MLH: A HIERARCHICAL HYPERCUBE NETWORK

Mokhtar A. Aboelaze *
Department of Computer Science
York University
4700 Keele st.
N.York, Ontario M3P 1J3 CANADA
E-mail aboelaze@cs.yorku.ca

ABSTRACT

In this paper, we present, and analyze, a new network called *Multi Level Hypercube* network, or MLH. The MLH is a hierarchical network that is built from small size hypercube networks. The main advantage of the MLH over the hypercube is that it require less links than a hypercube with the same number of nodes, and the node degree, for the majority of the nodes, is less than the node degree in a comparable size hypercube, (few nodes have the same node degree as a comparable size hypercube). The MLH also maintains the ease of routing enjoyed by the hypercube networks. In this paper, we investigate the MLH, and introduce a simple and efficient routing algorithm. We also introduce simple and efficient algorithms for broadcasting and multicasting in the MLH. We study the average number of hops between two nodes in the MLH assuming both uniform communication and clustered communication models. Finally, we compare the performance of our proposed network to other hierarchical networks.

Key words: parallel processing, interconnection networks, binary n-cube, Hierarchical networks, massively parallel systems

---

* This work is partially supported by a grant from the National Science and Engineering Council of Canada number NSERC-OGP0043688

1. INTRODUCTION

There has been an increasing interest in parallel processing in the past two decades. Nothing can prove this interest more than the number of multiprocessors recently built, or under development [AlG89], [BeD87], [FiC88], [Hil85], [LeL93], [Nic90]. The experience gained from designing and operating these systems has highlighted the crucial role the interconnection network plays in determining the performance of the whole system, especially in massively parallel systems with hundreds of thousands of processors. One class of networks that has received a great attention in the literature, as well as success in practice, is the direct Binary n-Cube, also known as the hypercube network [BhA84], [Pea77], [Sei85], [Sie90]. A major advantage of the hypercube network is the ease of routing. Routing in the hypercube is completely distributed, and can be performed with a minimal overhead. The hypercube network is a very attractive network when the number of processors is in the hundreds. However, when the number of processors approaches hundreds of thousands or more, the cost of building such a network is prohibitive. Some clustering techniques have been attempted to reduce the number of links in hypercube-based architecture [ChS91], [Som94].

In this paper, we propose a new interconnection network, the *Multilevel Hypercube Network* (MLH) [Abo91]. The MLH network retains the ease of routing and broadcasting enjoyed by the hypercube network, but it requires less links than a comparable size hypercube. Moreover, the node degree of the majority of the nodes is less than the node degree of a comparable size hypercube network, (few nodes still have the same node degree as in a hypercube of the same size). Another advantage of the MLH network, is that it is a hierarchical network, which makes it suitable for the connectionist models of computations [FeB82], which require a very large number of processors and its applications are hierarchical in nature [GhH89].

The organization of this paper is as follows. In section 2, we introduce the MLH network. In section 3 we introduce some properties of the MLH including the number of links and the node numbering. In section 4, we study routing in the MLH and we calculate the average distance between two nodes. In section 5, we study broadcasting, multicasting and partitioning of the MLH. Section 6 is a comparison to other hierarchical networks. Section 7 is a conclusion.

2. The MLH NETWORK

Before describing the MLH network we will, very briefly, review the hypercube network. An n-dimensional hypercube network is a network consists of $N = 2^n$ nodes. The nodes are numbered from 0 to $2^n - 1$. Two nodes whose addresses differ in exactly one bit position i, $0 \le i < n$, are connected by a link [Sie90], this link is said to connect these two nodes across the i^{th} dimension of the cube. This leads to a node degree of n, and a diameter of n.

The MLH is a modular interconnection network that consists of clusters of hypercube networks connected together in a hierarchical fashion. A MLH of height k is defined as $\{n_k , n_{k-1} , \ldots , n_1\}$-MLH with 2^n nodes, where $n = n_1 + n_2 + \ldots + n_k$. Nodes are numbered from 0 to $2^n - 1$. A node in a k-MLH can be identified as either S where $0 \le S \le 2^n - 1$ or $S = < s_k, s_{k-1}, \cdots, s_1 >$, where $0 \le s_i \le 2^{n_i} - 1$.

Definition A $\{n_k, n_{k-1}, \ldots n_1\}$-MLH of height k is formed by connecting together 2^{n_k} $\{n_{k-1}, n_{k-2}, \ldots n_1\}$-MLH's each of height $k-1$ in an n_k-cube fashion. Moreover, the 2^{n_k} MLH's are connected via the node number 0 in each $k-1$-height MLH
Recursively, a $\{n_{k-1}, n_{k-2} \ldots n_1\}$-MLH of height $k-1$ is formed by connecting together $2^{n_{k-1}}$ $\{n_{k-2}, \ldots, n_1\}$-MLH's each of height $k-2$ in an n_{k-1}-cube fashion, and so on. Finally, ($\{n\}$-MLH is an ordinary n-cube). Another equivalent but more formal definition will be introduced in the next section.

Figure 1 shows a 3 level {3,3,2}-MLH. In level 3 There are $2^3 = 8$ {3,2}-MLH networks arranged together in the form of a 3-cube. Each {3,2}-MLH network is formed by connecting together $2^3 = 8$ {2}-MLH networks in a 3-cube fashion. Finally a {2}-MLH is formed by connecting together 4 nodes in the form of 2-cube.

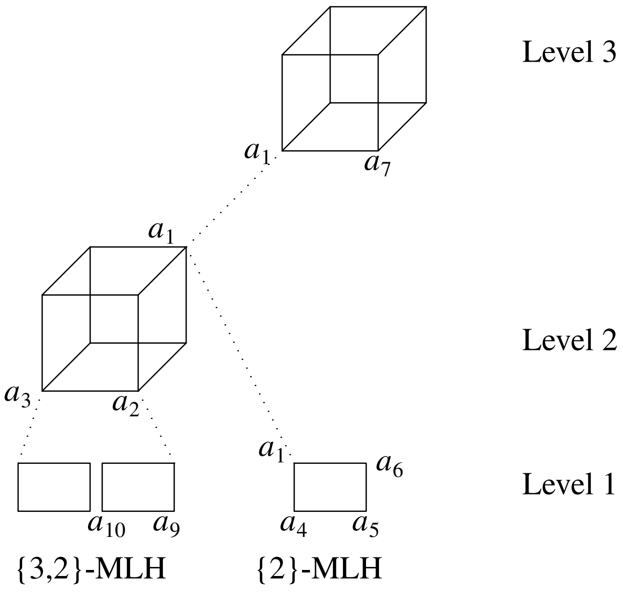

Figure 1
A {3,3,2}-MLH

The node degree in MLH is not fixed, for example in Figure 1, node a_1 is connected to three nodes in level 3, to three nodes in level 2, and to two nodes in level 1 i.e. a_1 is connected to a total of 8 nodes. Nodes a_2 and a_3 are connected to three nodes in level 2 and two nodes in level 1 i.e. a total of 5 nodes, node, while a_4 is connected to two other nodes at level 1.

The hierarchical nature of the MLH network is best illustrated by Figure 2 which shows the same {3,3,2}-MLH that is shown in Figure 1 but redrawn in a way to illustrate its hierarchical nature. Figure 2 does not show all the {3,3,2}-MLH. It just shows the partial connection for node a_1. However, in the actual network, each node at level 3 is connected to a 3-cube network at level 2, and each node at level 2 is connected to a 2-cube network at level 1.

The nodes in Figure 2 represent the processors, the solid horizontal lines represent connection between the different processors in a cube, i.e. a message from a processor to another processor actually travels across these horizontal solid lines. The dashed vertical lines do not represent any connections, they are drawn to connect together the same processor, i.e. if two nodes in Figure 2 are connected by a dashed line, this means that these two nodes are actually one node but it was drawn more than once to emphasize the fact that this node is connected to more than one hypercube network at two or more different levels. If a node is connected to more than one hypercube in more than one level, we consider it to belong to the highest level (node a_1 belongs to level 3, and a_2 belongs to level 2).

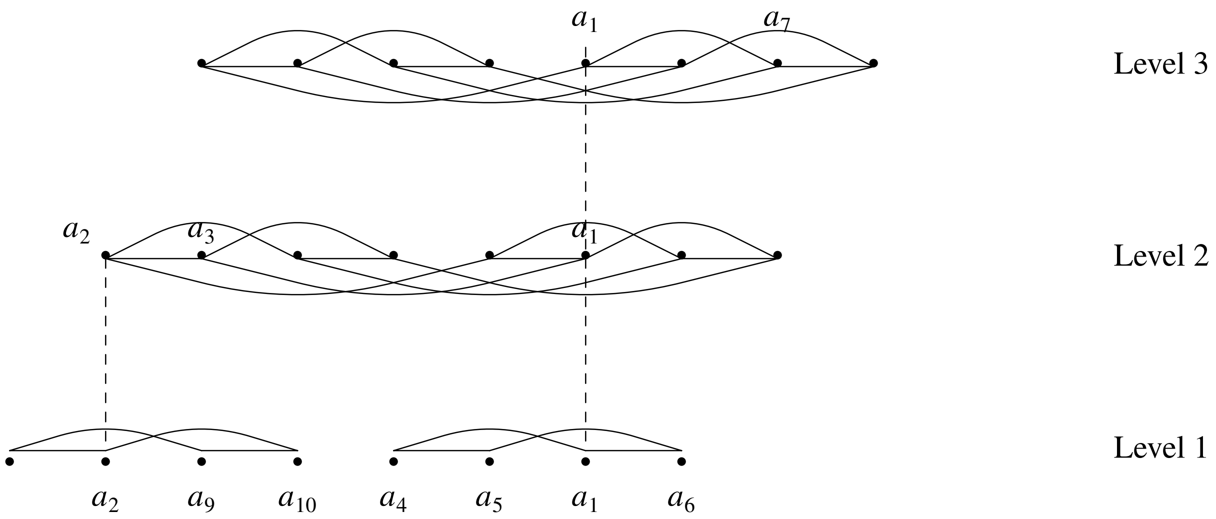

Figure 2
The {3,3,2}-MLH of Figure 2 redrawn

3. MLH PROPERTIES

In this section we will discuss the properties of the MLH network. First, we will calculate the number of links in the MLH network. Second, we will discuss how the nodes are numbered in the MLH network.

3.1. Number of links in MLH

in order to calculate the number of links in an $\{n_k , n_{k-1} , \ldots , n_1\}$-MLH, we have to understand that this number consists of two parts. The first part is the number of links in the k^{th} level, which is $(n_k)\,\frac{2^{n_k}}{2}$. The second part is the number of links in the 2^{n_k} $\{n_{k-1} \ldots , n_1\}$-MLH's at level $k-1$. Consider a sub-cube at level j , $1 \le j \le k$, the number of links in the sub-cube is.

$$C(j) = \frac{n_j}{2} 2^{n_j} + 2^{n_j} C(j-1)$$

with $C(1) = \frac{n_1}{2} 2^{n_1}$, and the total number of links in k levels MLH is $C(k)$.

3.2. Node Numbering

The nodes in $\{n_k \ldots, n_1\}$-MLH are numbered as binary numbers using $n = n_1 + n_2 + \ldots + n_k$ bits from 0 to $2^n - 1$. The n bits are grouped into k fields, corresponding to the k levels of the MLH. If the j^{th} field is n_j bits wide, this means that the hypercube at level j is of dimension n_j. The leftmost group represents the relative position of the node in the k^{th} level n_k-cube, while the rightmost group of bits represents the relative position of the node in the first level n_1-cube. Assume that the number in the i^{th} field is F_i.

Definition: A node is said to belong to level p, if all the fields F_1, $F_2 \ldots$, F_{p-1} contain 0's, while F_p is the node number in the cube at level p.

All the nodes that are descendant of a node at level p share the most significant $k - p + 1$ fields as their parent node (any node that is descendant of a node at level p inherit from it the fields $F_k, \ldots F_{p+1}, \ldots F_p$).

For example Figure 1 and Figure 2 show a (3, 3, 2) MLH with 2^8 nodes. All the nodes in the 3-cube at level 3 have an address on the form $< b_7 b_6 b_5\, 000\, 00 >$, where $b_7 b_6 b_5$ can uniquely identify the eight nodes at level 3. Assume that node a_1 address is $< xyz\, 000\, 00 >$, then all the nodes in level 2 that are descendant of node a_1 will have xyz as their three MSB's. In the same time they have 00 as their two LSB's i.e. their address is on the form $< xyz b_4 b_3 b_2 00 >$. This leads to another definition of the MLH.

Definition: Two nodes $B = < b_k, b_{k-1}, \cdots b_1 >$, and $C = < c_k, c_{k-1}, \cdots, c_1 >$ (note that b_i and c_i are not bits, they represent the number formed from the bits in the i^{th} field in the address) are connected if and only if B and C differ in one bit in their binary representation, and

$$\exists\, i\,, \quad 1 \le i \le k \quad \text{such that} \begin{cases} b_j = c_j & i < j \le k + 1 (\mathit{if\ exist}) \\ b_j = c_j = 0 & j < i \end{cases}$$

4. ROUTING

Using the above mentioned scheme of node numbering makes routing easy. In this section we will introduce a routing scheme that is both efficient and simple. The idea behind our routing scheme is as follows. If the source and destination nodes belong to the same hypercube, then the ordinary routing used in the hypercube can be used. We will assume a function called *send* $(A\,,\, j)$ is available to route a message from the current node, to, eventually, node A. The way *send* works is as follows: it chooses a bit that is different in the binary representation of the j^{th} field in the current node from the same bit in the binary representation of the j^{th} field in node A and send the message across that link (i.e. cube routing along a level j cube). Note that for this transmission to be possible, all the fields $1, \ldots, j-1$ in the current node address and in node A should be equal to 0. Moreover, the fields $j+1, j+2, \cdots, K$ in the current node should be identical to the same fields in node A.

If the source and destination nodes do not belong to the same hypercube. Then, the message should be routed up the hierarchy until it reaches a hypercube that is an ancestor of both the source and destination nodes, then it will be sent down the hierarchy to the destination node. The problem lies in determining the root of the shortest subtree that contains both the source hypercube and the destination hypercube. From Figure 2, if two nodes belong to the same hypercube at level p, then the binary representation of these two nodes will agree in all the fields F_{p+1}, F_{p+2}, . . ., F_k, and contain 0's in the fields F_1, F_2, . . . , F_{p-1}. Moreover any descendants of these two nodes will share the F_{p+1}, F_{p+2} . . ., F_k as their $k - p$ highest orders fields.

Lemma: two nodes $a_1 = < F_k, F_{k-1}, \ldots, F_1 >$, and $a_2 = < G_k, G_{k-1} \ldots, G_1 >$ have a hypercube at level i as the root of the shortest tree that contain both a_1 and a_2 iff

$F_i \neq G_i$ and

$F_j = G_j$ for $j > i$

Proof:

The proof directly follows from the numbering rule.

If a message is to be routed from node $a_1 = < F_k, F_{k-1}, \ldots, F_{i+1}, F_i, F_{i-1}, \ldots, F_1 >$ to $a_2 = < G_k, G_{k-1}, \ldots, G_{i+1}, G_i, G_{i-1}, \ldots G_1 >$, that share a cube at level i, it takes three steps

1- The message is routed up the hierarchy
   from $< F_k, \ldots F_{i+1}, F_i, F_{i-1}, \ldots F_1 > \rightarrow < F_k, \ldots, F_{i+1}, F_i, 0\ 0 \ldots 0 >$
2- Using the connections of the hypercube at level i, the message is routed
   $< F_k, \ldots, F_{i+1}, F_i, \ldots, 0\ 0 \ldots 0 > \rightarrow < G_k, \ldots, G_{i+1}, G_i, 0\ 0 \ldots 0 >$
   (notice that $F_j = G_j$ for $j > i$)
3- Finally, the message is sent down the hierarchy from node
   $< G_k, \ldots, G_{i+1}, G_i, 0\ 0 \ldots 0 > \rightarrow < G_k, G_{k-1}, \ldots, G_{i+1}, G_i, G_{i-1}, \ldots G_1 >$

4.1. Distance between two nodes

In sending a message from node $a = < F_k, F_{k-1} \ldots, F_1 >$ to node $b = < G_k, G_{k-1} \ldots G_1 >$ that share a common hypercube at level i. Then, according to the previous section the routing will require three steps

1- $< F_k, \ldots F_{i+1}, F_i, F_{i-1}, \ldots F_1 > \rightarrow < F_k, \ldots, F_{i+1}, F_i, 0\ 0 \ldots 0 >$
2- $< F_k, \ldots, F_{i+1}, F_i, \ldots, 0\ 0 \ldots 0 > \rightarrow < G_k, \ldots, G_{i+1}, G_i, 0\ 0 \ldots 0 >$
3- $< G_k, \ldots, G_{i+1}, G_i, 0\ 0 \ldots 0 > \rightarrow < G_k, G_{k-1}, \ldots, G_{i+1}, G_i, G_{i-1}, \ldots G_1 >$

The number of hops required to perform the first step is simply the number of $1's$ in the binary representations of $F_1, F_2, \ldots, F_{i-1}$

The number of hops required to perform the second step is the number of bits in the binary representation of F_i that differ from their corresponding bits in G_i

The number of hops required to perform the last step is the number of $1's$ in the binary

representation of $G_1, G_2, \ldots, G_{i-1}$
By defining $\emptyset$ as the sum of the bits in the first $i-1$ fields

$$\emptyset = n_1 + n_2 + \ldots + n_{i-1}$$

The distance between two nodes A and B is

$$D(A, B) = \sum_{j=0}^{\emptyset-1}(a_j + b_j) \; + \sum_{\emptyset}^{\emptyset+n_i-1} (\, a_j \oplus b_j \,)$$

where a_i , b_i are either 0 or 1 (bits in the binary representation of a and b respectively.
The first part represents the number of 1's in the binary representation of the first $i-1$ fields in the source and destination nodes. The second part represents the number of hops in the hypercube at level i.
The maximum distance between any two nodes (the diameter of the network) is

$$D_{\max} = \sum_{i=1}^{k-1} n_i + n_i \; + n_k \; = \; 2n - n_k$$

4.2. Average distance between two nodes

In this section, we study and analyze the average distance between two nodes in MLH in two cases, the first is when the communication between the different nodes is uniform (each node communicates with any other node with the same probability). The second case is when the communication is clustered such that the probability of communication between two close nodes (two nodes that belong to the same hypercube) is larger than the probability of communication between two nodes that don't belong to the same hypercube.

4.2.1. Uniform Communications

We assume that the probability of communication between any two nodes is the same disregarding the distance between these two nodes. If the two nodes belong to the same hypercube at level 1, then the distance is the average distance between two nodes in a hypercube at level 1. If the two nodes belong to the same hypercube at level 2, but not to the same hypercube at level 1, then, the message should be forwarded one level up the hierarchy to level 2, where it is routed to the appropriate node, and then back down the hierarchy to level 1. In this case the average distance is the average number of 1's in the 1^{st} field of the source address plus the average number of 1's in the 1^{st} field in the destination address, plus the average number of hops to rout a message in a hypercube at level 2.

To calculate the average number of hops in n-cube, we know that there are $\binom{n}{i}$ nodes at a distance i from any node. Then the average number of hops in a hypercube of dimension n can be calculated from the following formula

$$S_n = \frac{\sum_{i=1}^{n} \binom{n}{i} i}{\sum_{i=1}^{n} \binom{n}{i}} = \frac{\sum_{i=1}^{n} \binom{n}{i} i}{2^n - 1}$$

For more details about the delay in n-cube network, the reader is referred to [AbH89]. The average number of 1's in any field is simply half the length of this field. What is remaining is to calculate the probability that both sender and receiver belong to the same hypercube at level i, and not to the same hypercube at level $i-1$. Assume that we have two nodes, $< F_k, \ldots, F_1 >$ and $< G_k, \ldots, G_1 >$, the probability that these two nodes share a cube at level k as the root of the shortest subtree containing these nodes is the probability that $F_k \neq G_k$, since field k is n_k bits wide, then F_k and G_k can take any value of 2^{n_k} different values. In this case,

$$p_k = \frac{2^{n_k} - 1}{2^{n_k}}$$

p_{k-1} is the probability that these nodes share a cube at level $k-1$ as the root of the shortest subtree containing them. this is the same as the probability of $F_k = G_k$, and $F_{k-1} \neq G_{k-1}$, then

$$p_{k-1} = \frac{1}{2^{n_k}} \times \frac{2^{n_{k-1}} - 1}{2^{n_{k-1}}}$$

In general

$$p_i = \left(\prod_{j=i+1}^{k} \frac{1}{2^{n_j}} \right) \times \frac{2^{n_i} - 1}{2^{n_i}}$$

In this case, the average distance is

$$D = \sum_{i=1}^{k} p_i (S_i + \sum_{j=1}^{i-1} \frac{n_j}{2} + \sum_{j=1}^{i-1} \frac{n_j}{2}) \quad = \sum_{i=1}^{k} p_i (S_i + \sum_{j=1}^{i-1} n_j)$$

Where, S_i is the average distance between any two nodes in a hypercube at level i,the first $\sum_{j=1}^{i-1} \frac{n_j}{2}$ is the sum of the average number of 1's in the first $i-1$ fields of the source address, and the second $\sum_{j=1}^{i-1} \frac{n_j}{2}$ is the average number of 1's in the first $i-1$ fields of the destination address.

4.2.2. Clustered Communication

In this method, we drop the assumption that any node communicate with any other node with equal probability. We assume that the nodes in a hypercube communicate together with a higher probability than two nodes from two different hypercube communicate. Similarly, a node in a cube at level i communicate with a node at level $i+1$ more than it does with a node at level $i+2$. This is consistent with problem partitioning and a good policy of workload allocation in a multiprocessor system.

Table 1 shows the average distance in three different MLH's and a hypercube with 2^{12} nodes, p_i is the probability that both sender and receiver belong to the same hypercube at level i.

We assumed a communication pattern in the hypercube such that any node communicates with a probability 50% with a node that is at a distance of 2 or less from it, and with a probability 50% with the rest of the nodes (we chose these numbers at random just to show that the average distance between two nodes decrease with increasing the probability of local communication). p_i's are adjusted to reflect this communication pattern.

Network	Uniform	Clustered
12-Cube	6.0	3.96
{6,6}-MLH	8.95	6.05 $p_1 = 0.5$, $p_2 = 0.5$
{4,4,4}-MLH	9.85	7.5 $p_1 = 0.1$, $p_2 = 0.45$, $p_3 = 0.45$
{3,3,3,3}-MLH	10.28	7.7 $p_1 = 0.047$, $p_2 = 0.38$, $p_3 = 0.1$, $p_4 = 0.473$

Table 1
The average distance between two nodes (measured in average number of hops)

5. BROADCASTING AND MULTICASTING

Broadcasting was implemented in hypercube networks by forming a spanning binomial tree. A Binomial tree can be formed as follows, a binomial tree of height 0 is a single node (which is considered the root). A binomial tree of height n can be constructed from a binomial tree of height $n-1$ by adding a leaf to every node of the $n-1$ binomial tree.

Another form of data communication is known as *Multicasting*. Multicasting is a very important operation in hierarchical algorithms and architectures. In multicasting a piece of data is sent from a source node to a subset of the rest of the nodes. Broadcasting, can be considered a special case of multicasting in which this subset covers all the nodes of the architecture graph.

As we mentioned before, that, as one extreme, the n-cube graph can be considered as MLH with just one level. The other extreme, in which the MLH has $n = \log N$ levels, each with a cube of dimension 1, i.e. $(11 \cdots 1)$-MLH, or (1^n)-MLH for short. It is not difficult to prove that, not only the (1^n)-MLH is a tree, but also it is a spanning binomial tree for the n-cube with root at node 0. It is also not difficult to prove that all the links in the (1^n)-MLH are a subset of the links in any MLH with 2^n nodes. Figure 3 shows a (1^4)-MLH, which is the same as the spanning binomial tree of a 4-cube.

In the next section, we will consider two methods for broadcasting in the MLH, the first method is very simple to implement but requires $2n-1$ hops. The second method requires $2n - n_k$ hops, which is optimal for the MLH.

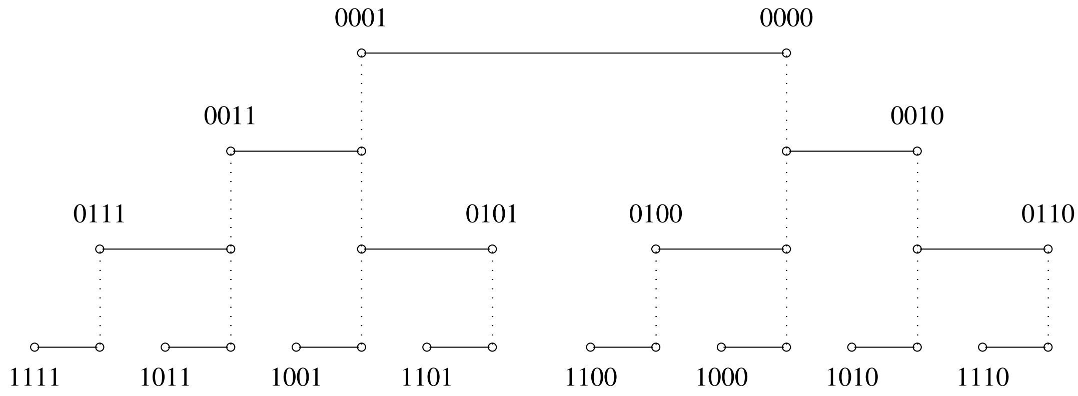

Figure 3 (1^4)-MLH

5.1. Using the (1^n)-MLH for Broadcasting and Multicasting

Since the (1^n)-MLH is a part of any MLH with 2^n nodes, and since it is the same as the spanning binomial tree for n-cube, it is easy to use it as a broadcasting tree, the idea is that a message should follow the links of the (1^n)-MLH tree until it reaches all the nodes in the network. The maximum number of links to be crossed is $2n-1$. The relation between a nodes, its parent node, and its children can be found in [HoJ86]. Consider a node $a = a_{n-1}, \ldots, a_0$ and assume that k is the position of the leftmost 1 in a, i.e. $a_k = 1$, and $a_j = 0$ for all $j > k$, then

$$children(a) = a_{n-1}, \ldots \bar{a}_m, \ldots a_0 \qquad k < m < n$$

$$parent(a) = a_{n-1} \ldots \bar{a}_k \ldots a_0$$

Our broadcasting strategy depends on knowing where the message is coming from. If the message is coming to a certain node from its parent, then pass it to all of its children. If the message is coming to a node from one of its children, pass it to the rest of the children and the parent. Assume that a node $a = a_{n-1}, \ldots, a_0$, with a_m is the leftmost 1 in the binary representation of a, received a broadcast message across link corresponding to bit z then

1 if $z \le m$ then it is arriving from the parent node and should be sent to all the children, send it to node $a_{n-1}, a_{n-2}, \ldots \overline{a_\ell}, \ldots a_0 \quad m < \ell \le n-1$

2 if $z > m$ it is arriving from a child node and should be sent to the parent node and the rest of the children, send it to nodes $a_{n-1}, \ldots, \overline{a_m}, \ldots a_0$, and $a_{n-1}, a_{n-2}, \ldots \overline{a_\ell}, \ldots a_0 \quad m < \ell \le n-1, \; l \ne z$

3 if a is the source node, send it to the parent node and all the children, i.e. send it to $a_{n-1}, \ldots, \overline{a_m}, \ldots a_0$, and $a_{n-1}, a_{n-2}, \ldots \overline{a_\ell}, \ldots a_0 \quad m < \ell \le n-1$

For multicasting, the previous algorithm could be used by limiting the algorithm to a subset of the bits in the binary representation of a node (usually, but not necessarily the algorithm is limited to the most, or least, significant i bits).

5.2. Broadcasting Using SBT

As we mentioned before, broadcasting in hypercube is done by constructing Spanning Binomial Tree (SBT) with a root at the source node and following the links of this tree to broadcast the message to all the nodes. How to construct an SBT of a hypercube network, can be found in [HoJ86].

The same strategy can be used for broadcasting in the MLH with some modification. From the architecture of the MLH, any node at level k is connected to hypercubes at levels $1, 2, \ldots k-1, k$, which means that a node at level k receiving a broadcast message it should relay the message to all the hypercubes connected to it. The modification is as follows, if a node n_1, at level k receives a broadcast message from a source s, across a link connected to a node at level j. First, it uses the SBT of level j cube to broadcast the message at level j. Then, it changes the source node to itself (n_1), and send the messages to all the cubes it is connected to except level j cube (i.e. it sends it to level i, $1 \le i \le k$, $i \ne j$).

6. Comparison to Other Hierarchical Networks

In this section, we compare our proposed network with other hierarchical networks, we chose the Hypernet [HwG87], and the Star Connected Cycles SCC [LaA93]. The reason for choosing these two networks is that they both are hierarchical networks that are similar in construction to our proposed MLH. The Hypernet is a fixed degree hierarchical network that uses a combination of a basic graph (cubes, busses, or trees), and a complete graph architecture to achieve efficient communication in multiprocessors. In our comparison, we will use the cublets based Hypernet. The Star Connected Cycles (SCC) is a fixed degree network that uses the star graph as a building block to establish hierarchical networks.

Table 2 shows a comparison between N-cube, MLH, Hypernet, and SCC. The comparison is based on the number of nodes (N), the number of links (L), the network diameter (D), and the average distance between any two nodes under a uniform communication assumption (d). Note that there is no published data on the average distance between two nodes in SCC network.

Both SCC and Hypernet have an advantage over the MLH, since both are a fixed node degree networks, while MLH is a variable node degree network. However, MLH has many advantages over these two networks that will be summarized in the rest of this section.

The SCC is a very powerful network, however, since the number of nodes in a n-SCC is $n\,!(n-1)$, this makes it difficult to accommodate a specific number of nodes. As shown in table 2 the number of nodes in n-SCC grows with n from 480, to 3600, to 30,000, and the next larger network contains more than quarter of a million nodes. Also, there is no published data on the performance of the SCC (average number of hops).

The Hypernet suffers from the same problem, for example a practical Hypernet with more than 64K nodes contains more than 2 million nodes. Although, we can construct a Hypernet with any number of nodes that is a power of 2, the network diameter and average distance becomes prohibitively high if we used more than few levels of hierarchy.

On the other hand, the MLH can be constructed from any number of nodes that is a power of 2. To compare the performance of the MLH and Hypernet, consider for example the case of a 64K node network. Comparing 64K Hypernet with a $< 1^{16} >$-MLH shows that the number of links in the MLH is almost one third that of the Hypernet (64K vs. 187K), while the performance degradation is 15% (average distance of 15 for MLH vs. 13 for Hypernet).

Another major advantage of the MLH is the wide range of choices it give to the designer with a small variation in performance. For example, a 64K node MLH could vary from a 16-cube with 263K links, and average distance of 11.98 to a tree with 64K links (same as the number of nodes), and 15.0 average distance.

7. CONCLUSION

In this paper, we have introduced a new interconnection network, the Multi Level Hypercube network or MLH. The MLH is a modular, variable node degree, network that requires less links than a comparable size hypercube. The MLH is suitable for systems with a very large number of processors. Also, because MLH is a hierarchical network, it is suitable for hierarchical systems such as the connectionist models of computations. The routing in the MLH is simple, and the network diameter is less than twice the diameter of a corresponding hypercube with the same number of nodes. We have calculated the average distance between any two nodes in the MLH in two cases, the first is uniform communication, the second is clustered communication. We also proposed algorithms for routing, broadcasting, and multicasting in the MLH network.

REFERENCES

[AbH89] M. A. Aboelaze and C. E. Houstis , "Delay Analysis in Hypercube Interconnection Network," *Great Lakes Computer Science Conference,* Kalamazoo, MI October 1989.

[Abo91] M. A. Aboelaze, "Multi-Level Hypercube Network," *Proceeddings of Fifth International Parallel Processing Symposium,* Anaheim, California April 1991, pp. 475-480.

[AlG89] G. S. Almasi and A. Gottlieb, *Highly Parallel Computing,* The Benjamin/Cumming Publishing Company, 1989.

[BeD87] J. Beetman, M. Denneau, and D. Weingarten, "The GF11 Parallel Computer," in *Experimental Parallel Computing Architecture,* J. J. Dongarra, ed., North Holland, Amsterdam, 1987.

[BhA84] L. N. Bhuyan and D. P. Agrawal, "Generalized Hypercubes and Hyperbus Structures for a Computer Network," *IEEE Transactions on Computers,* Vol. C-33, April

	Cube	MLH				Hypernet	SCC
N	128		128			128	72
		<3,4>	<3,2,2>	<2,2,2,1>	$\langle 1^{7} \rangle$		
L	448	268	172	148	127	284	128
D	7.0	10	12	13	13	9	8
d	3.5	4.9	5.65	5.84	6.0	5.0	-
N	512		512			512	480
		<4,5>	<3,3,3>	<1,2,3,3>	$\langle 1^{9} \rangle$		
L	2304	1312	876	873	511	1400	720
D	4.5	13	15	15	17	11	16
	4.5	6.4	7.3	7.3	8.0	6	-
N	4096		4096			4096	3600
		<6,6>	<4,4,4>	<3,3,3,3>	$\langle 1^{12} \rangle$		
L	24576	12480	8736	7020	4095	9584	5400
D	12	18	20	21	23	19	19
d	6.0	8.95	9.86	10.3	11	11	-
N	8192		8192			8192	
		<6,7>	<4,4,5>	$\langle 2^{6}, 1 \rangle$	$\langle 1^{13} \rangle$		
L	53248	28864	21024	9556	8191	15216	
D	13	19	21	25	25	31	
d	6.5	9.5	10.4	11.8	12	19	
N	32K		32K				30K
		<7,8>	<5,5,5>	<4,4,4,3>	$\langle 1^{15} \rangle$		
L	245K	131K	84K	57K	32K		45K
D	15	22	25	27	29		31
d	7.5	10.98	12.4	13.23	14.0		-
N	64K		64K			64K	
		<8,8>	<1,5,5,5>	$\langle 2^{8} \rangle$	$\langle 1^{16} \rangle$		
L	524K	263K	169K	87K	64K	187K	
D	16	24	27	30	31	23	
d	8.0	11.98	13.42	14.66	15.0	13	

Table 2
A comparison between MLH, Hypernet, and SCC

1984, pp. p 323-333.

[ChS91] S. B. Choi and A. K. Somani, "The Generalized Folding-Cube Network," *Networks, An International Journal,* Vol. 21, March 1991, pp. 267-254.

[FeB82] J. A. Feldman and D. H. Ballard, "Connectionist Models and their Properties," *Cognitive Science,* Vol. 6 No. 3, 1982, pp. 205-254.

[FiC88] S. A. Fineberg, T. L. Casavant, T. Schwederski, and H. J. Siegel, "Non-deterministic instruction time experiments on the PASM system prototype," *1988 International Conference on Parallel Processing,* August 1988.

[GhH89] J. Ghosh and K. Hwang, "Mapping Neural Networks Onto Message Passing Multicomputers," *Journal of Parallel and Distributed Computing,* Vol. 6, No. 2, April 1989, pp. 1-19.

[Hil85] W. D. Hillis, *The Connection Machine,* MIT Press, 1985.

[HoJ86] C. T. Ho and L. Johnsson, "Distributed Routing Algorithms for Broadcasting and Personalized Communication in Hypercube," *Proc. of International Conference on Parallel Processing,* 1986, pp. 640-648.

[HwG87] K. Hwang and J. Ghosh, "Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computers," *IEEE Transactions on Computers,* Vol. C-36, No. 12, December 1987, pp. 1450-1466.

[LaA93] S. Latifi, M. M. de Azevado, and N. Bagherzadeh, "The Star Connected Cycles: A Fixed-Degree Network for Parallel Processing," *Proceedings og the International Conference on Parallel Processing,* August 1993, pp. I-91-I-95.

[LeL93] D. Lenoski, J. Lauden, T. Joe, D. Nakahira, L. Stevens, A. Gupta, and J. Hennessy, "The DASH Prototype: Logic Overhead and Performance," *IEEE Transactions on Parallel and Distributed Systems,* Vol. 4, No. 1, January 1993, pp. 41-61.

[Nic90] J. R. Nickolls, "The Design of the MasPar MP-1: A Cost Effective Massively Parallel Computer," *IEEE Digest of Paper, Comcom,* 1990, pp. 25-28.

[Pea77] M. C. Peas, III, "The Indirect Binary n-Cube Microprocessor Array," *IEEE Transactions on Computers,* Vol. C-26, No. 5, May 1977, pp. 458-473.

[Sei85] C. L. Seitz, "The Cosmic Cube," *Communication of the ACM,* Vol. 28, 1985, pp. 22-33.

[Sie90] H. J. Siegel, *Introduction to Interconnection Networks for Large-scale Parallel Processing,* McGraw-Hill, 1990.

[Som94] A. K. Somani, "Design of an Efficient Network," *Proceedings of Seventh International Parallel Processing Symposium,* Newport Beach, CA April 13-16 1994, pp. 413-418.