Traffic Analysis and Simulation Performance of Incomplete Hypercubes

Nian-Feng Tzeng, *Senior Member, IEEE*, and Harish Kumar

Abstract—The incomplete hypercube with arbitrary nodes provides far better incremental flexibility than the complete hypercube, whose size is restricted to exactly a power of 2. After faults arise in a complete hypercube system, it is desirable to reconfigure the system so as to retain as many healthy nodes as possible, often leading to an incomplete hypercube of arbitrary size. In this paper, the highest traffic density over links in an incomplete hypercube under uniform message distribution is shown to be bounded by 2 (messages per link per cycle), independent of its size and despite its structural nonhomogeneity. As a result, it is easily achievable to construct an incomplete hypercube with sufficient link communication capability where any potential points of congestion are avoided, ensuring high performance. Simulation results for the incomplete hypercube reveal that mean latency for delivering messages is roughly the same in an incomplete hypercube as in a compatible complete hypercube under both packet-switching and wormhole routing. The incomplete hypercube thus appears to be an attractive and practical architecture, since it shares every advantage of complete hypercubes while eliminating the restriction on the system size.

Index Terms—Incomplete hypercubes, mean latency, message routing, simulation, traffic density.

1 Introduction

For a large multiprocessor system, the interconnection architecture used to connect processors together critically dictates system performance. While various interconnection schemes have been proposed [1], [2], the binary hypercube is a powerful interconnection topology due to its many attractive features [7] and good support of numerous parallel algorithms [3]. Parallel machines based on this topology have been built and are commercially available [3], [4], [5], [6], [11]

It is desirable that an interconnection scheme allow any sized construction, offering maximum incremental flexibility. The hypercube topology, however, can interconnect exactly 2^n nodes only (n is a positive integer), severely restricting allowable system sizes. A flexible version of the hypercube topology, called the *incomplete hypercube* [8], eliminates the restriction on the node number and thus makes it possible to construct parallel machines with arbitrary sizes. In a large hypercube system, operational faults happen with a nonnegligible probability, and it is commonly advisable to reconfigure the system in response to operational faults such that after reconfiguration, the system retains as many workable nodes as possible. If only complete hypercubes are allowed, then a reconfigured system loses a considerable amount of nodes even in the presence of a single fault, simply due to the strong restriction on the system size that results in many healthy node being unnecessarily discarded. On the other hand, if incomplete hypercubes are considered acceptable, then after a fault arises in a complete hypercube, a reconfigured system reduces its size by just one, giving rise to a larger system. It should be noted that although programming on a hypercube often has the cube topology assumed, a reconfigured incomplete hypercube has a performance edge over a reconfigured complete hypercube, because the former always involves a copy of the latter plus some smaller sized cube(s) and thus may execute multiple jobs of different sizes simultaneously.

Simple and deadlock-free algorithms for routing and broadcasting messages in the incomplete hypercube have been developed by Katseff [8]. The structural properties of a class of incomplete hypercube systems have been studied recently [9]. The system under that study was limited to a size of $2^n + 2^k$, $0 \le k < n$, i.e., an incomplete hypercube composed of two complete cubes. It was shown [9] that the highest traffic density in such an incomplete hypercube is no more than 2 (messages per link per cycle). Efficient broadcasting based on edge-disjoint spanning trees in that type of incomplete hypercubes is presented and analyzed by Tien, Ho, and Yang [13]. Tree embedding in such an incomplete hypercube is pursued in [16].

Unlike a complete hypercube, the incomplete hypercube under consideration is asymmetric in nature, because cube nodes no longer have the same degree and they play different roles. It is recognized that in general, an asymmetric structure, such as the tree, star, and any other irregular topology, tends to have one or several excessively loaded links or nodes that may become vulnerable points with respect to performance and reliability. We are interested in finding out whether or not there is any vulnerable point present in the incomplete hypercube. If the incomplete hypercube exhibits no point of vulnerability, it not only manifests itself as an interesting and important topology, but also potentially makes the complete hypercube more useful after operational faults occur by retaining more healthy nodes.

In this paper, we analyze traffic density over links in an incomplete hypercube with arbitrary nodes under the uni-

• *N.-F. Tzeng is with the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, LA 70504. E-mail: tzeng@cacs.usl.edu.*
• *H. Kumar is with the Intel Corporation, Beaverton, Oregon.*

Manuscript received Dec. 12 1993; revised May 26, 1995.
For information on obtaining reprints of this article, please send e-mail to: transpds@computer.org, and reference IEEECS Log Number D95200.

1045-9219/96$05.00 ©1996 IEEE

form message distribution to get a better insight into this flexible architecture. Interestingly, link traffic density in any sized incomplete hypercube is found to be bounded by 2, regardless of its structural nonhomogeneity. Therefore, cube links can easily be designed to prevent a traffic bottleneck from arising. This suggests that the incomplete hypercube has a clear advantage over other nonhomogeneous topologies, such as trees and stars, where points of congestion are likely to exist and serious performance degradation possibly results, because the highest traffic density in such a topology is proportional to its size (as shown in [1]). Simulation studies have been carried out to evaluate incomplete hypercube performance under packet-switching and wormhole routing when the queueing delay is taken into account, with mean latency and throughput chosen as the performance measures. Mean latency for sending a message from one node to another arbitrary node in incomplete hypercubes has been obtained and compared with that in the complete counterparts. Simulation behaviors under nonuniform message distributions are also examined and discussed.

This paper is organized as follows. Section 2 introduces necessary nomenclature and background to facilitate the subsequent presentation. A brief review of the routing algorithm proposed for incomplete hypercubes [8] is also given. Section 3 analyzes the incomplete hypercube to get its basic characteristics and Section 4 derives the upper bound of link traffic density. Simulation results are presented in Section 5.

2 Notations and Background

An n-dimensional complete hypercube, denoted by H_n, comprises 2^n nodes, each with n links connected directly to n nearest neighbors. Nodes in H_n are numbered from 0 to $2^n - 1$, by n-bit binary numbers $(x_{n-1} \cdots x_i \cdots x_0)$ as their *addresses*. A node and a nearest neighbor have exactly one bit differing in their addresses. A d-dimensional subcube in H_n contains exactly d *don't care* bits (denoted by *'s) in its binary address representation. For instance, like $(0*^{n-1})$ and $(1*^{n-1})$, $(*0*^{n-2})$ and $(*^i 1*^{n-1-i})$ are example $(n-1)$-dimensional subcubes in H_n, where $\times^l$ represents l consecutive $\times$.

The incomplete hypercube under consideration comprises multiple complete cubes of distinct dimensions. Fig. 1 shows an incomplete hypercube comprising three complete cubes H_4, H_3, and H_1. Each node address involves 5 bits and the three constituent cubes are addressed by $(0*^4)$, $(10*^3)$, and $(1100*)$, respectively. A link exists between a node A in H_4 and another node B in $H_3 \cup H_1$, if the addresses of A and B differ in bit 4. Similarly, a link is present between a node in H_3 and another node in H_1, if the two node addresses differ in bit 3. This incomplete hypercube is denoted by I_5^{26}, where 5 and 26 are the system dimension and the total number of nodes in the system, respectively.

In general, an n-dimensional incomplete hypercube with M nodes, I_n^M, $2^{n-1} \le M < 2^n$, can be defined recursively as follows: I_n^M consists of two components, H_{n-1} and $I_k^{M-2^{n-1}}$ $(k = \lceil \log_2 (M - 2^{n-1}) \rceil)$, with nodes in H_{n-1} numbered from 0 to $2^{n-1} - 1$ and nodes in $I_k^{M-2^{n-1}}$ numbered from 2^{n-1} to $M - 1$; a link exists between a node A in H_{n-1} and another node B in $I_k^{M-2^{n-1}}$, if and only if the addresses of A and B differ in bit $n - 1$. I_n^M is characterized by a bit vector $V_n^M = < 1, x_{n-2}, x_{n-3}, \cdots, x_i, \cdots, x_1, x_0 >$ such that x_i equals 1 (or 0) if H_i is present (or absent) in I_n^M. Clearly, $2^{n-1} + x_i \sum_{i=0}^{n-2} 2^i = M$ and the bit vector is the binary representation of M. The incomplete hypercube given in Fig. 1, for example, is characterized by bit vector $V_5^{26} = < 1\ 1\ 0\ 1\ 0 >$. Note that an incomplete hypercube reconfigured from a complete hypercube does not necessarily contain all the healthy nodes, and it often requires renumbering the constituent nodes. A method provided in [10] can be used for this renumbering.

The link between two neighboring nodes A and B is denoted as λ_B^A. A message traveling from one node to a nearest neighbor is called a *traversal*. Let $D(A, B)$ represent the number of differing bits between the node A address and the node B address, i.e., their Hamming distance. It is apparent that the number of traversals required from node A to node B is $D(A, B)$. The *relative address* of two nodes is the bitwise Exclusive-OR of their addresses. A link is said to have *link number i* if it connects two nodes whose addresses differ only in the ith bit position, denoted by $\lambda|_i$. The link between nodes (1101) and (0101), for example, is referred to as link 3, namely, $\lambda|_3$.

An important property of a topology is *traffic density* over links (denoted by TD), which indicates the average number of messages traversing a link during one unit time (i.e., cycle) and reflects link utilization. A topology with low traffic density is preferable because it would avoid any potential communication bottleneck and reduce processing/queueing delay when messages visit a node. TD is dependent upon the message distribution, which describes

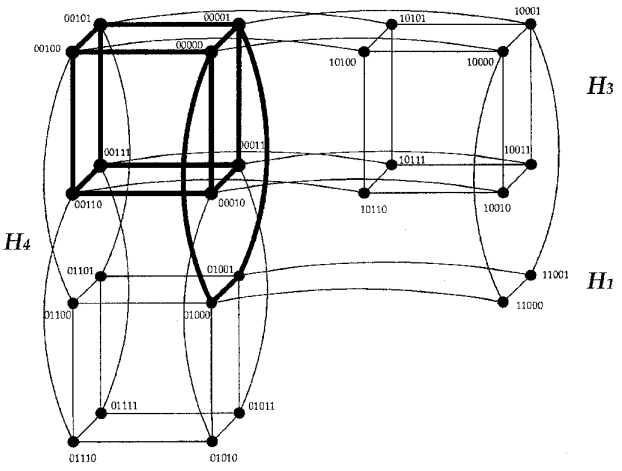

Fig. 1. An incomplete hypercube with 26 nodes, I_5^{26}. Ψ_4 is shown bold.

the probability of message exchanges among nodes. In our analytic study, a *uniform* message distribution is assumed, i.e., the average rate at which node A sends messages to node B is the same for all nodes A and B, where $A \neq B$. TD in the incomplete hypercube under a uniform message distribution will be analyzed subsequently.

Routing Algorithm for Incomplete Hypercubes

The routing algorithm for incomplete hypercubes by Katseff [8] is similar to *e*-cube routing and always finds shortest paths for messages.

ALGORITHM R. (send or forward a message from node *src* to node *dest* with $tag \leftarrow src \oplus dest$ in an incomplete hypercube).

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

A message carries with it a routing tag and is sent through link i only if bit i is the least significant nonzero bit in the tag *and* link i exists. Algorithm R checks the routing tag leftwards, starting from the least significant bit.

3 Characteristics of the Incomplete Hypercube

As opposed to those in a complete hypercube, nodes in an incomplete hypercube no longer play an identical role. For example, nodes in subcubes (00***) and (0100*) inside constituent cube H_4 of I_5^{26} shown in Fig. 1 have five links each (while the other nodes inside H_4 have four links each) and are connected to corresponding nodes in subcubes (10***) and (1100*), respectively. The nodes in subcubes (00***), (0100*), and (1000*) are particular (for having more links than other nodes in their respective constituent cubes) and are referred to as *pivot* nodes. A set of pivot nodes is generally defined as follows: for constituent cubes H_i and H_j, $i > j$, of I_n^M, the *pivot nodes* associated with the two cubes involve the collection of nodes $P_{i,j}$ = {node x | node x in H_i *and* node x has a direct link connected to a node in H_j}.

From our construction of incomplete hypercubes, it is clear that the direct link between a node in $P_{i,j}$ and another node in H_j is $\lambda|_i$. In Fig. 1, for example, $P_{4,3}$ is the subcube (00***), and the direct link connecting a node in $P_{4,3}$ and another node in H_3 is $\lambda|_4$; whereas $P_{3,1}$ is (1000*), and the direct link between a node in $P_{3,1}$ and a node in H_1 is $\lambda|_3$.

LEMMA 1. *Suppose H_i, H_j, and H_k are among the constituent cubes of I_n^M, with $i > j, k$. The two sets of pivot nodes associated respectively with H_i and H_j and with H_i and H_k, $P_{i,j}$ and $P_{i,k}$, satisfy $P_{i,j} \cap P_{i,k} = \varnothing$.*

This lemma is obvious since the links connecting a node in $P_{i,j}$ to a node in H_j, and the links connecting a node in $P_{i,k}$ to a node in H_k are both of type $\lambda|_i$, and each node in H_i has exactly one link $\lambda|_i$. Considering Fig. 1, for example, we have $P_{4,3}$ = (00***) and $P_{4,1}$ = (0100*), which have no common node.

Based on the recursive definition of incomplete hypercubes and the above result, we arrive at an abstract structure for I_n^M as depicted in Fig. 2, where arrows indicate links between constituent cubes, with the link type and the number of present links given next to the arrow. The next observation results from the abstract structure immediately.

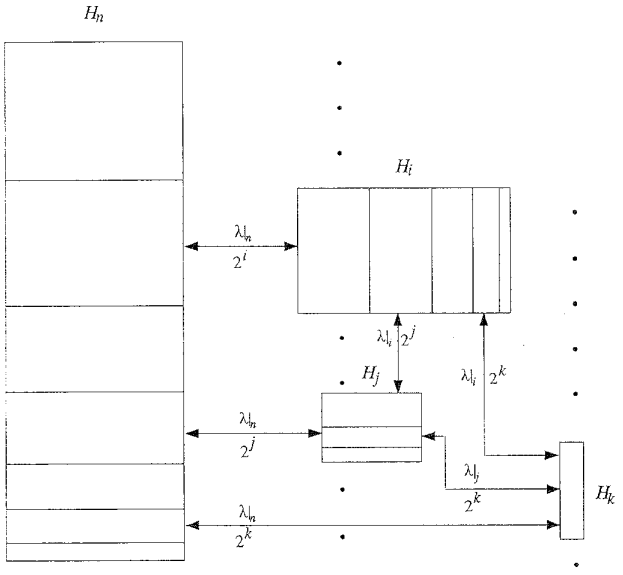

Fig. 2. An abstract structure of an incomplete hypercube I_n^M.

OBSERVATION 1. A node in the constituent cube H_i of I_n^M has the following links present: 1) $\lambda|_x$ for all $x < i$, 2) $\lambda|_y$ for any $y > i$ such that the yth bit in the bit vector V_n^M is "1," and 3) $\lambda|_i$ only if the node belongs to any pivot set $P_{i,j}$.

In Fig. 1, for instance, a node in H_3 has 1) $\lambda|_0$, $\lambda|_1$, and $\lambda|_2$, which form the connections inside H_3, 2) $\lambda|_4$, which connects to the "higher" constituent cube (i.e., a higher dimensional cube) H_4, and 3) $\lambda|_3$ only if the node is in $P_{3,1}$. Generally speaking, a node in H_i is connected to a node in a "higher" constituent cube H_h by $\lambda|_h$, whereas it is connected to a node, if any, in a "lower" constituent cube by $\lambda|_i$. Consider two pivot sets $P_{i,j}$ and $P_{i,k}$ in constituent cube H_i, $i > j > k$. It can also be observed that every node in $P_{i,k}$ is connected to a corresponding node in $P_{i,j}$ by using link $\lambda|_j$. As an example, nodes in $P_{4,1}$ are connected to corresponding nodes in $P_{4,3}$ through link $\lambda|_3$'s inside H_4 of Fig. 1. Let $\Phi(< i)$ be the set of all constituent cubes H_j, $j < i$, together with all the intercube links among these cubes inside I_n^M. Note that $\Phi(< i)$ itself forms an incomplete hypercube. The following lemma describes an interesting configuration

property of the incomplete hypercube. This property is useful for later traffic density derivation.

LEMMA 2. *Let ψ_i be the collection of all pivot nodes associated with H_j, for all $j < i$, plus all the connections among these nodes inside constituent cube H_i, then $\Phi(< i)$ is isomorphic to ψ_i.*

This lemma can be proved according to the facts that 1) every constituent cube H_j, $j < i$, is associated with a set of 2^j pivot nodes, $P_{i,j}$, in H_i, and 2) the connections between any two constituent cubes H_j and H_k (for all $j, k < i$) are the same as those between pivot node sets $P_{i,j}$ and $P_{i,k}$. The implication of Lemma 2 is that the structure of $\Phi(< i)$ is reflected by ψ_i inside H_i. In Fig. 1, for example, $\Phi(< 4)$ has the same structure as Ψ_4 inside H_4, illustrated by bold lines.

We next develop a partitioning concept which helps to illustrate the behavior of message traversals. Let $y_{u-1}y_{u-2} \cdots y_1y_0$ be a bit string and $O_w(y_{u-1}y_{u-2} \cdots y_1y_0, j)$ be the position of the wth nonzero bit after (i.e., right of) position j in the bit string. For example, $O_1(11010, 3) = 1$, $O_1(11101, 4) = 3$, and $O_2(11101, 4) = 2$. Consider a collection Ω_i of subcubes in constituent cube H_i of I_n^M with bit vector

$$< a_{n-1}, a_{n-2}, \cdots, a_{i+1}, a_i, a_{i-1}, a_{i-2}, \cdots, a_1, a_0 >:$$

$$\Omega_i = \{(a_{n-1}a_{n-2} \cdots a_{i+1}0x_{i-1}x_{i-2} \cdots x_{m+2}x_{m+1}{}^{*^{m+1}}) \mid$$

$$m = O_1(a_{n-1}a_{n-2} \cdots a_{i+1}a_ia_{i-1} \cdots a_1a_0, i), \text{ and}$$

$$x_{i-1}x_{i-2} \cdots x_{m+2}x_{m+1} \text{ denotes any possible bit string}\}.$$

Every element in Ω_i is an $(m + 1)$-dimensional subcube and each node in H_i belongs to one and only one element (subcube). One may view Ω_i as a result of partitioning H_i into identically sized subcubes, with a total of 2^{i-m-1} subcubes. As an example, Ω_3 of I_5^{26} shown in Fig. 1 is $\{(100^{**}), (101^{**})\}$, as m equals 1 in this case. Let T_i^p be an element in Ω_i, namely $(a_{n-1}a_{n-2} \cdots a_{i+1}0x_{i-1}x_{i-2} \cdots x_{m+1}{}^{*^{m+1}})$, such that the value of $x_{i-1}x_{i-2} \cdots x_{m+1}$ is p. Then, we have the following lemma, whose proof can be found in the appendix.

LEMMA 3. *All the pivot nodes in any constituent cube H_i of I_n^M lie in the single subcube T_i^0.*

An abstract structure of the partitioned H_i is illustrated in Fig. 3, where all $\lambda|_i$'s (for connecting nodes in H_j, $i > j$) terminate at T_i^0. Every constituent cube of an incomplete hypercube can be partitioned accordingly, with one partition involving all the pivot nodes in the hypercube. Notice that, due to the recursive nature of an incomplete hypercube, the way of partitioning nodes in H_i highlighted above can be applied to further partition nodes inside T_i^0 into subcubes each with the size of $2^{m'+1}$, where

$$m' = O_2(a_{n-1}a_{n-2} \cdots a_{i+1}a_ia_{i-1} \cdots a_1a_0, i),$$

when I_n^M with bit vector

$$< a_{n-1}, a_{n-2}, \cdots, a_{i+1}, a_i, a_{i-1}, \cdots, a_1, a_0 >$$

is concerned. The partition in T_i^0 which involves $P_{i,j}$ and $P_{i,k}$, for all $k < j$, is denoted by Π_i^{j+1}, as depicted in Fig. 3. The partitioning concept helps to understand the behavior of message traversals.

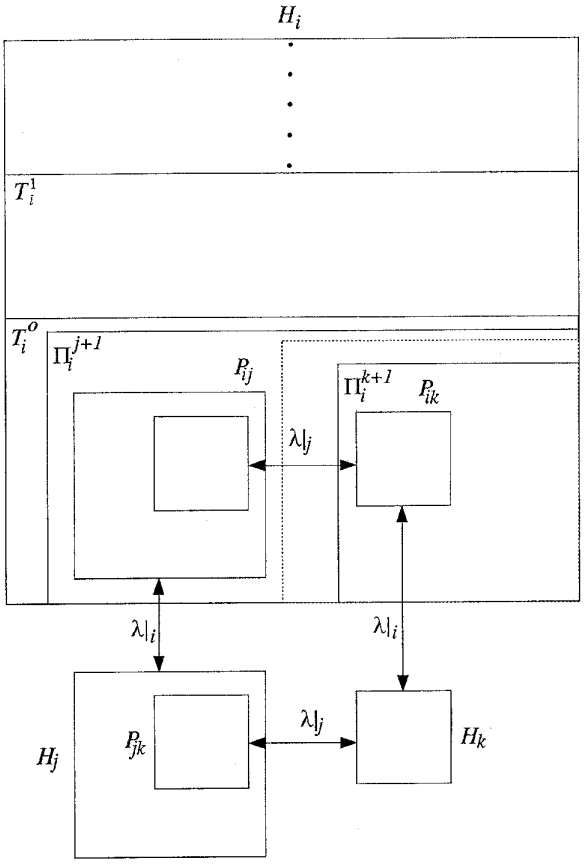

Fig. 3. The abstract structure of H_i showing T_i^0 and its recursive partitioning into Π_i^j's.

Let a message coming from a node in constituent cube H_j be denoted by $\mu|_{\leftarrow H_j}$. Messages in the incomplete hypercube are routed by Algorithm R given in Section 2. The following lemmas are helpful for the derivation carried out in the next section. It should be noted that a set of similar lemmas can be derived if messages are routed by another deadlock-free algorithm different from Algorithm R.

LEMMA 4. *Consider any two constituent cubes H_i and H_j, $i > j$, of I_n^M. A message $\mu|_{\leftarrow H_j}$ may enter any partition in H_i at most once, where a partition refers to an element in collection Ω_i.*

A proof of this lemma is straightforward and thus omitted. It is clear that a given message may or may not visit partitions in a constituent cube. If the message visits a partition, according to this lemma, it never reenters the partition after leaving the partition.

The next lemma results directly from the fact that the structure of $\Phi(< i)$ also exists in constituent cube H_i, and thus a message will traverse links in H_i, if needed, to "correct" nonzero tag bits with positions lower than i before taking a link $\lambda|_i$ (which is the link for connecting H_i and $\Phi(< i)$, see Observation 1). In other words, if a message is destined for a node in $\Phi(< i)$, it reaches $\Phi(< i)$ directly via its destination node; it never passes through any other node in $\Phi(< i)$, as described below. This result enables us to preclude certain messages from contributing traffic over a given link in our later traffic density derivation.

LEMMA 5. *A message $\mu|_{\leftarrow H_i}$ destined for a node in $\Phi(< i)$ never traverses any link in $\Phi(< i)$.*

If tag bit j of a message $\mu|_{\leftarrow H_k}$, $k < j$, is nonzero and H_j is present, then the message must visit H_j, regardless of whether the message is destined for a node in H_j or not. This is because 1) every node in $\Phi(< j)$ is connected to a corresponding node in H_j through a link $\lambda|_j$ and no link $\lambda|_j$ exists inside $\Phi(< j)$ (from Observation 1), and 2) the message is routed, according to Algorithm R, in a way that the nonzero tag bit j is corrected before nonzero tag bits p, $p > j$, if any. This fact is provided in the next lemma, which helps determine the messages that contribute traffic over a given link in our traffic density derivation.

LEMMA 6. *Any message $\mu|_{\leftarrow H_k}$ with nonzero tag bit j, $j > k$, must visit H_j, provided that H_j exists.*

Suppose $\Phi(\geq i)$ is the set of all constituent cubes H_j, $j \geq i$, together with all the intercube links among these cubes inside I_n^M. Let a message issued at node X in $\Phi(< i)$ and destined for node X' in $\Phi(\geq i)$ be denoted by $\mu^{X \to X'}$, $X \in \Phi(< i)$ and $X' \in \Phi\ (\geq i)$. Now consider a message $\mu^{X \to X'}$. It is clear from the above lemma that this message will leave $\Phi(< i)$ only when there exists no nonzero tag bit q for which a corresponding cube H_q exists in $\Phi(< i)$. This conclusion may be formally stated as follows.

LEMMA 7. *Given H_q is present in $\Phi(< i)$, a message $\mu^{X \to X'}$, $X \in \Phi(< i)$ and $X' \in \Phi(\geq i)$, leaves $\Phi(< i)$ via a node A inside H_j, $j < q < i$, if and only if there exists no nonzero bit q in the tag upon its exit from A.*

The isomorphism between Ψ_i and $\Phi(< i)$, as shown in Lemma 2, along with the above lemma leads us to the next lemma (which is similar in form to Lemma 5).

LEMMA 8. *A message $\mu^{X \to X'}$, $X \in \Phi(< i)$ and $X' \in \Phi(\geq i)$, never traverses any link in Ψ_i.*

Suppose H_m is the constituent cube immediately below H_i, then, a proof of this lemma follows from the fact that a message issued at a constituent cube lower than H_i, before entering H_i, must correct all its nonzero tag bits with positions either lower than $m + 1$, if a constituent cube below H_m exists; or lower than m, otherwise. After arriving at H_i, the message traverses no link which has a corresponding link present in $\Phi(< i)$, and thus no link in Ψ_i.

4 Deriving the Highest Traffic Density

Traffic density over links in I_n^M is not fixed and is location-dependent. The highest traffic density is of our major concern, since it tends to dictate the longest time taken to traverse a link, and thus the worst message transmission scenario. We derive the highest traffic density, TD_h, in I_n^M by evaluating traffic density over an arbitrary link λ_B^A (which connects nodes A and B), denoted by $\rho_{\lambda_B^A}$, and determining the maximum possible $\rho_{\lambda_B^A}$. The subsequent derivation assumes that messages are routed by Algorithm R, following packet switching for communication. A packet consists of one message and can be transmitted to a neighboring node during one *cycle*, if no congestion arises. Note that any other deadlock-free routing algorithm would result in the same traffic density bound.

Under the uniform message distribution, a node X in I_n^M issues, on an average, one message to each node other than X over a period of $M - 1$ cycles, provided that a node generates one message per cycle. In order to evaluate TD_h, every node is assumed to issue one message per cycle, under which we consider how many messages go through a given link over a period of $M - 1$ cycles. Let $\Xi(Adr, \lambda_B^A)$ denote the set of all messages with tag (when issued) being Adr which travel through link λ_B^A over a period of $M - 1$ cycles. It is clear that only messages in $\Xi(Adr, \lambda_B^A)$ can possibly contribute to the traffic density of link λ_B^A. Let $|\Xi(Adr, \lambda_B^A)|$ be the number of elements (i.e., messages) in the set. We are interested in the *average* of $|\Xi(Adr, \lambda_B^A)|$, and without confusion, $|\Xi(Adr, \lambda_B^A)|$ refers to average $|\Xi(Adr, \lambda_B^A)|$. Traffic density over λ_B^A is then expressed by

$$\rho_{\lambda_B^A} = \sum_{Adr} |\Xi(Adr, \lambda_B^A)| / (M-1), \quad (1)$$

since it gives the mean number of messages traversing the link per cycle.

We want to determine the maximum value of $\rho_{\lambda_B^A}$. As noted earlier, the incomplete hypercube is an asymmetric structure, so $|\Xi(Adr, \lambda_B^A)|$ is not constant for a link λ_B^A. There are two kinds of links in I_n^M: intercube links and intracube links. All links are bidirectional and full duplex, i.e., two messages can be transmitted simultaneously in the opposite directions of any link. The intercube links are called type a links. We further classify intracube links in a constituent cube H_i into three types and examine $|\Xi(Adr, \lambda_B^A)|$ for different types of intracube link λ_B^A separately.

b) $\lambda_{\in\Psi_i}^{\in\Psi_i}$, each of which connects two pivot nodes;

c) $\lambda_{NP}^{\in P_{i,j}}$, each of which connects a node in $P_{i,j}$ to a nonpivot node, NP; and

d) $\lambda_{NP'}^{NP}$, each of which connects two nonpivot nodes NP and NP'.

In order to facilitate our subsequent derivation, we classify the messages which can possibly traverse intracube link $\lambda_B^A \in H_i$ into the following classes, for any constituent cube H_i.

0) $\mu^{C\to D}$, for $C, D \in H_i$;
1) $\mu^{Y\to Y'}$, for $Y \in H_i$ and $Y' \in \Phi(< i)$;
2) $\mu^{Z\to Z'}$, for $Z \in H_i$ and $Z' \in \Phi(> i)$; and
3) $\mu^{X\to X'}$, for $X \in \Phi(< i)$ and $X' \in \Phi(\geq i)$,

where $\Phi(< i)$ is defined earlier, and $\Phi(> i)$ is the collection of all constituent cubes H_h, $h > i$, together with the intercube links among them inside I_n^M. Notice that messages of classes $\mu^{Z'\to Z}$, and $\mu^{U\to U'}$, (where $U \in \Phi(> i)$, $U' \in \Phi(< i)$) are not considered because according to Lemma 5, none of them could possibly traverse any intracube link.

In an incomplete hypercube, the normal order of traversing links by a given message may be violated due to the absence of a required link, resulting in the traversal of a higher order intercube link first. Specifically, messages of classes 0, 1, and 2 would never violate the normal order of traversing intracube links inside H_i (because intracube links with numbers less than i are all present), but messages of class 3 could. In the following, we provide an important result which precludes the possibility of having, on an average, more than one message of class 3 which traverses an intracube link $\lambda|_l$ over a period of $M-1$ cycles, for a given tag. Let $\Xi_3(Adr, \lambda|_l \in H_i)$ be the set of all the possible class 3 messages which have a given initial tag value Adr, and which traverse an intracube link $\lambda|_l$ in H_i along a given direction.

LEMMA 9. *Over a period of $M-1$ cycles, the mean number of messages in $\Xi_3(Adr, \lambda|_l \in H_i)$ is no more than one.*

A proof of this lemma is given in the appendix. Using the lemma, we now estimate upper bounds on the value $|\Xi(Adr,\ \lambda_B^A)|$ for the three types of intracube links. From Lemma 8, it is known that no message of class $\mu^{X\to X'}$, $X \in \Phi(< i)$, and $X' \in \Phi(\geq i)$, traverses a link of type $\lambda_{\in\Psi_i}^{\in\Psi_i}$ in H_i. This leads to the following theorem, which characterizes the traffic bound on any link $\lambda_{\in\Psi_i}^{\in\Psi_i}$ over $(M-1)$ cycles.

THEOREM 1. *The maximum value of $|\Xi(Adr,\ \lambda_{\in\Psi_i}^{\in\Psi_i})|$ for any link $\lambda_{\in\Psi_i}^{\in\Psi_i}$ in a constituent cube H_i over any period of $M-1$ cycles is 2.*

PROOF. The only messages that can traverse $\lambda_{\in\Psi_i}^{\in\Psi_i}$ are of classes 0, 1, and 2. None of these messages involve any violation of the normal order of traversing links within H_i, and the messages inside H_i behave in the same way as in a complete hypercube. Therefore, for each tag, the average number of messages traversing a given link over a period of $M-1$ cycles is no more than 2, one traveling in each direction. □

LEMMA 10. *For any given tag, the average number of class 3 messages traversing a link of type $\lambda_{NP}^{\in P_{i,j}}$ over a period of $M-1$ cycles is no more than 1.*

LEMMA 11. *For any given tag, the average number of class 3 messages traversing a link of type $\lambda_{NP'}^{NP}$ over a period of $M-1$ cycles is no more than 1.*

Proofs of the above two lemmas are provided in the appendix. These two lemmas imply that for a given tag, there can be at most one more message of class 3 traversing any given link, in addition to the two messages (of classes 0, 1, or 2) which are originated inside H_i with tag Adr and travel over the link in two opposite directions. The next two theorems thus follow.

THEOREM 2. *The maximum value of $|\Xi(Adr,\ \lambda_{NP}^{\in P_{i,j}})|$ for any link $\lambda_{NP}^{\in P_{i,j}}$ in a constituent cube H_i is 3.*

THEOREM 3. *The maximum value of $|\Xi(Adr,\ \lambda_{NP'}^{NP})|$ for any link $\lambda_{NP'}^{NP}$ in a constituent cube H_i is 3.*

We have proved from the above results that, for a given tag Adr in I_n^M, the mean number of messages which traverse a particular intracube link over a period of $M-1$ cycles is bounded by 3. For intercube links, a similar result is derived below.

From the abstract structure of I_n^M shown in Fig. 2, we observe that the messages which traverse an intercube link λ_B^A with $A \in H_j$ and $B \in \Phi\ (\geq q)$ for $q > j$, can be grouped into three classes:

1) $\mu^{X\to X'}$, $X \in \Phi(\geq q)$ and $X' \in \Phi(\leq j)$;
2) $\mu^{Y\to Y'}$, $Y \in \Phi(\leq j)$ and $Y' \in \Phi(\geq q)$;.
3) $\mu^{Z\to Z'}$, $Z \in \Phi(< q, > j)$ and $Z' \in \Phi(\geq q)$,

where $\Phi(< q, > j)$ is the collection of all constituent cubes H_h, $q > h > j$, together with the intercube links among them inside I_n^M. Notice that messages of class $\mu^{U\to U'}$, $U \in \Phi(< q, > j)$ and $U' \in \Phi(\leq j)$, are not considered because they never traverse link λ_B^A.

LEMMA 12. *Given a tag, the mean number of class 2 messages traversing an intercube link λ_B^A with $A \in H_j$ and $B \in \Phi\ (\geq q)$, for $q > j$, over a period of $M-1$ cycles is no more than 2.*

PROOF. See the appendix. □

As a result of the above lemma, we have a bound on $|\Xi(Adr,\ \lambda_B^A)|$ below, where λ_B^A is an intercube link. A proof of Theorem 4 can be found in the appendix.

THEOREM 4. *The maximum value of $|\Xi(Adr,\ \lambda_B^A)|$ for an intercube link λ_B^A is less than or equal to 3.*

Finally, we arrive at the upper bound on the traffic density of any link in I_n^M, whose proof is given in the appendix. The existence of a constant traffic density upper bound

suggests that cube links may be designed easily to avoid any traffic bottleneck, making I_n^M superior to other nonhomogeneous topologies, such as trees and stars.

THEOREM 5. *Traffic density on any link in an incomplete hypercube I_n^M is bounded by 2, for all $n \geq 2$.*

5 Experimental Performance

5.1 Simulation Model

Simulation studies have been performed to evaluate and compare the communication behaviors of complete and incomplete hypercubes. Every node in a simulated system consists of a processing element (PE) and a hardware router. For each node, one of those router links is connected to the PE, and the remaining links are to its immediate neighboring nodes. Messages generated at the PE are sent over the specific link to the router, from which they are delivered through appropriate links determined by routing Algorithm R (provided in Section 2) to neighboring nodes. Likewise, messages destined for the node, once they arrive at the router, are forwarded over this link to the PE. All links are bidirectional and full duplex.

Simulated systems operate in the packet-switched mode or under wormhole routing. In the packet-switched mode, a separate buffer is associated with each link. The message at the head of a buffer is transmitted in one cycle to the node connected at the other end of the link and, then, directed to an appropriate buffer chosen following the routing procedure. If multiple messages compete for a link in one cycle, they are all stored in the associated buffer, with the message at the buffer head proceeding over the link in that cycle and the rest being forwarded in sequence during subsequent cycles. Under wormhole routing, a message is broken into one or more fixed units (called "flits") for transmission, and a flit transfer between two connected nodes is assumed to take one cycle. A physical channel in the system is composed of one or several virtual channels, which share the bandwidth of the physical channel. Each virtual channel is allocated on a message-by-message basis, independent of other virtual channel allocation. The virtual channel assigned to a message is released only after all flits comprising the message are transmitted. Each virtual channel has a buffer to hold one flit.

In our simulation, messages are generated independently by all the nodes, with their destinations governed by message distributions. A node generates a message with probability r in a cycle, called the message generation rate. The system operates in a synchronous manner, and messages are issued by nodes at the beginning of each cycle. The message service discipline is first-come-first-served, and in one cycle, the head message in a buffer (or the flit receiving a physical channel under wormhole routing), advances to the connected neighboring node, whenever possible. A node may send or forward messages simultaneously over all incident links. A PE can receive one message directed to it from the router in a cycle; if multiple messages bound for the same PE arrive in one cycle, a random one is selected to advance to the PE and the rest stay in their original buffers (because there is only one link between a local PE and a router), referred to as the *single-accepting strategy* in [12].

The measures of interest in this experimental study are mean latency and throughput. In packet-switching, mean latency (L) is the number of cycles spent by a typical message from its source to its destination, taking the queueing delay into consideration. Under wormhole routing, L is the time from message creation until the first flit of the message is accepted at the destination. In either case, the source queueing time is included in L. Throughput (T) is the probability of a node receiving a message (or flit) during a cycle; it indicates accepted traffic, or equivalently, the load. When a link experiences heavier traffic under packet-switching, its associated buffers at both ends involve more messages on the average, so that a message traveling through that link tends to encounter higher latency. Similarly, under wormhole routing, a link with heavy traffic tends to result in high blockage, making a message traversing the link incur large latency. For a given system, mean latency generally grows with the increase of T. When T is low, latency is contributed mainly by the number of hops a typical message makes, because little queuing delay (due to contention or blockage) is involved. As T increases, a longer queueing delay results, leading to higher latency.

It is assumed that, in the packet-switching mode, the buffer associated with a link has the capacity of holding three messages; whereas under wormhole routing, a physical channel is composed of three virtual channels. The simulation study under uniform traffic was carried out first to investigate the situations where the nature of tasks to be executed is unknown, and no assumption is made about the type of computation producing the messages. This study also enables us to verify our earlier analysis. Since many computations involve certain type of communication locality practically, simulation was then conducted to pursue system performance under nonuniform traffic, and the results are provided in a separate subsection. The simulation results are averaged over eight independent runs, with an approximate 95% confidence interval for each point shown in the figures equal to the point value ±3%.

5.2 Numerical Results under Uniform Traffic

Fig. 4 depicts mean latency (L) vs. throughput (T) for three incomplete hypercubes of sizes 1048, 1114, and 1818, respectively, under uniform traffic with packet-switching. The three system sizes are chosen for illustration, because 1048 is slightly more than 2^{10} and 1818 is close to 2^{11}, with 1114 in between. (Many other sizes were simulated and their results followed similar trends.) When the system size is 1024, L starts with 5 and gradually increases as T grows due to increasing contention. For the system with 1048 nodes (composed of H_{10}, H_4, and H_3), L grows slowly until T reaches 0.6, and starts to change quickly thereafter. Compared with the complete hypercube of H_{10}, I_{11}^{1048} has virtually the same L value for any $T \leq 0.68$, implying that no serious contention exists in the system with low to moderately high traffic. Similarly, for the system with 1114 nodes (composed of H_{10}, H_6, H_4, H_3, and H_1), L stays pretty close to that of H_{10} until T approaches 0.68, and begins to increase

rapidly thereafter. The system can deliver good performance unless the load is high (say, > 0.7).

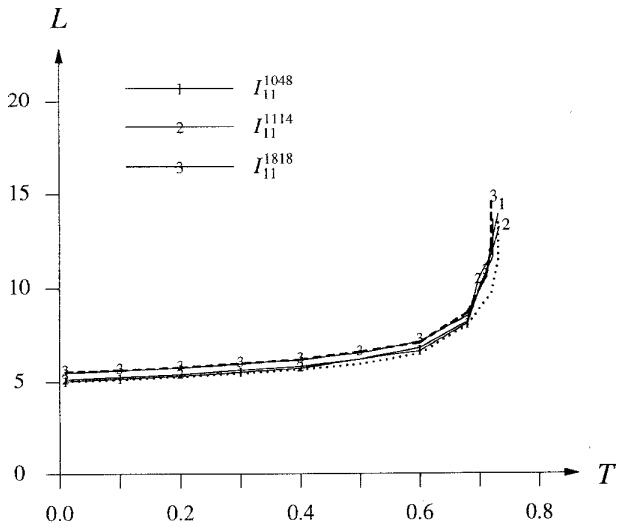

Fig. 4. Mean latency (L) vs. Throughput (T) under uniform traffic with packet-switching. (H_{10} and H_{11} results are shown, respectively, by the dotted and the dashed curves.)

For the system with 1818 nodes (comprising H_{10}, H_9, H_8, H_4, H_3, and H_1), its L is always slightly larger than L of H_{10} and approaches L of H_{11}, because its size is close to that of H_{11}. This incomplete hypercube is expected to deliver performance as good as H_{11}, the compatible complete hypercube.

It is known that L comprises two components: 1) the mean number of hops between the source and the destination of a typical message, and 2) the queuing delay due to contention. Under a given traffic pattern, component 1) is determined only by the system size and is independent of T (the system load). Component 1) values for H_{10}, I_{11}^{1048}, I_{11}^{1114}, I_{11}^{1818}, and H_{11} are 5.00, 5.05, 5.14, 5.47, and 5.50, respectively. From the curves of Fig. 4, component 2) for each system under different T can be obtained immediately.

Simulated peak traffic density data was also gathered; values were 1.00, 1.62, 1.64, 1.06, and 1.00 for H_{10}, I_{11}^{1048}, I_{11}^{1114} I_{11}^{1818}, and H_{11}, respectively, all bounded by 2, as predicted. Since the degree of contention in a system can be indexed by traffic density, it is interesting to find out the percentage of links with high traffic density. To this end, we collected the traffic density distribution of links in every simulated system. It turns out that for I_{11}^{1048}, less than 0.6% of the links carry traffic density greater than 1.0 under any traffic load (T). Similarly, for I_{11}^{1114} (or I_{11}^{1818}) under any T, less than 2.1% (0.2%) of its links have traffic density beyond 1.0. In general, only a small fraction of links in an incomplete system carry very high traffic density.

The simulation results of the same set of systems under wormhole routing are demonstrated in Fig. 5, where a message is assumed to contains 20 flits. Again, incomplete hypercubes are shown to perform quite closely to their complete counterparts for any T. The queuing delay due to blockage for each system can be obtained directly from a curve in the figure, because the mean number of hops between source and destination is fixed, irrespective of T. At a low T, latency is contributed chiefly by the mean number of hops, and when T grows, latency increases as a result of larger blockage. Compared with the curves in Fig. 4, it is observed that blockage under wormhole routing causes L to grow much faster than contention under packet-switching, as T increases. This is because a blocked message under wormhole routing could span up to 20 nodes (one flit at a node), holding many resources.

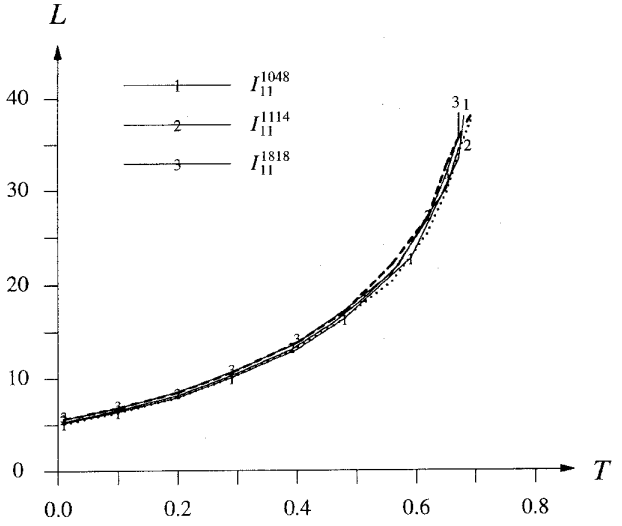

Fig. 5. Mean latency (L) vs. Throughput (T) under uniform traffic with wormhole routing. (H_{10} and H_{11} results are shown, respectively, by the dotted and the dashed curves.)

Simulated peak traffic density for every system under wormhole routing is also bounded by 2. Specifically, peak traffic density values are 1.00, 1.67, 1.58, 1.15, and 1.00 for H_{10}, I_{11}^{1048}, I_{11}^{1114}, I_{11}^{1818}, and H_{11}, respectively. The percentage of extremely heavily loaded links in an incomplete system is very small. In I_{11}^{1048}, for example, less than 0.5% of the links carry traffic density exceeding 1.0. Similarly, I_{11}^{1114} (or I_{11}^{1818}) has less than 1.4% (or 0.4%) of its links with traffic density > 1.0.

5.3 Numerical Results under Nonuniform Traffic

There are many forms of nonuniform message distributions, reflecting different types of reference locality. In this study, we focused on two types of nonuniform message distributions, referred to, respectively, as sphere of locality and decreasing probability reference [11]. An abstraction of sphere of locality is as follows: each node is considered to be at the center of a sphere comprising all nodes which are no more than B hops away, and the center node sends messages to other nodes inside its sphere equiprobably with total probability (which is usually high), with probability $1 - \zeta$ being addressed uniformly to all the nodes (other than the originating node). Decreasing probability reference intuitively captures the notion that the probability of sending

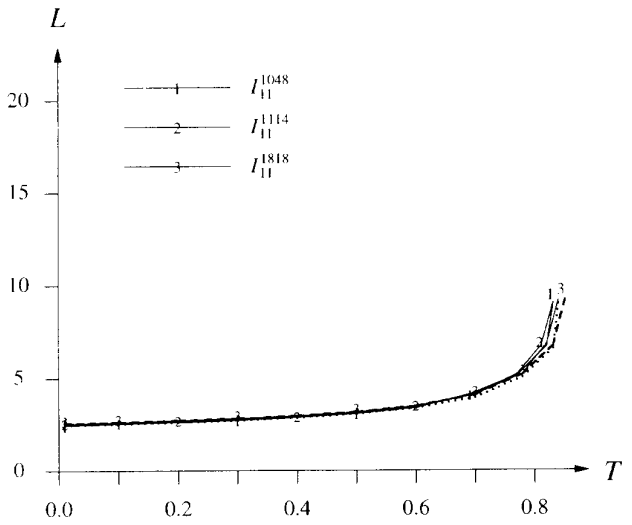

Fig. 6. Mean latency (L) vs. Throughput (T) under sphere of locality traffic with packet-switching. (H_{10} and H_{11} results are shown, respectively, by the dotted and the dashed curves.)

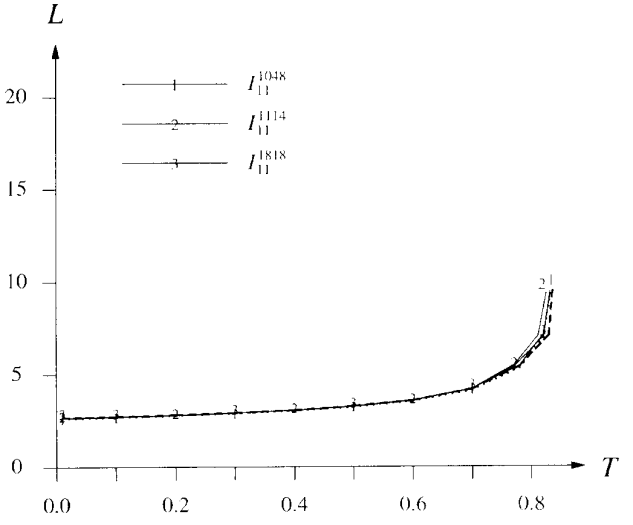

Fig. 8. Mean latency (L) vs. throughput (T) under decreasing probability reference with packet-switching. (H_{10} and H_{11} results are shown, respectively, by the dotted and the dashed curves.)

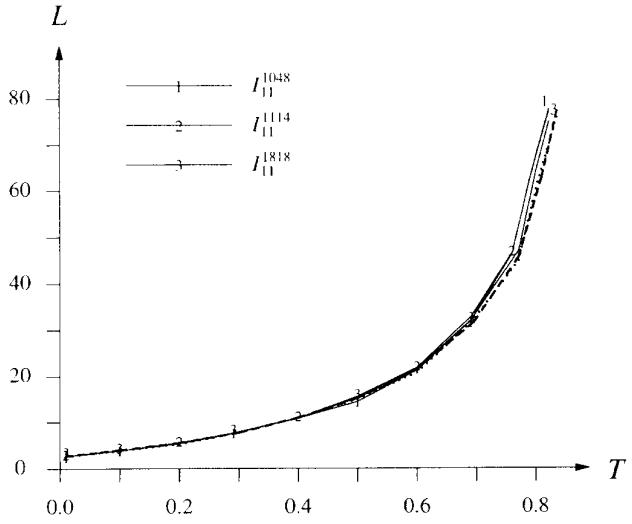

Fig. 7. Mean latency (L) vs. throughput (I) under sphere of locality traffic with wormhole routing. (H_{10} and H_{11} results are shown, respectively, by the dotted and the dashed curves.)

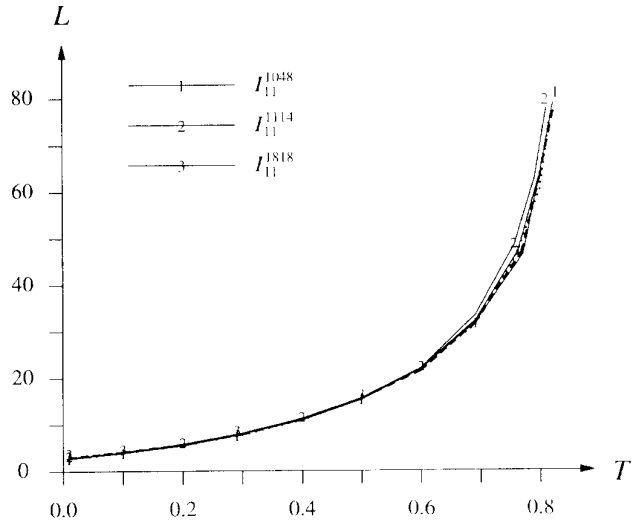

Fig. 9. Mean latency (L) vs. throughput (T) under decreasing probability reference with wormhole routing. (H_{10} and H_{11} results are shown, respectively, by the dotted and the dashed curves.)

messages to a node decreases as its distance from the source node increases. This is often expected, as mapping a distributed computation onto a hypercube system desires more frequent message exchanges with physically closer nodes. The results under these two nonuniform message distributions are depicted in Figs. 6, 7, 8, and 9.

Mean latency vs. throughput under sphere of locality traffic with $B = 4$ and $\zeta = 0.8$ for the three earlier incomplete hypercube systems with packet-switching is illustrated in Fig. 6. Under this reference locality, a generated message is more than 10 times as likely to be destined for a node within a sphere as for a node outside the sphere. The mean latency of an incomplete hypercube increases gradually as T grows up to 0.7, and beyond that, L starts to change more rapidly. All the three incomplete hypercubes, H_{10}, and H_{11} all have virtually the same mean latency when T is less than 0.78, indicating that an incomplete system can handle traffic as satisfactorily as a compatible complete system, unless the traffic load (T) is exceedingly high. The mean number of hops between source and destination of a typical message under this traffic pattern is 2.46, 2.46, 2.48, 2.55, and 2.55 for H_{10}, I_{11}^{1048}, I_{11}^{1114}, I_{11}^{1818}, and H_{11}, respectively (independent of T). When compared with the results in Fig. 4, it can be observed that for any given T, an incomplete hypercube consistently exhibits lower mean latency under reference locality than under uniform traffic. These results show that the mean latency derived under uniform traffic provides an upper bound on L of a real execution situation, where a certain type of reference locality usually exists. It is also found from our simulation that the peak traffic density over

a link in any system now is no more than 0.9, much lower than that observed under uniform message distributions.

The simulation results of same systems with the sphere of locality traffic pattern but under wormhole routing is depicted in Fig. 7, where a message is assumed to involve 20 flits. An incomplete system exhibits performance almost identical to its complete counterpart, unless T is extremely high. Like a complete system, an incomplete system sees its L grow much more swiftly under wormhole routing than under packet-switching, as T increases. The peak traffic density in any system under this routing strategy is found to be lower than 1.03.

Simulation results for the three incomplete hypercube systems with decreasing probability reference under packet-switching are shown in Fig. 8. They are obtained under the reference distribution that the probabilities of message destinations to be 1 hop, 2 hops, 3 hops, and 4 hops away when generated, are 0.34, 0.20, 0.16, and 0.10, respectively, with the remaining probability of 0.20 having messages uniformly destined for all nodes (other than the originating node). Under this reference distribution, the rate for a message to terminate at an immediate neighbor is more than seven times higher than that at a node 2 hops away, which in turn is about 3.3 times higher than that at a node 3 hops away, which is approximately 2.8 times higher than that at a node 4 hops away. The probability of addressing a node decreases as the distance grows. This message distribution exhibits more severe reference locality than the prior nonuniform message distribution. From Fig. 8, L is found to grow slowly as T increases until 0.78, and it then begins to changes rapidly thereafter. An incomplete hypercube is able to handle messages fluently, unless its traffic load is fairly high. According to our simulation, the peak traffic density over a link in any system under this decreasing probability reference is less than 0.93.

The results under this decreasing probability reference with wormhole routing are given in Fig. 9, where a message contains 20 flits. Again, an incomplete system has almost identical performance as a corresponding complete system, unless the system load (T) is very high. It is clear from the curves in Figs. 8 and 9 that the latency (L) of any system grows much faster under wormhole routing than under packet-switching, as T increases. The peak traffic density over a link in any system under this reference pattern is below 1.02.

6 Conclusions

We have analyzed traffic behaviors over links in any sized incomplete hypercube under the uniform message distribution, following packet switching for communication. Our analytic result reveals that link traffic density is upper bounded by 2, independent of the system size, when deadlock-free routing, like the e-cube routing algorithm or its variation, is employed to create paths for message delivery. The simulation study indicates that mean latency for transmitting messages is roughly the same in an incomplete hypercube as in a compatible complete hypercube under both packet-switching and wormhole routing. This interesting result implies that an incomplete hypercube, despite its structural nonhomogeneity, can easily be so constructed that any potential congestion points are avoided, ensuring good performance always.

Points of congestion are likely to occur in other nonhomogeneous topologies such as trees and stars. For example, the peak traffic density of a binary tree with l levels under uniform traffic is 2^{l-1}, as derived in [1]. This high traffic density tends to result in poor performance. The incomplete hypercube is thus favorable when compared with other nonhomogeneous topologies, making it a practical and interesting structure.

There are some open issues related to incomplete hypercubes. Since multicast is a useful operation and has been studied earlier for complete hypercubes [14], it is interesting to examine how to carry out multicast efficiently in incomplete hypercubes. Hot spots in a network system often cause serious performance degradation [15]. The behavior of an incomplete hypercube with hot spots is worth investigation.

Appendix

Proof of Lemma 3. From the definition of Ω_i, it is clear that H_m is the constituent cube immediately below H_i and T_i^0 is an $(m+1)$-dimensional subcube, which is larger in size than the summation of all constituent cubes with dimensions less than $m+1$. Since all pivot nodes in H_i are directly connected to nodes in constituent cubes H_j, for all $j \le m$, through links with number i (from Observation 1), every such connected pair thus differ only in bit i of their addresses (see Fig. 3). Nodes in T_i^0 are addressed by

$$(a_{n-1}, a_{n-2}, \cdots, a_{i+1}, 0^{i-m-1}, *^{m+1}),$$

whereas nodes in constituent cubes H_j have addresses of

$$(a_{n-1}, a_{n-2}, \cdots, a_{i+1}, 1, 0^{i-m-2}, x_m, x_{m-1}, \cdots, x_j, \cdots, x_0)$$

where $x_j \in \{0, 1\}$. As a result, each node in H_j, for all $j \le m$, has its address differing in bit i of the address of a node in T_i^0, indicating that each node in H_j is connected to a node (which is a pivot node) in T_i^0 through a link with number i. This proves the lemma. □

Proof of Lemma 9. Consider two messages $\mu^{X \to X'}$ and $\mu^{Y \to Y'}$. We prove this lemma by showing that after the two messages traverse the given link $\lambda|_l$ in H_i, they have the same tag value (note that the two messages have an identical tag Adr initially). If this is shown, then the two messages must terminate at the same destination, because they are routed by the same routing algorithm, from the common node (after traversing the given link), with an identical tag value. Therefore, nodes X' and Y' are, in fact, the same node. Since there can be only one node whose relative address with respect to X' (i.e., Y') is Adr, it is clear that nodes X and Y coincide, indicating that the two messages are actually the same one.

The subsequent three observations follow immediately: 1) the two messages correct all nonzero tag bits lower than j before departing from H_j (as all links with numbers less than j exist in H_j); 2) at the time when the two messages traverse the common link $\lambda|_l$, none of the links with numbers larger than l, $l \neq i$, have been traversed by any of the messages (because inside H_i, the normal link traversal order is preserved for all links with numbers less than i, and no link with number greater than $i-1$ will be traversed); and 3) both messages take links with number i to reach H_i and thus their tag bits i are corrected upon entering H_i. From the above observations, we know that only tag bits with positions between l and j must be examined to check if the tag values of the two messages are the same immediately before they traverse the common link $\lambda|_l$.

We consider two cases, depending on whether or not there is a nonzero bit p, $l > p > j$, in Adr. If there is no such bit, the two messages certainly have the same tag value upon traversing the common link. On the other hand, if there is a nonzero bit p, $l > p > j$, in Adr, three possible situations arise: 1) constituent cube H_p exists, 2) cube H_p is absent, but for a nonzero bit s, $i > s > p$, in Adr, constituent cube H_s exists, and 3) cube H_p is absent, and so is H_s for each nonzero bit s, $i > s > p$, in Adr. For situation 1), both messages will visit H_p after leaving H_j (from Lemma 6), and thus the tag bit p in both messages gets corrected before entering H_i. Similarly, for situation 2), both messages will visit H_s before reaching H_i, and the traversal of link p is made inside H_s. For situation 3), the two messages will enter H_i with their tag bit p uncorrected. However, the traversal of link p will be carried out in H_i before they traverse the common link, because p is less than l and the normal link traversal order is preserved for all links with numbers less than i inside H_i. This completes our proof. □

PROOF OF LEMMA 10. Consider the abstract structure of H_i given in Fig. 3. Let $\lambda|_l$ be a link of type $\lambda_{NP}^{\in P_{i,j}}$. Suppose there is a class 3 message with initial tag Adr, say $\mu^{X\to X'}$, which traverses the given link $\lambda|_l$ in the $P_{i,j} \to NP$ direction. We know from Lemma 9 that only one such class 3 message exists. What remains to be shown is that there exists no other message of class 3 having initial tag Adr, say $\mu^{Y\to Y'}$ with $Y \in \Phi(< i)$ and $Y' \in \Phi(\geq i)$, which traverses $\lambda|_l$ in the opposite direction, namely, the direction of $NP \to P_{i,j}$. Assume that there is such a message $\mu^{Y\to Y'}$, then, three possible cases are considered, depending on the location where $\mu^{Y\to Y'}$ enters H_i: 1) $\mu^{Y\to Y'}$ enters H_i through $P_{i,k}$ with $k < j$, 2) $\mu^{Y\to Y'}$ enters H_i through $P_{i,q}$ with $q > j$, and 3) $\mu^{Y\to Y'}$ enters H_i through $P_{i,j}$ itself.

The following facts are useful: The number of the given link, l, is greater than $j-1$, as links with numbers 0, 1, ..., $j-1$ are all inside $P_{i,j}$ and never connect pivot nodes in $P_{i,j}$ to nonpivot nodes; $P_{i,j}$ and pivot sets $P_{i,k}$, for all $k < j$, are enclosed in a subcube of dimension $j+1$, denoted by Π_i^{j+1} (see Fig. 3).

Case 1. Since $l \geq j$, we consider two subcases: $l = j$ and $l \geq j+1$. For $l = j$, any message $\mu^{Y\to Y'}$ entering H_i through $P_{i,k}$ will never traverse links with number l in H_i, since all nodes in H_k are connected to H_j, $j > k$, using links with number j (from Observation 1), and message $\mu^{Y\to Y'}$ visits H_j (for bit j (= l) in Adr is nonzero) before reaching H_i. For $l \geq j+1$, message $\mu^{Y\to Y'}$ has to leave Π_i^{j+1} in order to traverse link $\lambda|_l$ along the direction of $NP \to P_{i,j}$ (because link $\lambda|_l$, $l \geq j+1$, connects a node in Π_i^{j+1} to another node outside Π_i^{j+1}). Once it does so, however, it cannot reenter Π_i^{j+1} again (from Lemma 4). Hence, this subcase is impossible, because the message will reenter Π_i^{j+1} after the traversal of the given link along the direction of $NP \to P_{i,j}$.

Case 2. In this case, l is greater than q, because nodes in $P_{i,j}$ are connected to corresponding nodes in $P_{i,q}$ by links with number q, $q > j$, and all links with numbers less than q exist in $P_{i,q}$. By a similar argument as above, message $\mu^{Y\to Y'}$ must leave Π_i^{q+1} in order to traverse the given link along the direction of $NP \to P_{i,j}$. Again, the message is prohibited from traversing the given link along direction $NP \to P_{i,j}$, according to Lemma 4.

Case 3. This case is ruled out similarly, because it requires message $\mu^{Y\to Y'}$ to leave Π_i^{j+1} before traversing the given link along the direction of $NP \to P_{i,j}$. □

PROOF OF LEMMA 11. Let $\lambda|_l$ be a link of type $\lambda_{NP}^{NP'}$. We have two possible cases, depending on whether or not the two nonpivot nodes are located in the same partition.

NP in T_i^a and NP' in $T_i^b (\neq T_i^a)$

Suppose $\mu^{X\to X'}$ is a class 3 message which traverses $\lambda|_l$ along the direction of $T_i^a \to T_i^b$. From Lemma 9, we know that there exists only one such message. Since links inside H_i are traversed in the normal order without violation, if message $\mu^{X\to X'}$, after entering H_i through T_i^0, reaches T_i^a before T_i^b, every other message of class 3 must also visit T_i^a first, after entering H_i through T_i^0. Therefore, it is impossible for any

message to traverse $\lambda|_l$ along the direction of $T_i^b \rightarrow T_i^a$.

Both NP and NP' in T_i^a

Since every partition is of the same size, let us assume that a partition involves an $(m+1)$-dimensional subcube in H_i. For $a = 0$, NP and NP' are connected by a link with number less than m, because each link $\lambda|_m$ in H_i connects either two pivot nodes or one pivot node to one nonpivot node, and a message of class 3 terminates directly at NP or NP' without taking the link between the two nodes, as such a message enters H_i (from a lower constituent cube) via a pivot node and traverses links in H_i following the normal order. For $a > 0$, a message of class 3 does not traverse the link connecting NP and NP' either, since the message, after entering T_i^0, corrects all nonzero tag bits p, $p < m+1$, before proceeding to T_i^a, $a > 0$, and the two nodes in T_i^a, $a > 0$, are connected by a link with number less than $m+1$. □

PROOF OF LEMMA 12. For a given tag, say Adr, we consider the class 2 messages $\mu^{Y\rightarrow Y'}$, $Y \in \Phi(\leq j)$, and $Y' \in \Phi(\geq q)$, for $q > j$, which originate from H_j and from $\Phi(< j)$ separately. A message originating from H_j must correct all nonzero Adr bits whose positions are lower than j before taking an intercube link to leave H_j, and once it leaves H_j, it never reenters H_j. Since H_j links with numbers less than j are traversed, if needed, in the normal order, for any Adr, there is only one message $\mu^{Y\rightarrow Y'}$ with $Y \in H_j$ passing through the given intercube link λ_B^A, $A \in H_j$ and $B \in \Phi(\geq q)$.

We next show that there can be at most one more message originating from $\Phi(< j)$ which traverses the given intercube link. Suppose that there exist two such messages, $\mu^{Y1\rightarrow Y1'}$ and $\mu^{Y2\rightarrow Y2'}$, where $Y1$, $Y2 \in \Phi(< j)$. When the two messages leave H_j after taking λ_B^A, their tag bits with positions lower than $j+1$ have become zero, and no link with number greater than j has ever been traversed by either message (from Lemma 7). Since the two messages have the same initial tag value, Adr, their tag values after arriving at node B must be identical, indicating that $Y1'$ and $Y2'$ coincide (because the same routing algorithm is employed). For any given Adr, if $Y1'$ is the same as $Y2'$, then $Y1$ must be the same as $Y2$, implying that there cannot be more than one message issued at $\Phi(< j)$. As a result, there are at most two messages of class 2 which traverse a given intercube link, for any given tag. □

PROOF OF THEOREM 4. Messages traversing an intercube link λ_B^A, $A \in H_j$ and $B \in \Phi(\geq q)$, for $q > j$, consist of classes 1, 2, and 3. Class 1 messages are issued from $\Phi(\geq q)$ and terminate at $\Phi(\leq j)$. From Lemma 5, a message of class 1 never traverses any link in $\Phi(\leq j)$, namely, the message arrives at its destination (i.e., node A) immediately after taking the intercube link λ_B^A. For a given tag Adr and destination (i.e., A), there is a single class 1 message.

We next show that a message of class 2, $\mu^{Y\rightarrow Y'}$ with $Y \in \Phi(\leq j)$ and $Y' \in \Phi(\geq q)$, and a message of class 3, $\mu^{Z\rightarrow Z'}$ with $Z \in \Phi(< q, > j)$ and $Z' \in \Phi(\geq q)$, cannot coexist for a given Adr. A message of class 3 has to traverse an intercube link, say $\lambda|_f$, to reach node A in H_j before taking the intercube link λ_B^A. If the link number of λ_B^A is l, then l should be greater than f, for otherwise, $\lambda|_l$ would have been taken before $\lambda|_f$. Assume that message $\mu^{Y\rightarrow Y'}$ exists. In this case, bit f in Adr must be zero, because every node in $\Phi(\leq j)$ has intercube link $\lambda|_f$ connected (since $f > j$, from Observation 1), and the message would otherwise have traversed link $\lambda|_f$ prior to $\lambda|_l$ (for $l > f$). This implies that message $\mu^{Z\rightarrow Z'}$ cannot exist. On the other hand, if $\mu^{Z\rightarrow Z'}$ exists, Adr must have a nonzero bit f and therefore message $\mu^{Y\rightarrow Y'}$ cannot exist.

Now, if messages of class 2 exist, then from Lemma 12, there can be at most two such messages. If messages of class 3 exist, we explain below that there can be only one such message. In this case, bit f in Adr is nonzero and the traversal of $\lambda|_f$ is made to depart from a node in H_f, say node C, for H_f. It can be shown that constituent cube H_f must be present and the originating node Z has to be in $\Phi(\leq f)$, because otherwise if Z is in $\Phi(> f)$, the traversal of $\lambda|_f$ is confined inside $\Phi(> f)$ (since a node in $\Phi(> f)$ is connected by $\lambda|_f$ to another node in $\Phi(> f)$). At node C, all tag bits lower than f and no tag bits higher than f have been corrected. As a result, any two messages of class 3 with the given tag should have an identical tag value at node C. Using an argument similar to that provided in the proof of Lemma 12, we know that, for any given tag, there can be only a single message of class 3. This completes our proof. □

PROOF OF THEOREM 5. The traffic density bounds on the intracube links and on the intercube links are treated separately.

Bound on Intracube Links

We analyze the changes on the traffic density bounds when a constituent cube is added to an existing incomplete hypercube, which starts with a configuration composed of only two cubes H_{n-1} and H_i, $(n-1) > i$ (note that this special class of incomplete hypercubes has been analyzed earlier [9]). Consider an arbitrary link $\lambda|_l$ inside constituent cube H_i of the in-

complete hypercube $I_n^{2^{n-1}+2^i}$ (the case of $\lambda|_l$ in constituent cube H_{n-1} is examined later). From Theorems 1, 2, and 3, it is clear that for a given tag *Adr*, the mean number of messages passing through $\lambda|_l$ over a period of $M-1$ cycles (here, $M = 2^{n-1} + 2^i$) is no more than three. There are $(M-1)$ *Adr*'s in total, with bit l equal to 1 (so that link $\lambda|_l$ is traversed), resulting in $\lceil (M-1)/2 \rceil$ different tag values contributed to (1). The worst case scenario of traffic density, from (1), is bounded by

$$\frac{3(2^{n-2} + 2^{i-1} - 1)}{2^{n-1} + 2^i - 1}.$$

Now, let an arbitrary cube $H_q, n-1 > q > i$, be added to the initial incomplete hypercube, giving rise to a system with size $M + 2^q$. The new messages traversing $\lambda|_l$, introduced due to the addition of H_q, are those which originate from H_i and terminate in H_q because H_q is a higher cube with respect to the cube in which $\lambda|_l$ resides, and messages issued from a higher cube will not traverse the given intracube link, according to Lemma 5. Over a period of $M-1$ cycles, the number of newly introduced tag values is thus 2^q or less. Since any new tag contributed to traffic density over link $\lambda|_l$ must have a nonzero bit l, and for each tag, again, no more than three messages traverse $\lambda|_l$, we have the bound of

$$\frac{3(2^{n-2} + 2^{i-1} - 1 + 2^{q-1})}{2^{n-1} + 2^i - 1 + 2^q}.$$

It can be shown by repeating this process that if multiple constituent cubes with K_u total nodes are added in between H_{n-1} and H_i, traffic density on link $\lambda|_l$ is bounded by

$$\frac{3(2^{n-2} + 2^{i-1} - 1 + K_u/2)}{2^{n-1} + 2^i - 1 + K_u}.$$

We next consider the impact of adding a constituent cube $H_j, j < i$, to the incomplete hypercube obtained so far. The new messages traversing $\lambda|_l$, introduced by this addition, are those between H_j and H_i (because those between H_j and $\Phi(> i)$ have been taken into account previously when the number of messages traversing $\lambda|_l$ for a given tag is assumed to be 3). Over the period of $M-1$ cycles, the average number of newly introduced tag values is no more than 2^j (with 2^{j-1} originating from H_j and another 2^{j-1} terminating in H_j, for $l \geq j$; if $j > l$, no message originating from H_j traverses $\lambda|_l$ in H_i). There are no more than three messages traversing the given link for any tag, yielding the traffic density upper bound of

$$\frac{3(2^{n-2} + 2^{i-1} - 1 + K_u/2 + 2^j)}{2^{n-1} + 2^i - 1 + K_u + 2^j}.$$

If this process is repeated and multiple constituent cubes of dimensions less than i, with K_b total nodes, are added, the bound can be expressed by

$$\frac{3(2^{n-2} + 2^{i-1} - 1 + K_u/2 + K_b)}{2^{n-1} + 2^i - 1 + K_u + K_b},$$

which is less than 2 provided that $2^{n-2} + 2^{i-1} + 1 + K_u/2 > K_b$. It is clear in this case that traffic density is below 2, for all $n \geq 2$.

Next, consider any link $\lambda|_l$ inside constituent cube H_{n-1}. According to Lemmas 10 and 11, there is no more than one class 3 message traversing the given link $\lambda|_l$. For any tag, the three messages which might traverse the given link over a period of $M-1$ cycles are $\mu^{X \to X'}$, $X \in \Phi(< n-1)$, and $X' \in H_{n-1}$, $\mu^{Y1 \to Y1'}$, $Y1 \in H_{n-1}$, and $Y1' \in \Phi(< n-1)$, and $\mu^{Y2 \to Y2'}$, $Y2 \in H_{n-1}$ and $Y2' \in \Phi(< n-1)$, where the latter two messages traverse the given link along opposite directions. We show in the following that there are only two possible messages (out of the three) which can pass through the given link, by treating the subcases of 1) X' being a nonpivot node and 2) X' being a pivot node ($\in H_{n-1}$) in sequence.

For 1), suppose that messages $\mu^{X \to X'}$ and $\mu^{Y2 \to Y2'}$ traverse the given link along the same direction. Then, it is clear that the two messages must take the same path until node X', after passing through the given link $\lambda|_l$ (because they have the same initial tag, and the normal link traversal order is maintained inside H_{n-1}). At node X', message $\mu^{Y2 \to Y2'}$ (which originates in H_{n-1}) has to take an intercube link $\lambda|_{n-1}$ to its destination $Y2'$ (in $\Phi(< n-1)$). However, X' has no $\lambda|_{n-1}$ since it is a nonpivot node. This contradiction implies that messag $\mu^{Y2 \to Y2'}$ is not present, and only two possible messages may traverse the given link.

For (2), suppose that node X is in H_i. If node X' belongs to $P_{n-1,i}$, then message $\mu^{X \to X'}$ never traverses any link in H_{n-1}, so it does not take the given link. If node X' belongs to $P_{n-1,q}, n-1 > q > i$, then constituent cube H_q exists and message $\mu^{X \to X'}$ issued in H_i will visit H_q (from Lemma 6), and then reaches its destination, X', immediately after taking link $\lambda|_{n-1}$ from H_q. Again, message $\mu^{X \to X'}$ never traverses the given link. Finally, if node X' belongs to $P_{n-1,j}$, $i > j$, then constituent cube H_j exists, and it can be easily shown that message $\mu^{X \to X'}$ does not take the given link, either. As a result, only two possible messages may traverse the given link.

For a message to traverse link $\lambda|_l$, its tag must have a nonzero l bit. According to (1), we arrive at the upper bound of traffic density over an arbitrary link in H_{n-1} being

$$\frac{2(2^{n-1}-1)}{2^{n-1}+K-1},$$

where K is the total number of nodes in $\Phi(< n-1)$.

Bound on Intercube Links

Now consider an arbitrary intercube link between constituent cubes H_i and H_j, $i > j$, in I_n^M. From Observation 1, $\lambda|_i$ is an intercube link between the two cubes. We have from the proof of Theorem 4 that for a given tag, the mean number of messages traversing this link over a period of $M-1$ cycles is 1) no more than 3, if the tag has a nonzero bit j (i.e., involving a traversal of link $\lambda|_j$); otherwise, it is 2) no more than 2. In either situation, the tag must have a nonzero bit i for a message to traverse the given link $\lambda|_i$. The total number of different possible tag values is M (which is less than 2^n). In situation 1), only tags whose bit positions i and j are "1" contribute to traffic density over the given link, yielding the bound of

$$\frac{3(2^{n-2})}{2^{n-1}+2^i+2^j+K'-1},$$

where K' equals $M-(2^{n-1}+2^i+2^j)$. In situation 2), only tags with bit position i being "1" contribute to traffic density over the given link, leading to the bound of

$$\frac{2(2^{n-1})}{2^{n-1}+2^i+2^j+K'-1}.$$

The worst case scenario of the upper bound on traffic density over an intercube link is therefore no larger than 2. □

List of Notations

I_n^M : n-dimensional incomplete hypercube with M nodes.

$P_{i,j}$: collection of pivot nodes associated with two constituent cubes H_i and H_j, $i > j$, of I_n^M.

$\Phi(< i)$: set of all constituent cubes H_j, $j < i$, together with all intercube links among these cubes in I_n^M.

$\Phi(< q, > j)$: collection of all constituent cubes H_h, $q > h > j$, together with the intercube links among them inside I_n^M.

Ψ_i : collection of all pivot nodes associated with H_j, for all $j < i$, plus all the connections among these nodes inside constituent cube H_i.

$O_w(y_{u-1}y_{u-2}\cdots y_1y_0, j)$: position of the wth nonzero bit after (i.e., right of) position j in bit string $y_{u-1}y_{u-2}\cdots y_1y_0$.

Ω_i : collection of $(m+1)$-dimensional disjoint subcubes in I_n^M, where the binary representation of M is

$$a_{n-1}a_{n-2}\cdots a_{i+1}a_ia_{i-1}\cdots a_1a_0$$

and

$$m = O_1(a_{n-1}a_{n-2}\cdots a_{i+1}a_ia_{i-1}\cdots a_1a_0, i)$$

in other words,

$$\Omega_i = \{(a_{n-1}a_{n-2}\cdots a_{i+1}0x_{i-1}x_{i-2}\cdots x_{m+2}x_{m+1}{}^{*^{m+1}}) \mid x_{i-1}x_{i-2}\cdots x_{m+2}x_{m+1} \text{ denotes any possible bit string}\}.$$

T_i^p : an element in Ω_i, namely

$$(a_{n-1}a_{n-2}\cdots a_{i+1}0x_{i-1}x_{i-2}\cdots x_{m+1}{}^{*^{m+1}}),$$

such that the value of $x_{i-1}x_{i-2}\cdots x_{m+1}$ equals p.

Π_i^{j+1} : partition in T_i^0 which involves $P_{i,j}$ and $P_{i,k}$, for all $k < j$.

$\mu|_{\leftarrow H_j}$: a message coming from a node in constituent cube H_j.

$\mu^{X\to X'}$: a message issued at node X and destined for node X'.

$\Xi(Adr, \lambda_B^A)$: set of all messages with tag (when issued) being Adr which travel through link λ_B^A over a period of $M-1$ cycles.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant No. CCR-9300075 and by the State of Louisiana under Contract LEQSF(1992-94)-RD-A-32. A part of this work was presented at the 22nd International Conference on Parallel Processing, Aug. 1993, and the Fifth IEEE Symposium on Parallel and Distributed Processing, Dec. 1993.

REFERENCES

[1] L.D. Wittie, "Communication Structures for Large Networks of Microcomputers," *IEEE Trans. Computers*, vol. 30, no. 4, pp. 264-273, Apr. 1981.

[2] T.-Y. Feng, "A Survey of Interconnection Networks," *Computer*, vol. 14, no. 12, pp. 12-27, Dec. 1981.

[3] C.L. Seitz, "The Cosmic Cube," *Comm. ACM*, vol. 28, no. 1, pp. 22-33, Jan. 1985.

[4] R. Arlauskas, "iPSC/2 System: A Second Generation Hypercube," *Proc. Third Conf. Hypercube Concurrent Computers and Applications*, pp. 38-42, Jan. 1988.

[5] NCUBE Corporation, "*n*-Cube 2 Processor Manual," NCUBE Corporation, 1990.

[6] W.D. Hillis, *The Connection Machine*. Cambridge, Mass.: MIT Press, 1985.

[7] Y. Saad and M.H. Schultz, "Topological Properties of Hypercubes," *IEEE Trans. Computers*, vol. 37, no. 7, pp. 867-872, July 1988.

[8] H.P. Katseff, "Incomplete Hypercubes," *IEEE Trans. Computers*, vol. 37, pp. 604-608, May 1988.

[9] N.-F. Tzeng, "Structural Properties of Incomplete Hypercube Computers," *Proc. 10th Int'l Conf. Distributed Computing Systems*, pp. 262-269, May 1990.

[10] H.P. Katseff, "Initializing Hypercubes," *Proc. Ninth Int'l Conf. Distributed Computing Systems*, pp. 246-253, June 1989.

[11] D.A. Reed and D.C. Grunwalk, "The Performance of Multicomputer Interconnection Networks," *Computer*, vol. 20, no. 6, pp. 63-73, June 1987.

[12] S. Abraham and K. Padmanabhan, "Performance of the Direct Binary *n*-Cube Network for Multiprocessors," *IEEE Trans. Computers*, vol. 38, no. 7, pp. 1000-1011, July 1989.
[13] J.-Y. Tien, C.-T. Ho, and W.-P. Yang, "Broadcasting on Incompleting Hypercubes," *IEEE Trans. Computers*, vol. 42, no. 11, pp. 1393-1398, Nov. 1993.
[14] Y. Lan, A.-H. Esfahanian, and L.M. Ni, "Multicast in Hypercube Multiprocessors," *J. Parallel and Distributed Computing*, vol. 8, pp. 30-41, Jan. 1990.
[15] A. Pombortsis and C. Halatsis, "Performance of Circuit-Switched Interconnection Networks under Nonuniform Traffic Patterns," *J. Systems and Software*, vol. 20, pp. 189-201, Feb. 1993.
[16] N.-F. Tzeng and H.-L. Chen, "Structural and Tree Embedding Aspects of Incomplete Hypercubes," *IEEE Trans. Computers*, vol. 43, pp. 1434-1439, Dec. 1994.

Nian-Feng Tzeng (S'85–M'86–SM'92) received the BS degree in computer science from National Chiao Tung University, Taiwan, the MS degree in electrical engineering from National Taiwan University, Taiwan, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 1978, 1980, and 1986, respectively.

He has been with the Center for Advanced Computer Studies at the University of Southwestern Louisiana, Lafayette, since June 1987. From 1986 to 1987, he was a member of the technical staff, AT&T Bell Laboratories, Columbus, Ohio. He is on the editorial board of *IEEE Transactions on Computers*, has served on program committees of several conferences, and is a distinguished visitor of the IEEE Computer Society. He is co-guest editor of a special issue of the *Journal of Parallel and Distribufed Computing* on Distributed Shared Memory Systems, 1995. His research interests include parallel and distributed processing, high-performance computer systems, high-speed networking, and fault tolerant computing.

Dr. Tzeng is a member of Tau Beta Pi, a member of the Association for Computing Machinery, and the recipient of the outstanding paper award of the 10th International Conference on Distributed Computing Systems, May 1990.

Harish Kumar received the BS degree in electronics and communication engineering from Regional Engineering College, Kurukshetra, India, and the MS degree in computer engineering from the Center for Advanced Computer Studies, University of Sourthwestern Louisiana, Lafayette, in 1988 and 1991, respectively.

He joined Intel Corporation in 1991 and is currently a senior engineer. He worked on high-speed VLSI routing chips for supercomputing applications and is presently involved in the development of a next-generation processor. His research interests include parallel processing architecture, high-speed clocking and clock distribution, and high-performance processor architectures.