Multi-level Hypercube Network

Mokhtar A. Aboelaze *
Department of Computer Science
York University
N.York, Ontario M3P 1J3 CANADA

ABSTRACT

In this paper, we propose a new interconnection network, the *Multi-level Hypercube Network* (MLH). The MLH network is suitable for connecting a very large number of processors. It retains the ease of routing and broadcasting enjoyed by the hypercube network, but it requires much less number of links than a comparable size hypercube network does. We analyze the MLH network and calculate the average distance between two nodes under two different modes of communication, we also introduce efficient routing and broadcasting algorithms for MLH.

1. INTRODUCTION

The performance of a multiprocessor system depends, to a large extent, on the interconnection network. Many interconnection networks were proposed, from the slow but inexpensive single bus, to the extremely fast and very expensive cross-bar network. In this paper, we propose a new interconnection network *Multi-Level Hypercube* network (MLH). The MLH network requires less links than a comparable size hypercube network. Another advantage of the MLH network, is that it is hierarchical network, which makes it suitable for the connectionist models of computations [3], which requires a very large number of processors and its applications are hierarchical in nature [5]. Hwang and Ghosh in [7] introduced a class of hierarchical network. However in this work we concentrate on a network that is not only hierarchical, but also enjoys the same ease of routing as in the hypercube network.

The organization of this paper is as follows. Section 2 describes the MLH. In section 3 we introduce the node numbering and routing in MLH. In section 4 we analyze the average distance between two nodes in the MLH. Section 5 deals with broadcasting in MLH.

2. MLH

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 $n = n_1 + n_2 + \ldots + n_k$.

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-hypercube fashion. Recursively, an $\{n_{k-1}, n_{k-2} \ldots n_1\}$-MLH of height $k-1$ is formed by connecting together $2^{n_{k-1}}$ $\{n_{k-2}, \ldots, n_2\}$-MLH's each of height $k-2$ in an n_{k-1}-hypercube fashion, and so on.

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 as 3*-cube*. Finally a {2}-MLH is formed by connecting together 4 nodes in the form of 2*-cube*. Notice that an n-hypercube is just an $\{n\}$-MLH of height 1.

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. While 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. One the other hand, node a_4 is connected to two nodes in level 1. In general a node in a MLH at level ℓ is connected to $\sum_{i=1}^{\ell} n_i$ other nodes (the node level is defined in section 3, for this discussion it suffice to say that a node is in level ℓ if the highest level hypercube it belongs to is in level ℓ).

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

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

TH0363-2/91/0000/0475$01.00 © 1991 IEEE

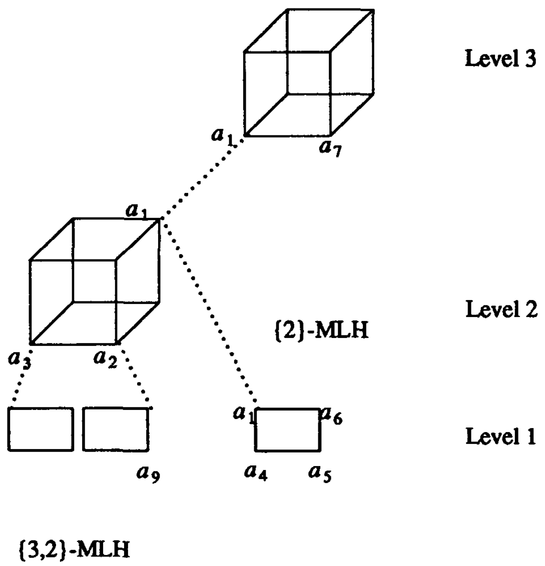

Figure 1
A {3,3,2}-MLH

the actual network, each node at level 3 is connected to a 3-hypercube network at level 2, and each node at level 2 is connected to a 2-hypercube network at level 1.

The nodes in Figure 2 represents processors, while the lines does not necessarily represent connections between the different processors. There are two kinds of lines in Figure 2. The solid horizontal lines, and the dotted vertical lines, The solid horizontal lines represent connections between the different processors in a cube, i.e. the messages actually travels across these horizontal solid lines, and this is the only actual communications that takes place in an MLH interconnection network. The dotted vertical lines does 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 dotted line, this means that these two nodes are actually one node. However, this node is 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.

To calculate the number of links in an $\{n_k, n_{k-1}, \ldots, n_1\}$-MLH. Consider a hypercube at level j $1 \le j \le k$, the number of links is the sum of the number of links in the n_j-cube, (which is $2^{n_j} n_j/2$) and the number of links in the 2^{n_j} MLH's at level $j-1$ connected to each node of the n_j-cube. i.e.

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

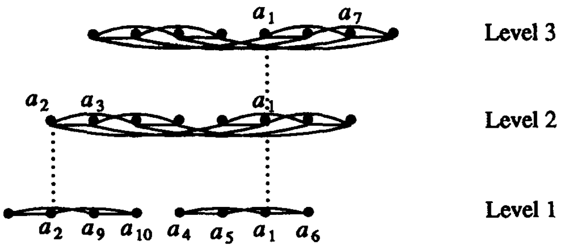

Figure 2
The {3,3,2}-MLH of Figure 1 redrawn

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

3. NODE NUMBERING AND ROUTING

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 in k fields. Field j corresponds the the position of this node in the j^{th} level hypercube, and the node number is simply the concatenation of the node numbers in the levels k ... 1.

Definition: A node is said to belong to a level p, if all the fields F_1, F_2 . . ., F_{p-1} contains 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, F_{k-1}, ... F_p).

For example Figure 1 and Figure 2 show a (3,3,2) MLH with 2^8 nodes. Each node address is represented by 8 bits, bits 0,1 represents the relative position of the node in the 2-cube at level 1. Bits 2,3,4 represents the relative position of the node in the 3-cube at level 2, Finally, bits 5,6,7 represent the relative position of the node in the 3-cube at level 3. For example, 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 have xyz as their three MSB's. In the same time they have 00 as their two LSB's (node a_2 = <xyz 011 00> and node a_3 = <xyz 111 00>. Since nodes a_4,a_5,a_6 are descendant of node a_1 they inherit the six MSB's of node a_1, i.e. their addresses are on the from <xyz 000 $b_1 b_0$>, where $b_1 b_0$ are 01 , 11 , 10 for nodes a_4,a_5,a_6 respectively. Similarly, since a_2 = <xyz 011 00>, all the nodes descendant of node a_2 inherit the six MSB's of a_2 i.e. their addresses are on the from <xyz 011 $b_1 b_0$>.

Using the above mentioned scheme of node numbering makes the routing easy. The idea behind our routing scheme is as follows. If the source and destination node belong to the same hypercube, then the ordinary routing used in the hypercube can be used. We assume a function called *send* $(a\ ,j)$ is available to route a message to node a in the hypercube at level j. i.e. it chooses at random a bit position in the j^{th} field that is different in the binary representation of a from the same bit position in the current node address, and send the message across the corresponding link

If the source and destination nodes do not belong to the same hypercube. Then, the message should be routed up the hierarchy till it reaches a hypercube that is an ancestor of both the source and destination nodes, then it will be send 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 agree in all the subfields $F_{p+1}, F_{p+2}, \ldots, F_k$

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: It is obvious from the way we choose to number the nodes that if two nodes are descendant from a hypercube at level i, then they inherit the same fields $i+1, i+2, \ldots, k$. Moreover, the i^{th} field of the address determine their relative position in the hypercube at level i, since the numbering of each node in a hypercube is unique, then $F_i \neq G_i$. To prove the second condition, notice that since the two nodes share a cube at level i, then they share the same cube at level $j > i$, and $F_i = G_j$ for $j > i$. □

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- message is routed up the hierarchy from $<F_k, \ldots F_{i+1}, F_i, F_{i-1}, \ldots F_1> \rightarrow <F_k, \ldots, F_{i+1}, F_i, 0 \ldots 0>$

2- Then using the connections of the hypercube at level i message is routed from node $<F_k, \ldots, F_{i+1}, F_i, 0 \ldots 0> \rightarrow <G_k, \ldots, G_{i+1}, G_i, 0 \ldots 0>$ (notice that $F_j = G_j$ for $j > i$)

3- Finally, the message is sent down the hierarchy from node $<G_k, \ldots, G_{i+1}, G_i, 0 \ldots 0> \rightarrow <G_k, \ldots, G_{i+1}, G_i, \ldots G_1>$

Figure 3 shows a routing procedure for MLH.

```
Procedure mlh_route(a,b)
/*
Input:      The input is two nodes a,b and their binary
            representation
            a = $<F_k, F_{k-1}, \ldots, F_1>$
            b = $<G_k, G_{k-1}, \ldots, G_1>$
            Assume a function called send($a$ ,$j$) to route
            the data to node $a$ in a hypercube at level $j$
                        */
begin       /* Find the root hypercube of the shortest
            tree that contains both $a$ and $b$          */
            j = k;
            while ($F_j = G_j$)      do        j=j-1

            /* $j$ is the level of the root of the shortest
            subtree that contain both $a$ and $b$        */
            node1 = $<F_k, F_{k-1}, \ldots, F_j, 0 \ldots 0>$
            node2 = $<G_k, G_{k-1}, \ldots, G_j\ 0 \ldots 0>$
            for ($\ell$ = 1 to j-1 step 1 ) do send(node1,$\ell$);
            send(node2,$j$);
            for ($\ell$ = j-1 to 1 step -1 ) do send(b,$\ell$);
end
```

Figure 3
Routing in MLH

4. DISTANCE BETWEEN TWO NODES

In the MLH network, the distance is defined as the number of links to be crossed in order to send a message from one node to another. In sending a message from node $A = <F_k, F_{k-1} \ldots, F_1>$ to $B = <G_k, G_{k-1} \ldots G_1>$ that share a common hypercube at level i. Then, according to the previous section the routing will require three steps

1- $<F_k, \ldots F_{i+1}, F_i, F_{i-1}, \ldots F_1> \rightarrow <F_k, \ldots, F_{i+1}, F_i, 0 \ldots 0>$

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

3- $<G_k, \ldots, G_{i+1}, G_i, 0 \ldots 0> \rightarrow <G_k, \ldots, G_{i+1}, G_i, \ldots G_1>$

The number of steps 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 steps 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 steps 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 ℓ as the sum of the bits in the first $i-1$ fields

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

The distance between two nodes A and B is

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

where $A = a_{n-1}, a_{n-2}, \ldots, a_0$, and $B = b_{n-1}, b_{n-2}, \ldots, b_0$.

The first part of the sum 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 diameter of the MLH is $2n - n_k$.

In the remaining of 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 communicate with any other node with the same probability). The second is when the communication is clustered such that the probability of communication between two close nodes (inter-cluster communication) is larger than the probability between two distant nodes (intra-cluster communication).

4.1. Uniform Communications

We assume that the probability of communication between any nodes is the same disregarding the distance between them. 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 does 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 1st subfield of the source address + the average number of 1's in the 1st subfield in the destination address, + the average number of hops to rout a message in a hypercube at level 2.

To calculate the average number of hops in n-hypercube, 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 is [1]

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

The average number of 1's in any subfield is simply half the length of this subfield. What is remaining is to calculate the probability that both sender and receiver belong to the same hypercube at level i, but not to the same hypercube at level $i-1$. Any hypercube at level 1 contains 2^{n_1} nodes, while there are 2^n nodes in the network. The probability that both the sender and the receiver belong to the same hypercube at level 1 is simply $2^{n_1}/2^n$. Similarly, the probability that the sender and the receiver belong to a cube at level 2 but not to a cube at level 1 is $(2^{n_2+n_1} - 2^{n_1})/2^n$. The probability that the sender and receiver belong to a hypercube at level i but not to a hypercube at level less than i is

$$p_i = \frac{2^{\sum_{j=1}^{i} n_j} - 2^{\sum_{j=1}^{i-1} n_j}}{2^n}$$

In this case, the average distance is

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

Where, S_{n_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. Clustered Communication

In this method, we drop the assumption that any two nodes communicate with equal probability. We assume that the nodes in a hypercube communicate together with a larger rate than two nodes from two different hypercube do. 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 a good policy of workload allocation in a multiprocessor system.

The only difference is in the calculation of the probability that two nodes with nearest common ancestor at level i communicate together, we assume that this probability is p_i, where p_i is chosen in such a way to discourage communication between distant nodes, and encourage communication between close nodes

Table 1 shows the average distance in MLH with 2^{12} nodes, p_i is the probability that both sender and receiver belong to the same hypercube at level i. the values of p_i's are chosen at random to reflect more communication between close nodes, and less communication between distant nodes. it also shows the number of links in each network.

In Table 1, we find that, while the average distance between two nodes in the 12-cube is 6.0, for the uniform communication, the average distance between two nodes in the MLH ranges from 8.95 to 10.28. While for the clustered communication, we notice much less disparity between the two networks. Also we find the average distance between two nodes in the clustered communication case is much less than the average distance between two nodes for the uniform communication, which emphasizes the suitability of the MLH network for the applications with hierarchical communication patterns.

Network	Uniform	Clustered
12-Cube 24576 links	6.0	4.24
{6,6}-MLH 12480 links	8.95	3.95 $p_1=0.8, p_2=0.2$
12-Cube	6.0	4.7
{4,4,4}-MLH 8736 links	9.87	4.8 $p_1=0.4, p_2=0.55$ $p_3=0.05$
12-Cube	6.0	4.74
{3,3,3,3}-MLH 7020 links	10.28	5.0 $p_1=0.2, p_2=0.6$ $p_3=0.19, p_4=0.01$

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

5. BROADCASTING

Broadcasting of data from a single source to all other nodes in a multiprocessor system is an important operations that is used in many applications [4]. This problem was addressed in the hypercube architecture by forming a Spanning Binomial Tree (SBT) with root in the source node, and the broadcast is done by following the nodes of this tree from the source node (root) to the rest of the nodes [6].

As we mentioned before, as one extreme, the n-hypercube 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, the (1^n)-MLH is SBT for the n-hypercube with root at node 0. It is not difficult also to prove that all the links in the (1^n)-MLH are a subset of the links in any MLH with 2^n nodes, the proof can be found in [2]. Figure 4 shows a (1^4)-MLH, which is the same as the spanning binomial tree for 4-cube.

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

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

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 till it reaches all the nodes in the network. The maximum number of links to be crossed is $2n - 1$. The relation between a node, its parent node, and its children can be found in [6]. 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

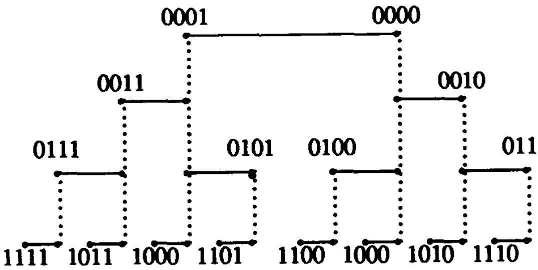

Figure 4 (1^4)-MLH

$$children(a) = a_{n-1}, \ldots \overline{a_m}, \ldots a_0 \quad k < m < n$$

$$parent(a) = a_{n-1} \ldots \overline{a_k} \ldots a_0$$

Our broadcasting strategy depends on knowing where the message are 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 $a_{n-1}, a_{n-2}, \ldots a_\ell, \ldots a_0 \quad m < \ell \le n-1$

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
$a_{n-1}, a_{n-2}, \ldots \overline{a_\ell}, \ldots a_0 \quad m < \ell \le n-1 \ , \ l \ne m$

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
$a_{n-1}, a_{n-2}, \ldots a_\ell, \ldots a_0 \quad m < \ell \le n-1$

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 for a hypercube can be found in [6].

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 $1,2, ... k-1,k$. Which means that a node at level k receiving a broadcast message it should relay the message to all the hypercubes connected to it. Figure 5 shows a (2,2)-MLH. Figure 6 shows a SBT for {2,2}-MLH with root at node 1110. Notice that, for example, when node 1000 receives a broadcast message it directs it to node 0000 and to the nodes of the 2-hypercube at level 1 (1010, 1011, and 1001).

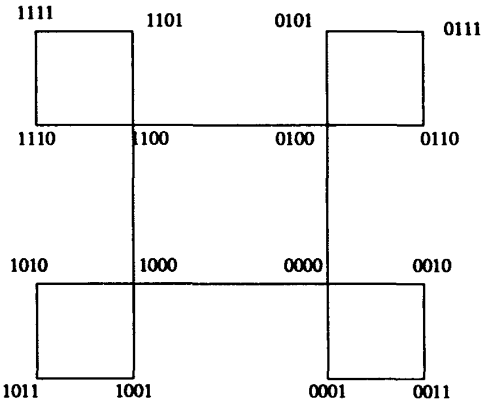

Figure 5
A (2,2)-MLH

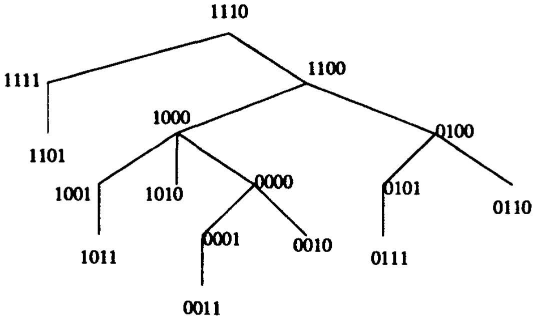

Figure 6
A SBT for (2,2)-MLH with root at node 1110

6. CONCLUSION

In this paper, we introduced a new interconnection network, the Multi-Level Hypercube network or MLH. MLH is a modular network that requires much less links than the hypercube, and the node degree (for the majority of the nodes) is less than the that of a hypercube with the same number of nodes, which makes it suitable for system with very large number of processors. Also, because MLH is hierarchical network, it is suitable for hierarchical systems such as the connectionist networks. The routing in this network is simple, and the network diameter is less than twice the diameter of a corresponding hypercube with the same number of nodes. We calculated the average distance between any two nodes in the MLH under two communication modes, uniform communication, and clustered communication. We also proposed algorithms for routing, and broadcasting, in the MLH network

7. REFERENCES

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

[2] M. A. Aboelaze, *Multi-Level Hypercube Network*, Tecnhnical Report, Dept. of Computer Science , York University, Toronto, Ontario Canada, 1991.

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

[4] D. Gannon and J. Van Rosendale, "On the Impact of Communication Complexity in the Design of Parallel Numerical Algorithms," *IEEE Transactions on Computers*, Vol. C-33, No. 12, December 1984, pp. 1180-1194.

[5] 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.

[6] 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.

[7] 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.