Journal of Information & Communication Technology
Vol. 2, No. 1, (Spring 2008) 40-48

Fast Four Heuristic Routing Algorithms in Optical Multistage Interconnection Switch Networks

Mohamed Othman*
Farzaneh Abed*
Department of Communication Technology and Network
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, D.E., Malaysia

ABSTRACT

One undesirable problem introduced by the Optical Multistage Interconnection network is crosstalk that is caused by coupling two signals with each other in switching element. One way to solve this problem is to avoid coupling two signals within the same switching element. But before any attempts, a method should be used to find out which message should not be in the same group. After finding conflicts and generating the conflict matrix, an algorithm is used to route the messages in different passes with respect to the conflict matrix. The purpose here is having less execution time. In this paper, four Heuristic routing algorithms are improved to have less execution time.

INSPEC Classification : B4125; B4185; B6260M; C3370R; C5620W

Keywords : Optical Multistage Interconnection Networks, Four Heuristic Algorithm, Omega Network

1) INTRODUCTION

Optical Multistage Interconnection Network (OMIN) is popular in switching and communication applications and has been studied extensively as an important interconnecting scheme for communication and parallel computing systems. The OMIN is frequently proposed as connections in multiprocessor systems or in high bandwidth network switches (Varma, 1999). A major problem in OMIN is a crosstalk. It is caused by coupling two signals within a switching element.

To reduce the negative effect of crosstalk, various approaches which apply the concept of dilation in either the space or time domain have been proposed (Pa0manabhan, 1987)(Qiao,1994). With the space domain approach, extra switching elements (SEs) (and links) are used to ensure that at most one input and one output of every SE will be used at any given time. With the time domain approach, the same objective is achieved by treating crosstalk as a conflict; that is, two connections will be established at different times if they use the same SE. Whereas we want to distribute the messages to be sent to the network into several groups, a method is used to find out which messages should not be in the same group because they will cause crosstalk. The Window Method (WM) (Shen, 2001) is used to find conflicts among all messages to be sent. Except a few attempts, authors have not found any improving of WM (Munir, 2005).

* The material presented by the authors does not necessarily portray the viewpoint of the editors and the management of the Institute of Business and Technology (BIZTEK) or Universiti Putra Malaysia.

* Mohamed Othman : mothman@fsktm.upm.edu.my
* Farzaneh Abed : feri.abed@gmail.com

Many algorithms for routing on optical Omega networks have been proposed (shen, 2001)(Munir, 1999). The purpose of these routing algorithms is to route the messages in order to avoid the path conflict in the network. These routing algorithms evaluated by the average number of passes and execution time.

The Omega network (ON) is an example of OMIN that is used in this research, ON has N input (N is the network size) signals and has log2N stages of N/2 bypass-exchange switches and a perfect-shuffle interconnection pattern between them (Pan,1999). Since many other topologies are equivalent to omega topology, performance results obtained for ON are also applicable to other MINs (Yang, 2000).

2) WINDOW METHOD

Window Method is a technique used to find which messages should not be in the same group because they introduce crosstalk in the network (Shen, 2001)(Pan,1999). WM can be described as follows:

For network size N*N, there are N source and N destination address. Each source and its corresponding destination address is combined to produce a combination matrix. From this matrix, the optical window size is M-1 where M= log 2N and N is the size of the network.

This window is used in the combination matrix from left to right except first and last column. If two messages have the same bit pattern, they will cause conflict in the network. Hence, they must be routed in different passes. To see how the WM works, refer to the following example. The network size is 8*8 and permutation is shown in Figure 1:

Figure 1
Permutation in binary format

Src		Dest
000	→	1 0 1
001	→	0 0 1
010	→	0 1 1
011	→	1 1 0
100	→	0 0 0
101	→	0 1 0
110	→	1 0 0
111	→	1 1 1

Using the window method is demonstrated in Figure 2. The window size is M-1=2 (M= $\log_2 8$=3) and the number of windows is M=3 (w_0, w_1, w_2).

Figure 2
Optical Window Method

0	0 0	1 0 1	message 000 and 100 have conflict
0	0 1	0 0 1	message 001 and 101 have conflict
0	1 0	0 1 1	message 010 and 110 have conflict
0	1 1	1 1 0	message 011 and 111 have conflict
1	0 0	0 0 0	
1	0 1	0 1 0	
1	1 0	1 0 0	
1	1 1	1 1 1	

Step 1(w0)

```
0 0 0 1  0 1        message 000 and 110 have conflict
0 0 1 0  0 1        message 001 and 101 have conflict
0 1 0 0  1 1        message 010 and 100 have conflict
0 1 1 1  1 0        message 011 and 111 have conflict
1 0 0 0  0 0
1 0 1 0  1 0
1 1 0 1  0 0
1 1 1 1  1 1
```

Step 2(w1)

```
0 0 0 1 0  1        message 000 and 110 have conflict
0 0 1 0 0  1        message 001 and 100 have conflict
0 1 0 0 1  1        message 010 and 101 have conflict
0 1 1 1 1  0        message 011 and 111 have conflict
1 0 0 0 0  0
1 0 1 0 1  0
1 1 0 1 0  0
1 1 1 1 1  1
```

Step 3(w2)

The pseudo code of WM is as shown in Figure 2.

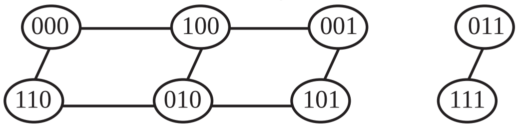
Figure 2
The pseudo code of WM

```
Window _Method ()
For l= 1 to m+1                                  // m is the number of stages
      For i=0 to N-1                               // N is the network size
         For k=1 to l+m-1 do
             A [k-1] =Combination_Matrix[i] [k];
         End for;
      For j=i+1 to N-1
         For k=1 to l+m-1 do
            B [k-1] =Combination_Matrix[j] [k]
         End for;
      If (A=B)
             Conflict_Matrix[i] [j] =1;                  //there is conflict
      End if;
      End for;
      End for;
End for;
```

3) CONFLICT GRAPH AND CONFLICT

After using a window method to find a conflict, the conflict graph can be shown like Figure 3 (QIAO, 1994).The number of nodes is the size of the network. If two nodes have conflict during routing, the nodes connected using an edge. To routing the messages in the network the degree of each message is the number of conflicts to other messages. For example Figure 3 shows the conflict graph of Figure 1.

Figure 3
Conflict graph



The conflict matrix is a square matrix with $N*N$ entry, it consists of the output of the window method .The definition of Conflict Matrix is the matrix M_{ij} with size $N*N$ that N is the size of the network.

$$M_{ij} = \begin{cases} 1 & \text{if conflict} \\ 2 & \text{if no coflict} \end{cases}$$

Table 1
The Conflict Matrix

Msg	000	001	010	011	100	101	110	111
000	0	0	0	0	1	0	1	0
001	0	0	0	0	1	1	0	0
010	0	0	0	0	1	1	1	0
011	0	0	0	0	0	0	0	1
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0

M_{ij} is the entry of the conflict matrix. M_{ij} assigned with value 1 when there is a conflict between the source and destination addresses otherwise the values is 0. Table 1 shows the conflict matrix of combination matrix in Figure 1.

For example in the first row because message 000 with 100, 101 with 110 have a conflict the value of these columns is assigned to 1 and the value of other columns is assigned to 0 because 000, 001, 010, 011, 101 and 111 do not have any conflict with 0. This function is a same for other rows.

4) PREVIOUS ROUTING ALGORITHMS

Many routing algorithms have been proposed in optical Omega networks (Yang,2000)(Pan,2002). The purpose of these routing algorithms is to route the messages in order to avoid the path conflict in the network.

Four Heuristic algorithms contain SeqInc, SeqDec, DegAsc and DegDsc, Genetic Algorithm Simulated Annealing (Mohammad et.al.,2005)`, Remove Last Passes (Siu,2004) and Zero Algorithm are the latest routing algorithms.

5) FOUR HEURISTIC ROUTING ALGORITHMS

The purpose of the routing algorithm is to schedule the messages in different time slots in order to avoid path conflict in the network. Generally, a routing algorithm has several phases, which are repeated to schedule all messages in the graph. In the first phase, has to be selected in order to be scheduled. In the second phase, the message is scheduled in a time slot.

Four heuristic routing algorithms include: Sequential Increasing, Sequential Decreasing, Degree Ascending and Degree Descending. Four strategies are used for selecting the message:

1. Selecting a message sequentially in increasing order of source addresses.
2. Selecting a message sequentially in decreasing order of source addresses.
3. Selecting a message based on the order of increasing degrees in the conflict graph.
4. Selecting a message based on the order of decreasing degrees in the conflict graph.

After selecting the appropriate message, assigning a time slot for the message that can be scheduled by several strategies:

1. Schedule a free time slot for a message in increasing order. If the message has a conflict with message in each of the existing time slots, then a new time slot is allocated to the message, otherwise, the first available time slot is selected for the message.
2. Schedule a time slot for a message in the decreasing order. If the message has a conflict with message in each of the existing time slots, then a new time slot is allocated to the message.
3. Schedule a time slot for a message randomly among time slots already scheduled for other messages. If the message has conflict with at least one message in each of the existing time slots, then a new time slot is allocated to the message.

6) BITWISE FOUR HEURISTIC ALGORITHMS

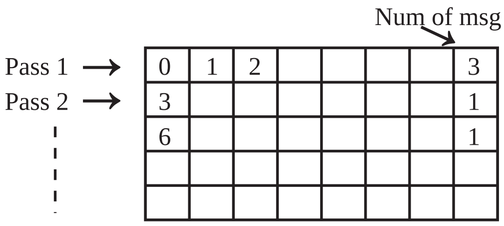
Four Heuristic algorithms contain SeqInc, SeqDec, DegAsc and DegDsc. As pointed out in the literature review, until now no researches has tried to improve the heuristic algorithms in terms of execution time.

In bitwise Four Heuristic (BFH) algorithms, the bitwise operations are used to route the messages. The efficiency of these algorithms is much better than the standard four heuristic algorithms. The algorithm of all heuristic algorithms can be described in short as fallows:

- Build a routing matrix by maximum 10 rows and columns (number of network size (10*N)).
- Put the messages in first pass; check the conflicts in the first pass.
- If there is conflict with the other messages in current pass, remove the message from that pass and try next passes until the messages fits into one suitable pass.

Refer to Figure 4 as an example to describe the mentioned algorithm.

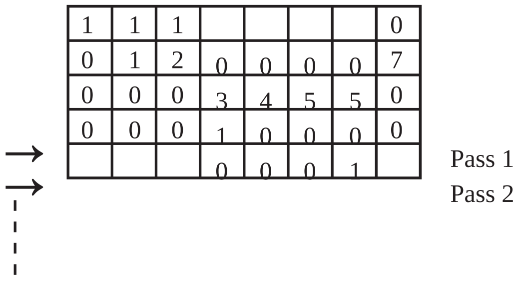
Figure 4
Message Routing in Heuristic Algorithms



The example in Figure 4 shows the routing matrix. Each row consists of a pass and the last column indicates how many messages routed in each pass. For instance, if the algorithm tries to route node 4, it tries first pass consisting of nodes 0, 1, 2. The conflict of nodes 4 is checked with all messages in that pass. In case of any conflict, node 4 moves to the next pass.

The difference of the heuristic algorithms is choosing order of messages to route in the network. In all heuristic algorithms the bottleneck is where the program has to find conflict in each pass with the new routed messages. To improve the heuristic algorithms a new routing matrix was defined. A bitwise matrix with maximum 10 rows and N columns is built. This matrix has only digital element (0 and 1) (Figure 5).
To overcome the mentioned bottleneck in heuristic algorithms, the algorithm should simply add the current pass to the row of conflict matrix with the index equal to the routed message number bitwisely. In the present algorithm, instead of several one by one comparisons there is only one comparison in each time.

Figure 5
New Routing Matrix for 8*8 Network Size



6.1 Bitwise Sequential Increasing and Decreasing Algorithm

In BSeqInc algorithm the message is selected sequentially in the increasing order of the source address as same as SeqInc. Consequently, applying the bitwise operation in SeqInc reduces the execution time more than 10 times.

BSeqDec algorithm selects the message sequentially in the decreasing order of the source address to route them in OMIN. Accordingly, applying the bitwise operation in SeqDec reduces the execution time because of bitwise operation essence in speed and simplicity.

6.2 Bitwise Degree Ascend and Degree Descending Algorithm

In BDegAsc a message choose based on the order of the increasing degrees in the conflict graph. The degree of each message is the number of conflicts to other messages.

BDegree Descending algorithm is a same as BDegree Ascending but this chooses a message based on the order of the decreasing degrees in the conflict graph.

7) EXPERIMENTAL RESULT AND DISCUSSIONS

Comparison result of SeqInc and BSeqInc, SeqDec and BSeqDec is presented in Figure 6 and Figure 7 respectively. Also Figure 8 and Figure 9 show the graph of DegAsc and BDegAsc, DegDes and BDegDes sequentially.

Concerning to Figure 7, it is clear that the execution time of BDegDes is less than the DegDes. For example the time of BDegDes for network size 256*256 is 9.562 milliseconds in case this time for DegDes is 31.317 milliseconds.

From the experiment, the proposed BFH improved the execution time around 10 times as compared to the time of four heuristic. These are due to the fact that bitwise operations are very fast. BSeqDec has the best performance between bitwise four heuristic algorithms. Figure 10 shows the performance of mentioned algorithms.

Figure 6
Execution Time of SeqInc and BSeqInc

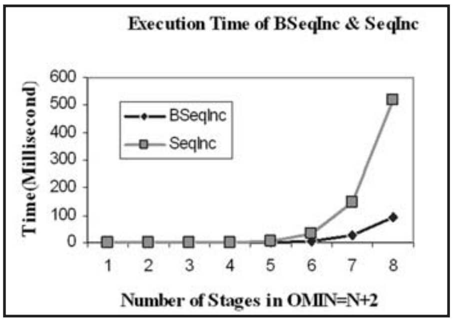


Figure 7
Execution Time of SeqDec and BSeqDec

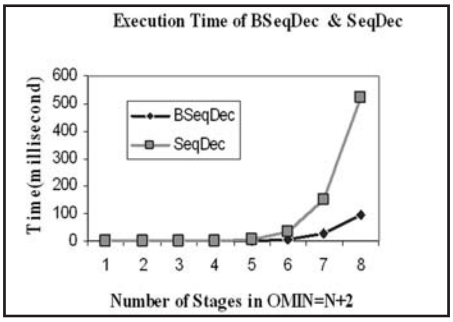


Figure 8
Execution Time of DegAsc and BDegAsc

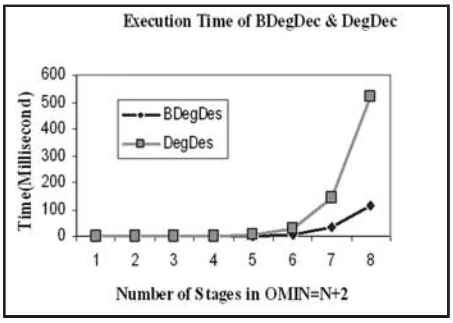


Figure 9
Execution Time of DegDes and BDegDes

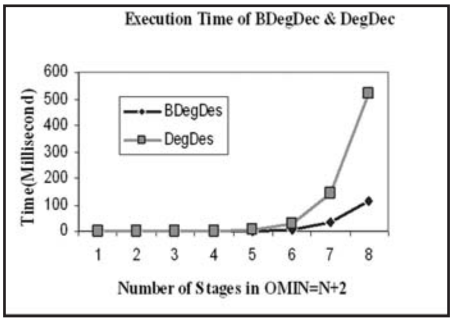
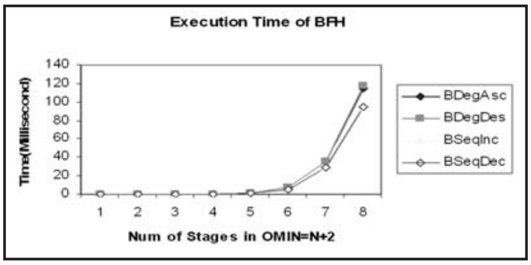


Figure 10
Execution Time of BFH



7) CONCLUSION AND FUTURE WORK

In this paper, we used bitwise operations to improve the performance of routing in OMIN. To compare the performance of BFH to find a conflict with previous FH, the results are consistent with our intuition. The BFH can improve the time nearly more than 10 times special when the network size is large. Efficient message routing algorithms directly affect the performance of communication networks.

Our future research topic is running the bitwise routing algorithms code on parallel computer to reach the minimum execution time of routing.

8) REFERENCES

VARMA, A. AND RAGHAVENDRA, C. S.(1994). Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice.IEEE Computer Society Press.

SHEN, X., YANG, F., AND PAN, Y. (2001). Equivalent permutation capabilities between time division optical omega networks and non-optical extra-stage omega networks.IEEE/ACM Trans. Netw. Vol. 9, No. 4, pp. 518 -524.

MUNIR, A, MOHAMED, O., AND ROZITA,J. (2005). An efficient approach to avoid crosstalk in optical Omega Network. International Journal of Computer, Internet and management, Vol.14, No.1, pp.50-60

KATANGUR, A. K. PAN, Y., AND FRASER, M. D. (2002). Message Routing and Scheduling in Optical Multistage Networks Using Simulated Annealing. International Proceeding of the Parallel and Distributed Processing Symposium (IPDPS).

PAN, Y. QIAO, C. AND YANG, Y.(1999). Optical Multistage Interconnection Networks:

New Challenges and Approaches, IEEE Communications Magazine, Feature Topic on Optical Networks, Communication Systems and Devices, Vol. 37, No. 2, pp. 50 -56.

YANG Y., WANG, J. AND PAN, Y.(2000). Permutation Capability of Optical Multistage Interconnection Networks, Journal of Parallel and Distributed Computing (JPDC), Vol. 60, No. 1, pp. 72-91.

PAN, Y. JI, C. LIN, X. JIA, X.(2002). Evolutionary Approach for Message Scheduling in Optical Omega Networks, Fifth International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP).

MOHAMMAD, A.A. MOHAMED, O., ROZITA, J. AND SHAMALA, S. (2005). New Algorithm to avoid Crosstalk in Optical Multistage Interconnection Networks, Proceeding of IEEE International Conference on Network(MICC-ICON 2005) Vol. 1,pp.501-504.

SIU-CHEUNG, C. AND TIEHONG, X.(2004). New Algorithm for Message Routing and Scheduling in Optical Multistage Interconnection Network, Optical Communications Systems and Networks, Proceeding International.

PADMANABHAN, K. AND NETRAVALI, A. N.(1987). Dilated networks for photonic switching, /EEE Trans Commun ,Vol 35, No 12, pp 1357-1365.

QIAO, C.(1994). A time domain approach for avoiding crosstalk in optical blocking multistage interconnection networks, Journal of Lightwave Tech, vol 12, No 10, pp 1854-1862.