A Hierarchical Multistage Interconnection Network

Mokhtar Aboelaze
Dept. of Computer Science
York University
Toronto, ON. CANADA M3J 1P3
aboelaze@cs.yorku.ca

Kashif Ali
Dept. of Computer Science
York University
Toronto, ON. CANADA M3J 1P3
kashif@cs.yorku.ca

ABSTRACT

Multistage interconnection networks are used in many applications ranging from connecting processors to memory modules in a parallel processing system, to high-speed network switches and routers. One of the major drawbacks in multistage networks is the lack of proximity concept. All nodes are at the same distance from any other node. In this paper, we propose a new hierarchical Multistage Network (HMN) based on the multi-stage Omega network. The new network requires less hardware (silicon area) than the Omega network, and enjoys the same ease of routing as the Omega network. Since the HMN is hierarchical in nature, not all the nodes are at the same distance from any other node. The distance between two nodes that belong to the same cluster are less than two nodes in two different clusters. That makes the HMN suitable for applications where there are preference among the nodes. We also introduce simple and efficient algorithm for routing in the HMN and we compare the performance of the HMN to the Omega network.

Keywords

Interconnection networks, multistage networks, extended binary cube network, Omega network.

1. INTRODUCTION

Multistage interconnection networks are used extensively in multiprocessor systems [2], [5]and, in high-speed network switches[1], [4], [9]. In this paper, we propose a new hierarchical interconnection network called hierarchical multistage network (HMN). The HMN requires less links and switching elements than a comparable size hypercube network. Another advantage of the HMN network is that it is hierarchical network, which makes it suitable for application that are clustered in nature such as many digital signal processing applications.

Multistage networks have the advantages of a constant node degree. Usually the network is built from smaller $k \times k$ switching elements. These switching elements are arranged in $\log_k N$, where N is the number of inputs (and outputs) in the switch, and k is a small integer usually 2 (if 2 the switching element is a 2 x 2 switch).

One of the main disadvantages of the Omega network is that there is a fixed distance between any two nodes For binary switches the distance between any two nodes is $\log_2 N$, where N is the number of nodes. That means there is no concept of locality or two nodes *close* to each other. That is suitable for systems that require uniform communication. However, in systems where there is *clustering*, it is much more advantageous to use a network that has a concept of locality.

In this paper, we propose a new interconnection network, the hierarchical multistage network HMN. The HMN network is suitable very large systems. It retains the ease of routing and broadcasting enjoyed by the Omega network, but it requires much less number of links and much less switching elements than a comparable size Omega network. The HMN is also suitable for clustered systems where nodes communicate with a small subset of nodes much more than they communicate with other nodes in the system. Another advantage of the HMN network is that the distance between two nodes in the same cluster (measured as the number of stages to cross) is less than the distance in a comparable size Omega network. We analyze the HMN network and calculate the average distance between two nodes under two different modes of communication, we also introduce efficient routing algorithms for HMN.

The organization of this paper as follows: In chapter 2 we introduce and define the proposed network, Section 3 states some of the properties of the network. Routing in the proposed network is introduced in section 4. We discuss our simulation results in section 5, Section 6 is a conclusion of our work.

2. HMN

Before describing the HMN, let us describe the Omega Network. The Omega network (also equivalent to the indirect binary cube) is an *N x N* network with $n=log_2N$ stages. Each stage consists of *N/2 2 x 2* crossbar switches [6] [7].

One of the main advantages of the Omega network (banyan networks in general) is the ease of routing. Routing is completely distributed. Switches in the different stages are configured either as *straight* or *cross* according to the binary representation of the source and destination. There is no need for a central controller, when the message reaches the i^{th} stage, the switch examine the i^{th} bit in the source and destination addresses, according to these two bits, the switch is set as either straight or cross.

Before we start in the formal definition of the HMN, we will explain a special case of a 2 levels HMN <3,2>. The 2-level HMN will be used to illustrate the idea of the HMN, then the formal definition of the HMN will be introduced.

An HMN of height 2 is defined as $\{n_2, n_1\}$-HMN with 2^n inputs and 2^n outputs where $n = n_1+n_2$. Input and output nodes are numbered from *0* to 2^n-1, or equivalently, nodes can be represented as $s=<s_1, s_2>$ where $0 \le s_i \le 2^{n_i} - 1$. The $\{n_2, n_1\}$-HMN is constructed by connecting together 2^{n_1} networks each is an 2^{n_2} x 2^{n_2} Omega network. The connection is made via a 2^{n_1} x 2^{n_1} Omega network such that node 00…0 in each 2^{n_2} x 2^{n_2} is an input to the 2^{n_1} x 2^{n_1} Omega network. The output of the $2^{n_1} x 2^{n_1}$ is fed back to one of the different 2^{n_1} Omega networks via input 00…0 in each module.

Figure 1 shows a <3,2>-HCN where 4 8 x 8 Omega networks are connected together via a 4 x 4 Omega network such that node 000 (0) in each 8 x 8 network is the input to the 4 x 4 network. The output of the 4 x 4 network is fed back to the inputs of the 8 x 8 network via node 000 (0) in each network.

.

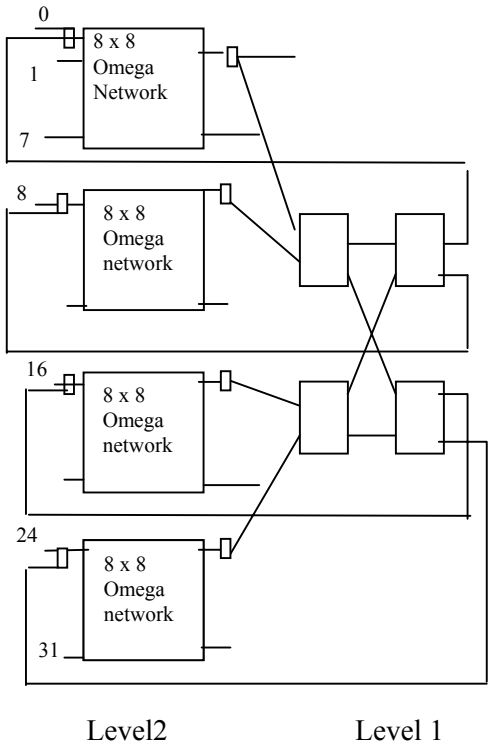


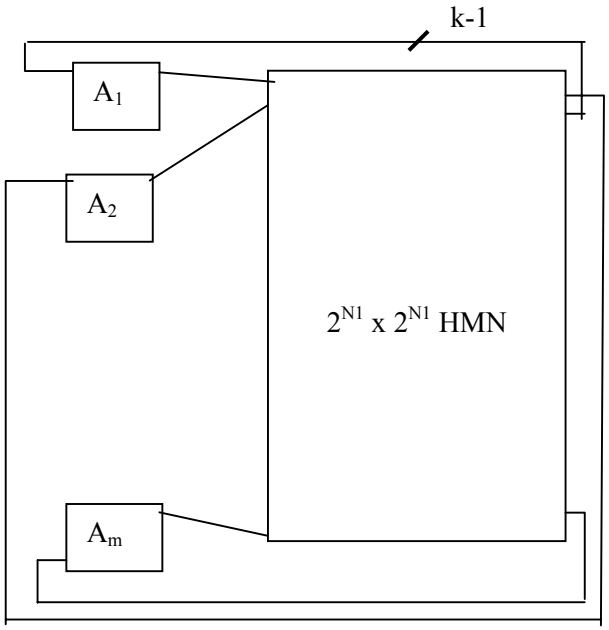
Figure 1 <3,2> HMN

To illustrate the operation of such a network, if node 0 sends a message to node 5. Both of these two nodes are in the same first stage network. The message is directed to node 5 going through 3 stages only in the 8x8 module. However, if the message is sent from node 0 to 18. Then it is routed from input 0 to output 0 in the first module. Then the 4x4 network directs it to output 2 in the 4 x 4 network. That output if feedback to input 16 in the stage 2(input 0 in the third 8-by-8 module in stage 2). From node 16 it is routed to node 18 thus passing through (3+2+3=8 stages).

Now, we can formally define the HMN.

Definition: $\{n_k, n_{k-1}, \ldots n_1\}$-HMN of height *k* with 2^N input and 2^N output nodes where $N = \sum_{i=1}^{k} n_i$, is constructed by connecting 2^{N_1} $2^{n_k} - by - 2^{n_k}$ Omega network connected together via the inputs of a $2^{N_1} - by - 2^{N_1}$ HMN, where $N_1 = \sum_{i=1}^{k-1} n_i$.

Another way to look at the HMN is as follows: in stage *i* there are 2^{N_i} modules, ($N_i = \sum_{j=1}^{i-1} n_j$) each is a is a $2^{n_i} - by - 2^{n_i}$ Omega network. Every module at stage *i* is a root of a tree with 2^{n_i} modules at stage *i+1*, and so on. That means that two leaf nodes share a root at stage *i* if and only iff they share an i^{th} stage address, and all the stages less than *i*



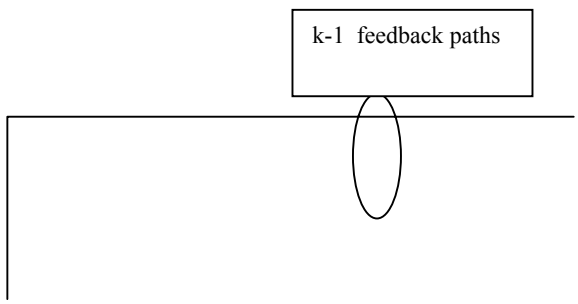
$A\ is\ a\ 2^{n_k}\ x 2^{n_k}\ Omega\ network, m = 2^{N_1}$,

$N_1 = \sum_{1=1}^{k-1} ni$

Figure 2 K-level HMN

In a mathematical form that could be expresses as follows:

Lemma: Two nodes $F=<F_k, F_{k-1}, ..., F_1>$ and $G=<G_k, G_{k-1}, \ldots G_1>$ share a module at stage *i* as a root iff $F_{i+1} \neq G_{i+1}$, and $F_j = G_j$ for $1 \leq j \leq i$ . .



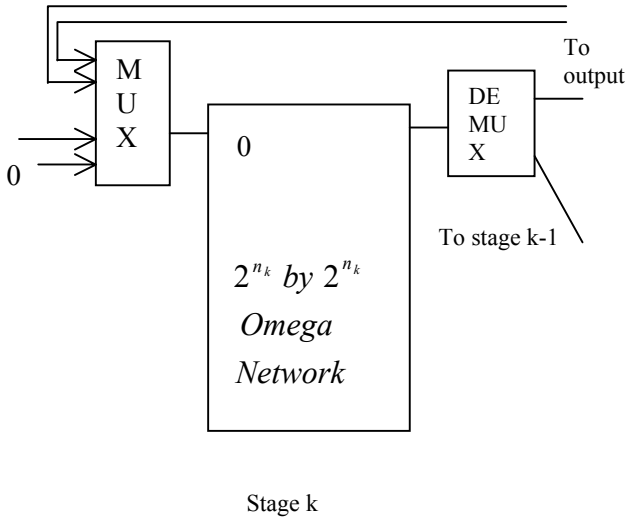


Figure 3 Input and output (de)multiplexers at stage k

Figure 3 shows a module in the k^{th} stage. Each module in the k^{th} stage is a $2^{n_k} - by - 2^{n_k}$ Omega network. There are 2^{n_k} inputs to this module and up to *k-1* feedback paths from the previous *k-1* stages. A multiplexer is used to connect the *k-1* feedback paths and input *0* of the network to input *0* of the module. The multiplexer is set to either forward a packet from the input of the HMN or from one of the feedback paths. Buffers are used to store packets if the input to the multiplexer is more than one packet. Output *0* of the module is connected to a demultiplexer. The two outputs of the demultiplexer are either the output of the HMN, or the input to stage *k-1*. Depending on if the packet is sent to the output, or it sent to stage *k-1* in order to be routed back to one of the other k^{th} stage modules, the demultiplexer is set accordingly.

For modules in stages *1,2,..,k-1* there is only an output demultiplexer at the output node 0 of each module. to determine if the message will be forwarded from stage *i* to stage *i-1* or fedback to stage *k*.

3. HMN Properties

In this section, we will discuss some of the properties of the HMN, and its cost.

3.1 Cost

One measure of the cost of a network is the number of *2 x 2* switching elements. The number of switching

elements in a $\{n_k, n_{k-1}, \ldots n_1\}$-HMN can be recursively calculated as follows: The number of switching elements up to and including these in stage *i* is equal to the number of switching elements up to and including stage *i-1* plus the number of switching elements in the i^{th} stage. By definition, at the i^{th} stage, there are $2^{\sum_{j=1}^{i-1} n_j}$ modules each is a $2^{n_i} - by - 2^{n_i}$ Omega networks. By using this definition, and the fact that the number of switching elements at the first stage is $\frac{2^{n_1}}{2} n_1$, we can recursively calculate the number of switching elements in a *k*-stage HMN. The following equation describes how to calculate the number of switching elements in a k-stage HMN

$$C(i) = C(i-1) + 2^{\sum_{j=1}^{i-1} n_j} \left(\frac{n_i 2^{n_i}}{2} \right)$$

with $C(1) = \frac{2^{n_1}}{2} n_1$ and the total number os switching elements in a *k*-stage HMN is *C(k)*.

3.2 Node numbering

The nodes in $\{n_k, , n_1\}$-HMN are numbered as binary numbers using $n=n_1 + n_2 + n_k$ bits from *0* to $2^n - 1$ The n bits are grouped in *k* fields corresponding the *k* levels of the HMN. Another way to number the nodes is as a k-tuple <F_1, F_2, …,F_k> where $0 \le F_i < 2^{n_i}$ For example node 7 in Figure 1 can be represented as <0,7> or in binary <000111>, where the left-most 2 bits represents it location in the 2 x 2 1st stage while the right-most three bits represent its position in the 8 x 8 Omega network in the second stage

3.3 Distance between 2 nodes

In HMN, we define the distance between two nodes to be the number of stages to be crossed in order to travel from the source node to the destination node. Clearly if the two nodes belong to the same cluster, the distance is less than if the two nodes belong to a different cluster.

In order to calculate the distance between two node F and G, Assume the following. $F=<F_1, F_2, \ldots, F_k>$ and $G=<G_1, G_2, \ldots G_k>$ The following procedure could be used to calculate the distance between *F* and *G*

procedure calculate_dist(F,G)

```
d=nk
for(i=1iI<k;i++) {
   if (Fi ≠Gi) {
      d = d + Σ_{j=1}^{i} nj     ;   Fi=Gi}
   }
}
```

Nodes	Organization	#of switching elements	Average distance between nodes
8	<3>	12	3
	<2,1>	9	3.25
	<1,2>	8	3.5
	<1,1,1>	7	3.5
32	<5>	80	5
	<2,3>	44	6.4
	<2,2,1>	41	7.5
	<1,1,1,1,1>	31	8
1024	<10)	5120	10
	<5,5>	2640	14.7
	<3,3,4>	1760	17.6
	<1,1,1,…,1>	1023	28

Table 1 The cost and the average distance between two nodes in HMN

Table 1 shows the cost and the average distance between two nodes in different sizes HMN and for different configuration for every size. The cost is represented as the number of 2-by-2 switching elements. The distance as stated before is the number of stages to be crosses to reach from the source to the destination.

Note that in Table 1 we considered a uniform communication where any node communicates with any other node with the same probability. As we mentioned before, HMN is suitable for clustered communication where nodes communicate with other nodes in the same cluster with a much higher probability than with nodes outside the cluster. That

of course tends to reduce the average distance between the nodes. Table 2 shows two cases from the ones mentioned in Table 1 when the communication is clustered. For every node, *p* is the probability of sending a message to any node in the same cluster, and *1-p* is the probability of a message being sent to any node outside the cluster. We can see that when *p* increases the average distance between nodes decreases even going below that of an Omega network.

P	0.1	0.2	0.4	0.6
<5,5>	14	13	11	9
<3,3,4>	16.3	14.8	11.85	8.9

Table 2 average distance between two nodes for different values of *p*

4. Routing

Routing in the HMN is completely distributed and the only information needed by any switch is the destination address only. A routing tag is attached to each message. The tag consists of *n+1* bits. *N* bits for the destination address, and one bit that we call the `forward_bit`. The use of the routing bit is explained in the next section. First, we will explain the idea of the routing algorithm, and then we formally present it.

For the k^{th} stage modules, the module has to check if the destination in the same module or not. If it is in the same module, it will be routed to the final destination. Else it will be routed to node 0 in this module with a `forward_bit` set to indicate that node 0 will set the demultiplexer to forward it to stage *k-1*.

For modules in stages *1* to *k-1*, the module checks if the destination belong to the tree rooted at this module, if yes, then there is no need to forward the message any further up the tree and the message is sent to the appropriate output with the `forward_bit` reset to indicate that the output multiplexer will feedback this message to stage *k*, else the message is sent to node 0 with the `forward_bit` set to indicate that would be forwarded up the tree to the next stage.

The routing tag in each packet consists of the destination address divided into *k* fields, where the i^{th} field contains G_i and one extra bit `forward_bit` for a total of *n+1* bits. As shown in Figure 3

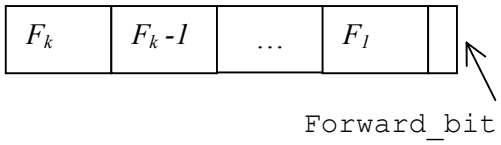


Figure 3:The routing tag in HMN

The `forward_bit` is used by the demultiplexer after node 0 in each stage. If that bit is set, the packet arriving at output *0* of this module is sent to the lower output (that means the next stage). If the `forward_bit` is reset, the packet is sent to the upper output of the demultiplexer (that means the feedback to stage *k* if the stage is stage *i,* $1 \le i \le k-1$ and is sent to the output if in stage *k*.

We present algorithm is a pseudo code format for the routing. We assume that there is already a function called *route(a,n)* that routes the message from the current node to node *a* in *n*-Omega network.

We assume that the routing is based on the routing tag which include *n* bits arranged in *k* fields $<G_1, G_2, .. G_k>$ and a `forward_bit`. We also assume that the current address of the node is $<F_1, F_2, .. F_k>$ The routing algorithm depends on the stage, for the input stage (stage *k*) the routing algorithm is as follows

Procedure routeK(G_k, G_{k-1}, .. G_1)

// Routing for node in stage k

if ($G_i = F_i$) for all i such that $1 \le i \le k-1$ Then

{Reset `forward_bit`; route(G_k,n_k)}

Else

{Set `forward_bit`; route(0,n_k)}

For nodes in stages i = 1,2,..k-1, the routing procedure is as follows

Procedure route(G_k, G_{k-1}, .. G_1,i)

// Routing for stages i= 1,2,…,k-1

if ($G_j = F_j$) for all j such that $1 \le j \le i-1$ Then

{Reset(forward_bit); route(G_i,n_k)}

Else

{Set(forward_bit); route(0,n_k)}

5. Simulation

We used simulation to simulate our proposed network. We used CSIM [8] to simulate HMN with 8,16, nodes with the following configuration <3>, <2,1>, <4>, <2,2> , <3,1> <1,3>. We assume a synchronized system where each node sends a packet in every cycle with a fixed probability p., we also assumed a clustered system where each packet is sent to a node in the same cluster with a probability p_c and to node outside the cluster with a probability $1-p_c$.

Our results indicates that as long as we are not overloading node 0 in each module, the delay is 10% more than the delay for a similar Omega network, and as you can see in Table 1 the cost is much less than the cost of a comparable size Omega network.

6. Conclusion

In this paper we presented a hierarchical multistage interconnection network. The proposed network enjoys the same ease of routing used in the Omega network and has a less number of switching elements than a comparable size Omega network. Our simulation and analytical results show that the performance of the proposed system (both in terms of the average distance between nodes, and total delay) is better than the Omega network for clustered systems, and slightly worse than the Omega network for non-clustered systems.

7. References

[1] Aboelaze, M; Mnaour, A; Elnaggar, A; Dynamic cell allocation to input queues in a combined I/O buffered ATM switch, Proceedings of Internet computing 2001,

[2] Bhuyan, L.N.; Iyer, R.R.; Askar, T.; Nanda, A.K.; Kumar, M Performance of multistage bus networks for a distributed shared memory multiprocessor IEEE Transactions on Parallel and Distributed Systems, Vol 8, No. 1, Jan. 1997 pp 82-95

[3] Chaney, T. Fingerhut, J. Flucke, M. and Turner J. "design of a gigabit ATM switch" Proc. IEEE INFOCOM, April 1997.

[4] Morino, H.; Bao, T T; Hoaison, N; Aida, H.; Saito, T. A scalable multistage packet switch for terabit IP router based on deflection routing and shortest path routing, IEEE International Conference on Communications, 2002, vol. 4, pp 2179-2185

[5] Omran, R.A.; Aboelaze, M.A., An efficient single copy cache coherence protocol for multiprocessors with multistage interconnection networks, Scalable High-Performance Computing Conference, 1994, pp 1-8

[6] Siegel, H. J. Hsu, W. T.-Y, and Jeng, M. An introduction to the multistage cube family of interconnection networks, "The Journal of Supercomputing", Vol. 1, No. 1, pp 13-42, 1987

[7] Siegel, H.J. Introduction to networks for large scale parallel processing, McGraw-Hill 1990.

[8] Schwetman, H. "CSIM User's Guide" Mesquite Software Inc Austin TX 1998.

[9] Yang, Y, and Wang J. A class of multistage conference switching networks for group communication, International conference on parallel processing, 2002 pp 73-80

.