadcast from a node to all the other nodes along two opposite directions at the same time. One direction starts from the originating node, say node X , to other nodes via regular links, following a similar process described in Algorithm B. The other direction is from the farthest node of X , node \bar{X} , to other nodes, after a message is forwarded from X to \bar{X} . This idea can be realized in an enhanced hypercube with given n and g as follows.

- 1) Assume that the skip is referred to as link n .
- 2) Each broadcast message carries a tag along with it. A tag is a 3-tuple $\langle op, index, count \rangle$. Field op with value "+" (or "-") indicates that the message is to be broadcast (or forwarded). Field $index$ serves as *weight* (see Algorithm B), if $op = "+"$, or denotes the link number through which the message is to be forwarded, if $op = "-"$. Field $count$ keeps track of the number of remaining link traversals (i.e., broadcast steps) a message needs to make before it stops. As a message advances one step along a broadcast direction, $count$ is subtracted by 1.
- 3) When a broadcast message is generated initially, its tag is prepared by the source node in the following manner. For a copy of the message that takes regular link l ($0 \leq l \leq n-1$), its tag is $\langle +, l, c \rangle$ with $c = k + \lceil (n-k)/2 \rceil$; whereas for the message that takes the skip, its tag is $\langle -, n-k, c' \rangle$ with $c' = \lfloor (n-k)/2 \rfloor$.
- 4) When a broadcast message reaches a node, the node responds differently, according to the tag of the message. If $op = "+"$ (which means that the message is to be broadcast) or $index = n$ (which means that the message has been forwarded to the farthest node, and from there on, the message is to be broadcast), the node, after keeping a copy, broadcasts the message through every link l such that l is less than $index$, provided $count$ is greater than 1; the tag of the message broadcast over link l is $\langle +, l, count-1 \rangle$. Otherwise, the node forwards the message with an

updated tag $\langle -, index+1, count \rangle$ through link $index$. In this case, the node does not keep a copy of the message.

It can be observed from the above statements that a broadcast message traverses one and only one skip, i.e., the skip of the originating node. It *never* passes through the skip of any other node. The broadcast algorithm for enhanced hypercubes is formally described below.

Algorithm D: (Broadcast a message with $tag = \langle op, index, count \rangle$ in an enhanced hypercube):

```

if (the node is the originating node)
  { send the message over each regular link  $l$  with a tag
     $\langle +, l, c \rangle$ .
    send the message over the skip with a tag  $\langle -, n-k, c' \rangle$ . }
else
  { if ( $op = "+"$ ) or ( $index = n$ )
    { keep a copy of the message at the node.
      if ( $count > 1$ )
        for each link  $l$  from this node with  $l$  less than  $index$ :
          send the message over link  $l$  with a tag  $\langle +, l,
            count-1 \rangle$ .
        }
      else
        { send the message with a tag  $\langle -, index+1, count \rangle$  over
          link  $index$ .
        }
    }
}

```

Fig. 9 illustrates an example of a message broadcast from node (00000) to all the other nodes in an enhanced hypercube with $n = 5$ and $k = 1$. A solid arrow originating from a node indicates that the node, after keeping a copy of the message, broadcasts the message, while a dashed arrow means that the node only forwards the message without keeping a copy. Every "sink" node also keeps a copy of the received message. Thus, node (01111) keeps a copy only when the message is from node (11111), which is the farthest node from node (00000).

The following theorem shows that Algorithm D sends a message nonredundantly to all the other nodes in $k + \lceil (n-k)/2 \rceil$ steps, equivalent to the diameter of the enhanced hypercube. A proof of this theorem can be found in the Appendix.

Theorem 3: Algorithm D broadcasts a message nonredundantly from a node to all the other nodes in an enhanced hypercube with given n and k in $k + \lceil (n-k)/2 \rceil$ steps.

It is clear from Theorem 3 that the broadcast time in enhanced hypercubes can be reduced as considerably as the diameter. When k is 0, for example, the broadcast time is only about one half of that in a regular hypercube with the same size. For a small k , the improvement is significant. It should be noted that the improvement amount is reduced if every node can forward a message to only a fixed, small number of outgoing links at a time.

VI. PRAGMATIC CONSIDERATIONS

It is known that in a multicomputer design, there is a tradeoff between the degree of a node and the diameter [11]–[13]. A structure with a low degree of nodes has a large diameter, and a structure that has a low diameter usually possesses a large degree of nodes. The (diameter \times degree of nodes) is therefore a good criterion for comparing different structures. In n -dimensional enhanced hypercubes, the degree of nodes is increased by $1/n$ whereas the diameter is reduced approximately by 50% under a uniform traffic distribution. It is clear that the measure of (diameter \times degree of nodes) is better (i.e., smaller) in enhanced hypercubes than in regular hypercubes. Likewise, under nonuniform traffic distributions with $g = 1.2$ or 1.5, it can

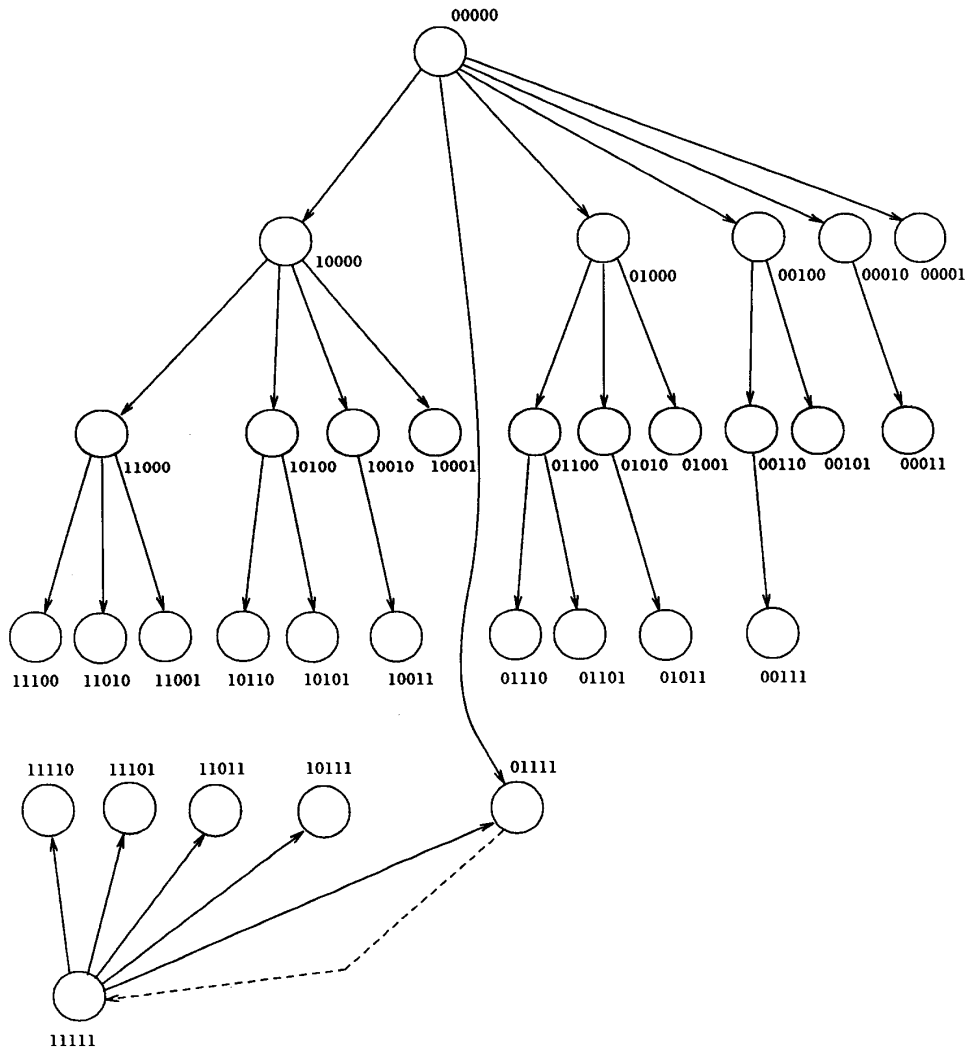


Fig. 9. Broadcast from node (0000) in an enhanced hypercube with $n = 5$ and $k = 1$.

also be seen that an enhanced hypercube has a better mentioned measure.

It is realized that skips in hypercubes can be created without incurring much additional hardware, as long as there is at least one link associated with each node left unused. For a hypercube with no unused links available, a natural question to ask is whether it is still beneficial to augment every node to accommodate an additional link for constructing the enhanced hypercube. To give an answer, we define the cost function and compute the cost-effectiveness measure of enhanced hypercubes in the following.

A. Cost-Effectiveness Measure

Assume that the package for nodes is not pin limited; then, a simple cost function for hypercubes may be defined [10] as

$$\text{cost} = C_{pe} \times \text{nodes} + C_{nl} \times \text{links} + C_{cc} \times \text{connections} \quad (8)$$

where C_{pe} , C_{nl} , and C_{cc} are the unit costs of a processing element (i.e., node), a node link, and a communication connection,

respectively. Here, we are interested in the case where the number of node links is the same as the number of communication connections (i.e., every available link is used). Since mean internode distance is more critical to network performance than diameter, as in [13], we adopt the product of cost and D_{mean} as a realistic cost-performance measure. A smaller value of the cost-performance measure is better.

To show the cost-effectiveness of enhanced hypercubes, we define the cost-performance ratio of regular hypercubes to enhanced hypercubes, represented by η . The case of $\eta > 1$ means that the enhanced hypercube is more cost-effective.

The upper half in Table II shows η of various enhanced hypercubes when $g = 1.0, 1.2,$ and 1.5 under the assumption that a processing element costs 10 times as much as a node link or a communication connection. Although the assumption is fairly conservative (because C_{pe} is normally far larger than C_{nl} and C_{cc}), it still leads to the result that η 's are always no less than 1. The lower half in Table II gives η obtained under the assumption that the unit cost for a processing element is 100 times the cost of a node link, whose cost is 10 times that of a communication

TABLE II
THE COST-EFFECTIVENESS OF ENHANCED HYPERCUBES

Unit cost ratio	Locality (g)	n (system size = 2^n)			
		5	10	15	20
$C_{pe} = 10 \times C_{nl}$ $C_{nl} = C_{ec}$	1.0	1.10	1.12	1.11	1.11
	1.2	1.07	1.07	1.05	1.05
	1.5	1.03	1.01	1.00	1.00
$C_{pe} = 100 \times C_{nl}$ $C_{nl} = 10 \times C_{ec}$	1.0	1.14	1.14	1.12	1.11
	1.2	1.10	1.08	1.06	1.05
	1.5	1.06	1.03	1.01	1.01

connection. This more realistic assumption results in η exceeding 1 for all the cases of interest (note that a larger cost ratio yields a bigger η). Consequently, enhanced hypercubes are preferable structures in the practical environments where reference locality is not extremely heavy (i.e., $g \leq 1.5$).

B. Implementation Issues

Algorithm C is employed for node-to-node routing in an enhanced hypercube. For an underpopulated hypercube, in addition to making skips through unused links, modifications at each node to install the new routing mechanism are necessary. If routing is microprogram controlled, the modifications require simply a new set of control code. If it is hardwired, an interface to bridge the gap between Algorithm A and Algorithm C is needed. The interface may be added to a node by placing it between the computation part and the communication part. This can be done without altering existing circuitry if the computation and communication parts of a node are distinguishable or separate (which is a common case). The interface serves to check whether a newly generated message has to make a traversal over the skip, when the message is first generated; it does not need to take care of pass-through traffic.

Suppose that the skip is connected to port w ($\geq n$). When a newly generated message is to be sent over the skip, the lower $n-k$ bits of its tag are complemented and bit w (corresponding to the skip port number) of the tag is set to 1, before being forwarded to the communication part, where the hardwired control would first route the message to port w . This requires a change on the hardwired control to make it examine bit w of the tag first. Such a change is simple because the control is designed to handle the extreme case where all ports are utilized. Therefore, to build an enhanced hypercube from a regular hypercube with unused ports is practically feasible.

While the cost-effective measure indicates that it is worthwhile to construct enhanced hypercubes even at the expense of augmenting each node to accommodate an additional link, in certain instances, to augment a node may be prohibitive because of the pin limitation or other constraints. For these situations, an enhanced hypercube cannot be constructed from a regular hypercube with no unused ports available.

VII. CONCLUDING REMARKS

A hypercube with extra connections added between pairs of nodes, dubbed an enhanced hypercube, has been investigated. Extra connections, called skips, can frequently be made by utilizing unused links left at each node in a real system, because each node is manufactured with a fixed number of links that permit

the construction of maximum allowable sizes. An enhanced hypercube requires essentially the same simple routing algorithm for regular hypercubes, making it easy for us to modify existing as well as future systems.

The criterion for making skips in hypercubes is to reduce a performance measure of interest as much as possible. The performance measures treated in this paper include diameter, mean internode distance, and traffic density over links. Optimal enhanced hypercubes with respect to any one of the measures are discussed and they always yield a respectable performance improvement. For example, under uniform traffic distributions, it is found that pairs of farthest nodes are connected so as to most reduce mean internode distance. As a result, the diameter is reduced by one half or so, and mean internode distance and traffic density are by noticeable amounts (e.g., by 17% and 25%, respectively, in a system with 1024 nodes). Under nonuniform traffic distributions, however, the two connected nodes need not lie farthest apart to have the largest improvement in mean internode distance. When the degree of reference locality increases, optimal skips would be established between pairs of closer nodes. As an enhanced hypercube enjoys several advantages, it appears profitable even for hypercube systems without unused links to augment each node so as to accommodate an extra link, especially when reference locality is not extremely severe (e.g., $g \leq 1.5$).

In general, if skips are made over pairs of nodes with $k = 0$ for an even dimensional hypercube, and with $k = 1$ for an odd dimensional hypercube, the improvement rates in mean internode distance and traffic density would be equal or close to the maximum ones, as long as reference locality is not extremely high (i.e., $g \leq 1.5$). Consequently, when n is even (or odd), creating a skip between a pair of nodes with $k = 0$ (or 1) can improve system performance optimally or near-optimally. Making skips this way is fairly promising and is recommended in general-purpose hypercubes, where the traffic distributions vary. This always leads to the maximum possible reduction in diameter (by approximately 50%), which makes possible a considerably faster broadcast operation, because a message can be broadcast at the same time along two opposite directions after a copy of the message reaches the farthest node (from the originating node through a skip).

Due to the presence of additional links, mapping processes onto nodes in an enhanced hypercube would be different from that in a regular hypercube, since far apart nodes now become nearby. It appears interesting to investigate the efficient mapping of processes to nodes in enhanced hypercubes. Different optimal structures may result when another performance measure (e.g., visit ratios) or practicality is of the uppermost concern. They

may adopt the proposed simple routing algorithm and provide reasonable performance improvements. It is also possible to add multiple extra connections to each node with a hope to further improve system performance, if multiple unused links are available at each node. The resulting hypercube is expected to need a more complex routing algorithm.

APPENDIX

Proof of Theorem 1: The hypercube of interest has a skip between $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ and $(x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_0)$. Suppose a message is to be transmitted from $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ to $(a_{n-1} \cdots a_{n-k} a_{n-k-1} \cdots a_0)$. Assume the message is at a node $(b_{n-1} \cdots b_{n-k} b_{n-k-1} \cdots b_0)$ after r traversals. Let the total number of differing bits between $b_{n-1} \cdots b_{n-k}$ and $a_{n-1} \cdots a_{n-k}$ be m_r^S , and between $b_{n-k-1} \cdots b_0$ and $a_{n-k-1} \cdots a_0$ be m_r^C .

I) $m_0^C > \lceil (n-k)/2 \rceil$: The message is transmitted to $(x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_0)$ through the skip, as described by the routing algorithm. Now, m_0^C changes to a value less than or equal to $\lceil (n-k)/2 \rceil$ and $m_1^S = m_0^S$ (because the leftmost k bits are unchanged). At that node, if the message takes the skip again, it returns back to the original node; so the message should take a regular link, as described by the routing algorithm. After taking the regular link, the message reaches a node with either $m_2^C = m_1^C - 1$ and $m_2^S = m_1^S$, or $m_2^C = m_1^C$ and $m_2^S = m_1^S - 1$.

Consider the case of $m_2^C = m_1^C - 1$ and $m_2^S = m_1^S$ first. Since $m_2^C < \lceil (n-k)/2 \rceil$, the message should take a regular link there; otherwise, it will be directed to a node with $m_3^C \geq \lceil (n-k)/2 \rceil$, an increased value. Similarly, for the case of $m_2^C = m_1^C$ and $m_2^S = m_1^S - 1$, m_3^S would become greater than $\lceil (n-k)/2 \rceil$ if the message takes the skip there. Hence, a regular link should be taken to avoid increasing m_3^C .

By the same token, we can show that the message has to make the j th traversal, for all $j \geq 1$, along a regular link, as described by the routing algorithm, and the traversed path would be the shortest possible since each traversal effectively corrects one differing bit between the source address and the destination address. A message over a shortest path to its destination would pass through only one skip.

II) $m_0^C \leq \lceil (n-k)/2 \rceil$: This situation can easily be proved that no skip traversal is required for a message to reach its destination along a shortest path. The theorem is thus proved. \square

Proof of Theorem 2: Since the diameter is the maximum distance between any two nodes, we can obtain the diameter of an enhanced hypercube by examining what kind of nodes are farthest away from a given node.

There is a skip between every pair of nodes, $(x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$ and $(x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_0)$, in an enhanced hypercube with a given k . Assume that the node $(a_{n-1} \cdots a_{n-k} a_{n-k-1} \cdots a_0)$ is one of the farthest nodes from node $S = (x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots x_0)$. Then, it is easy to see that its most significant k bits, $a_{n-1} \cdots a_{n-k}$, must be equal to $\bar{x}_{n-1} \cdots \bar{x}_{n-k}$, no matter whether the skip of node S is used or not. Now consider the remaining $(n-k)$ bits of the two nodes' addresses. Suppose there are l differing bits between $a_{n-k-1} \cdots a_0$ and $x_{n-k-1} \cdots x_0$. Due to the existence of the skip, the distance from node S to $(x_{n-1} \cdots x_{n-k} a_{n-k-1} \cdots a_0)$ equals $\min(l, 1 + n - k - l)$, which is maximum when $l = \lfloor (1 + n - k)/2 \rfloor = \lceil (n-k)/2 \rceil$.

The maximum distance between node $(a_{n-1} \cdots a_{n-k} a_{n-k-1} \cdots a_0)$ and node S is the maximum number of differing bits between $a_{n-1} \cdots a_{n-k} a_{n-k-1} \cdots a_0$ and $x_{n-1} \cdots x_{n-k} x_{n-k-1} \cdots$

x_0 , which is equivalent to the maximum number of differing bits between $a_{n-1} \cdots a_{n-k}$ and $x_{n-1} \cdots x_{n-k}$ plus the maximum number of differing bits between $a_{n-k-1} \cdots a_0$ and $x_{n-k-1} \cdots x_0$, which in turn equals $k + \lceil (n-k)/2 \rceil$. Thus, the diameter of the enhanced hypercube is $k + \lceil (n-k)/2 \rceil$. \square

Proof of Theorem 3: Let the originating node be $X = (x_{n-1} \cdots x_0)$. There is a skip connecting node X and node $X' = (x_{n-1} \cdots x_{n-k} \bar{x}_{n-k-1} \cdots \bar{x}_0)$. From the algorithm, we know that during the first k steps, all the nodes with k bits differing from X receive the message and keep a copy, while nodes X' , $(x_{n-1} \cdots x_{n-k+1} \bar{x}_{n-k} \cdots \bar{x}_0), \dots, (x_{n-1} \bar{x}_{n-2} \cdots \bar{x}_0)$ receive the message but only forward it without keeping a copy. At the $(k+1)$ step, all the nodes with $(k+1)$ bits differing from X receive the message and keep a copy, as does node X' . After that, the message is broadcast along two opposite directions simultaneously, i.e., nodes with $k+2, k+3, \dots, k + \lceil (n-k)/2 \rceil$ bits differing from X will receive the message and keep a copy step by step, and at the same time, nodes with $n-1, n-2, \dots, n - \lfloor (n-k)/2 \rfloor - 1$ bits differing from X also take the same action.

It is obvious that no node keeps duplicated copies of the message if the broadcast process can stop altogether after performing a total of n broadcast steps along the two directions. The process is controlled by the value of *count* in the tag. Note that *count* is decreased (by one) only when the message is broadcast, and is unchanged when it is forwarded. The initial value of *count* in the tag of a message through a regular link of X is c , and through the skip of X is c' . So, there are c broadcast steps starting from X (including X itself) along one direction together with c' broadcast steps starting from X' (including X' itself) along the other direction before the process stops. The total number of steps along both directions is $c + c' = n$. Every node thus accepts the message exactly once. Since c is no less than c' , to broadcast a message in the enhanced hypercube takes $c = k + \lceil (n-k)/2 \rceil$ steps, equal to its diameter. \square

REFERENCES

- [1] C. L. Seitz, "The Cosmic Cube," *Commun. ACM*, vol. 28, no. 1, pp. 22-33, Jan. 1985.
- [2] W. D. Hillis, *The Connection Machine*. Cambridge, MA: MIT Press, 1985.
- [3] Intel Corporation, *iPSC System Overview*, Jan. 1986.
- [4] J. P. Hayes, T. N. Mudge, and Q. F. Stout, "Architecture of a hypercube supercomputer," in *Proc. 1986 Int. Conf. Parallel Processing*, Aug. 1986, pp. 653-660.
- [5] W. C. Athas and C. L. Seitz, "Multicomputers: Message-passing concurrent computers," *IEEE Comput. Mag.*, vol. 21, pp. 9-24, Aug. 1988.
- [6] L. W. Tucker and G. G. Robertson, "Architecture and applications of the Connection Machine," *IEEE Comput. Mag.*, vol. 21, pp. 26-38, Aug. 1988.
- [7] W. T.-Y. Hsu, P.-C. Yew, and C.-Q. Zhu, "An enhancement scheme for hypercube interconnection networks," in *Proc. 1987 Int. Conf. Parallel Processing*, Aug. 1987, pp. 820-823.
- [8] D. A. Reed and D. C. Grunwald, "The performance of multicomputer interconnection networks," *IEEE Comput. Mag.*, vol. 20, pp. 63-73, June 1987.
- [9] H. Sullivan and T. R. Bashkow, "A large scale homogeneous, fully distributed parallel machine, I," in *Proc. 4th Symp. Comput. Architecture*, Mar. 1977, pp. 105-117.
- [10] D. A. Reed and H. D. Schwetman, "Cost-performance bounds for multimicrocomputer networks," *IEEE Trans. Comput.*, vol. C-32, pp. 83-95, Jan. 1983.
- [11] L. D. Wittie, "Communication structures for large networks of microcomputers," *IEEE Trans. Comput.*, vol. C-30, pp. 264-273, Apr. 1981.
- [12] P. W. Dowd and K. Jabbar, "Spanning multiaccess channel hypercube computer interconnection," *IEEE Trans. Comput.*, vol. C-37, pp. 1137-1142, Sept. 1988.

- [13] L.N. Bhuyan and D.P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," *IEEE Trans. Comput.*, vol. C-33, pp. 323-333, Apr. 1984.
- [14] J.P. Hayes *et al.*, "A microprocessor-based hypercube supercomputer," *IEEE Micro*, vol. 6, pp. 6-17, Oct. 1986.
- [15] J.C. Peterson *et al.*, "The Mark III hypercube-ensemble concurrent computer," in *Proc. 1985 Int. Conf. Parallel Processing*, Aug. 1985, pp. 71-73.
- [16] A. Esfahanian, L.M. Ni, and B.E. Sagan, "On enhancing hypercube multiprocessors," in *Proc. 1988 Int. Conf. Parallel Processing*, Aug. 1985, pp. 86-89.
- [17] D.P. Agrawal and V.K. Janakiram, "Evaluating the performance of multicomputer configurations," *IEEE Comput. Mag.*, vol. 19, pp. 23-37, May 1986.
- [18] S.L. Johnsson and C.-T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Comput.*, vol. 38, pp. 1249-1268, Sept. 1989.
- [19] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Programming Languages Syst.*, vol. 4, pp. 382-401, July 1982.
- [20] T.K. Srikanth and S. Toueg, "Optimal clock synchronization," *J. ACM*, vol. 34, pp. 626-645, July 1987.



Sizheng Wei (S'90) received the B.S. and M.S. degrees in computer science from the Shanghai Jiao Tong University, China, in 1982 and 1985, respectively, and the M.S. degree in computer engineering from the University of Southwestern Louisiana, Lafayette, in 1988.

He is currently pursuing the Ph.D. degree in the Department of Computer Science, Rutgers University, NJ. His research interests include computer architecture, parallel and distributed systems, and fault-tolerant computing.



Nian-Feng Tzeng (S'85-M'86) received the B.S. degree in computer science from National Chiao Tung University, Taiwan, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1978, 1980, and 1986, respectively.

He is currently an Assistant Professor in the Center for Advanced Computer Studies at the University of Southwestern Louisiana, Lafayette, where he has been on the faculty since June 1987. From 1986 to 1987, he was a member

of the Technical Staff, AT&T Bell Laboratories, Columbus, OH. He had been involved in the Cedar supercomputer project at the Center for Supercomputing Research and Development, University of Illinois, Urbana, for more than two years. His current research interest is in the areas of parallel and distributed processing, fault-tolerant computing, and VLSI/WSI-based systems.

Dr. Tzeng is a member of Tau Beta Pi, the Association for Computing Machinery, and the recipient of the Outstanding Paper Award of the Tenth International Conference on Distributed Computing Systems, May 1990.