pISSN 1976-913X
eISSN 2092-805X

J Inf Process Syst, Vol.9, No.2, June 2013 http://dx.doi.org/10.3745/JIPS.2013.9.2.301

On the Minimization of Crosstalk Conflicts in a Destination Based Modified Omega Network

Ved Prakash Bhardwaj* and Nitin*

Abstract—In a parallel processing system, Multi-stage Interconnection Networks (MINs) play a vital role in making the network reliable and cost effective. The MIN is an important piece of architecture for a multiprocessor system, and it has a good impact in the field of communication. Optical Multi-stage Interconnection Networks (OMINs) are the advanced version of MINs. The main problem with OMINs is crosstalk. This paper, presents the (1) Destination Based Modified Omega Network (DBMON) and the (2) Destination Based Scheduling Algorithm (DBSA). DBSA does the scheduling for a source and their corresponding destination address for messages transmission and these scheduled addresses are passed through DBMON. Furthermore, the performance of DBMON is compared with the Crosstalk-Free Modified Omega Network (CFMON). CFMON also minimizes the crosstalk in a minimum number of passes. Results show that DBMON is better than CFMON in terms of the average number of passes and execution time. DBSA can transmit all the messages in only two passes from any source to any destination, through DBMON and without crosstalk. This network is the modified form of the original omega network. Crosstalk minimization is the main objective of the proposed algorithm and proposed network.

Keywords—Optical Multistage Interconnection Network, Crosstalk, Time Domain Approach, Omega Network, Destination Based Modified Omega Network, Crosstalk Free Modified Omega Network

1. INTRODUCTION

The Multi-stage Interconnection Network is an essential network for parallel computing applications. It connects N inputs to N outputs and is known as an $N x N$ MIN [1-4]. The parameter N is called the size of the network. It is a class of Dynamic Interconnection Networks. MIN can be one-sided, where both inputs and outputs are on the same side, or it can be two-sided, where inputs and outputs are on the opposite side of the network. The two-sided MIN can be a rearrangeable, blocking and nonblocking network [4-8]. Although MIN is very popular in the field of parallel processing, in the present scenario with an increasing demand for bandwidth, an Optical Multi-stage Interconnection Network is used. The conventional system uses electronic signals in switching, but optical communication uses optical signals. This interconnection has proficient potential and it offers better performance than the electrical interconnections [9-18].

Manuscript received March 6, 2013; accepted May 12, 2013.
Corresponding Author: Dr. Nitin
* Dept. of CSE and ICT, Jaypee University of Information Technology, Waknaghat, Solan-173234, Himachal Pradesh, India (ved.juit@gmail.com, delnitin@ieee.org)

Crosstalk is the major shortcoming of this interconnection network [9-13].

Crosstalk may be electrical or optical. Electrical crosstalk [12, 15] is known as link conflict [12, 15] and optical crosstalk [12, 15] is known as switch conflict [12, 15]. In link conflict [12, 15], two or more communication signals or messages follow the same path in the same time slot. In switch conflict, two or more messages interact with each other within the same switching element (SE) in the same time slot. The term "crosstalk" includes both link and switch conflict and we have minimized both conflicts in this paper. In this paper, we compare the performance of CFMON [18] and DBMON against the crosstalk problem.

1.1 Optical Omega Network (OON)

The OON has a shuffle exchange connection pattern. In this pattern the address is shifted one bit to the left circularly in each connection. This network connects the N input to N output nodes using n stages, where $n = log_2 N$ with each stage containing 2^{n-1} SEs. When any source sends the communication signal to a destination then it has to pass through the OON. Each communication signal has a definite path from the given source to the given destination. Crosstalk occurs when two or more signals follow the same path in the same time period. In Fig. 1 the dotted black arrows show the switch conflict and the solid black arrows show the link conflict problem in OON.

The remainder of this paper is organized as follows. In the second section, related work and

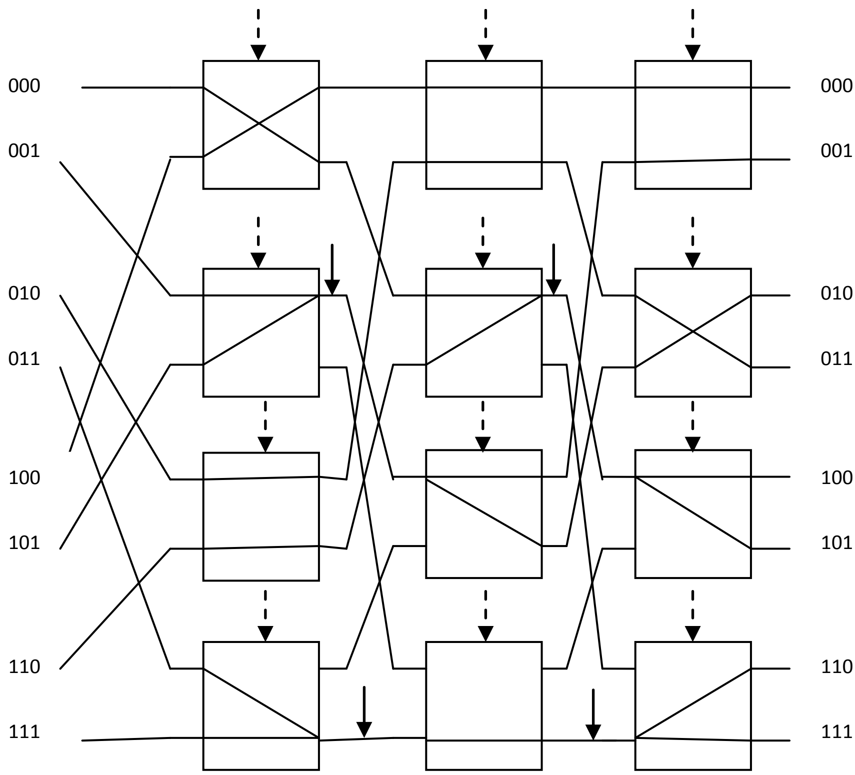

Fig. 1. 8×8 OON

Table 1. Symbol Table

Symbol	Meaning of the Symbol
SA	Source Address
DA	Destination Address
Address	SA and its corresponding DA
DB	Destination Bit
NOP	Number of Passes
USW	Upper Straight Way
LUS	Lower to Upper Straight Way
LSW	Lower Straight Way
ULS	Upper to Lower Straight Way
NOP	Average Number of Passes
ET	Execution Time

the theoretical background about previously proposed algorithms is presented. We describe our interconnection network and its routing algorithm in Section 3. In Section 4, we present the performance evaluation parameters. In Section 5, the performance of DBMON and CFMON is compared, and finally we give our conclusions in Section 6.

Before moving to the next section, let's have a look are the symbols that are used in various places in the paper. These symbols are presented in Table 1.

2. Related Work

Using any one of the following three techniques we can minimize the crosstalk: the Space Domain Approach [12-20], the Wavelength Domain Approach [12-20], and the Time Domain Approach (TDA) [12-20]. The Time Domain Approach [13, 14] reduces the crosstalk problem by allowing only one source and its corresponding destination address to be active at a time within a SE in the network. The Window Method, the Improved Window Method, and many other TDA based approaches have come into limelight in recent years.

The aim of ASA [11] is to select a particular source address that does not create conflict in the network in the first pass, and the remaining source address can be transmitted in the second pass. This algorithm is applicable for the 8 x 8 Optical Multistage Interconnection Networks [11]. Initially, in this algorithm, the SA and DA are obtained sequentially. Next, the combination matrix of the source and corresponding destination address is obtained. Furthermore, transformation and row selection operations are applied on the combination matrix. In this way, two pairs of rows can be obtained [11]. In the next step, addition and subtraction operations will be performed between the corresponding bits in each pair. Finally, some SAs and their DAs are selected for a current pass and again the ASA is applied to the rest of the addresses [11].

The routing process of RSA [12] is little bit different from ASA [11]. This algorithm emphasizes two operations (i.e., column selection and the construction of a conflict matrix table that is based on these columns). The rest of the operations of RSA are same as in ASA. Furthermore, providing crosstalk free routes in minimum passes is an exigent problem. For this problem, CFMON [18] and its routing algorithm [18] were proposed in [18]. After going through [18], we found that the routing algorithm of CFMON does not provide the crosstalk-free routes in two passes in some exceptional cases. Hence, we have compared our research work with CFMON [18]. A short description on CFMON is given in the next subsection.

2.1 The Crosstalk-Free Modified Omega Network

The structure of this network is based on the optical omega network [14-16] and the optical clos network [18]. Fig. 2 depicts the connection pattern of CFMON. The output links of the SEs of the first stage and the input links of the SEs of the middle stage follow the connection pattern of OON [18]. Similarly, the output links of the SEs of the middle stage and the input links of the SEs of the last stage follow the connection pattern of the optical clos network [18]. Furthermore, in the routing process of CFMON, the SAs, which are in green color, will send the communication signal to their given destination [18]. In the second pass, the SAs that are red will transmit the communication signal to the given destination [18]. In case of crosstalk, the conflicting address pair will be transmitted in separate passes. However, these solutions increase the number of passes [18].

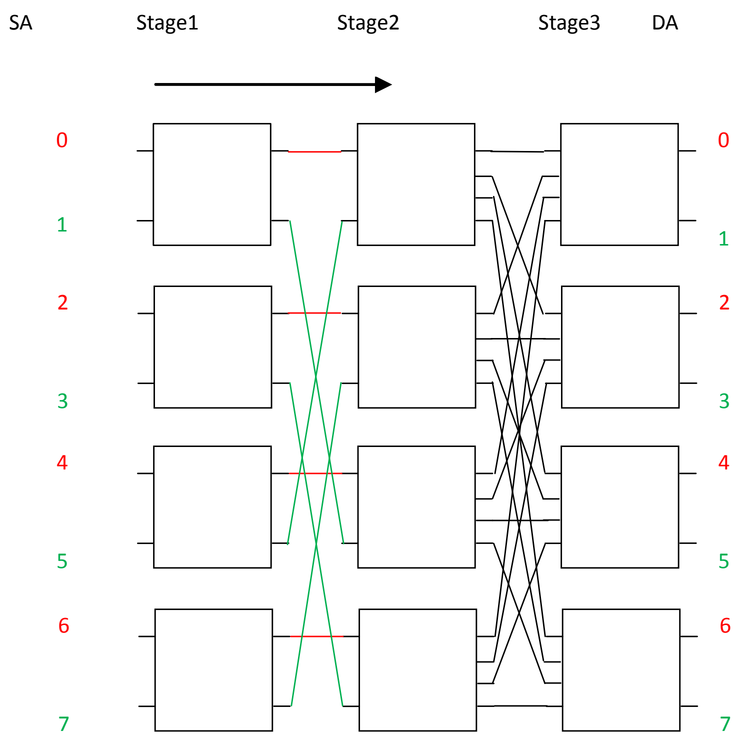

Fig. 2. 8×8 CFMON

3. The Proposed Interconnection Network

The structure of the Destination Based Modified Omega Network is based on OON [12, 13]. However, it has a different connection pattern among the stages, as compared to OON. There are $m = log_2 N$ stages in DBMON and each stage has 2^{m-1}switching elements (SEs) and therefore, DBMON has $TSE = (log_2 N \times 2^{m-1})$ SEs, where TSE is the total number of SEs. In DBMON,

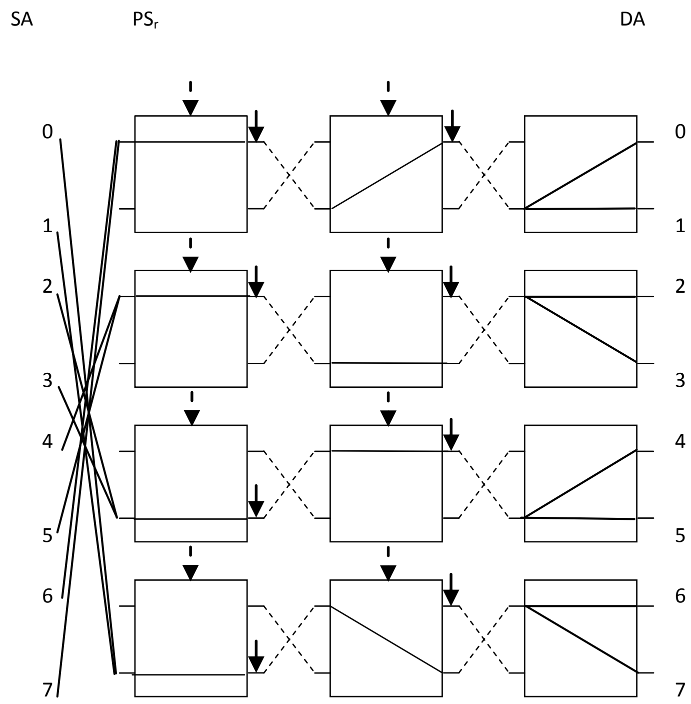

Fig. 3. 8×8 DBMON

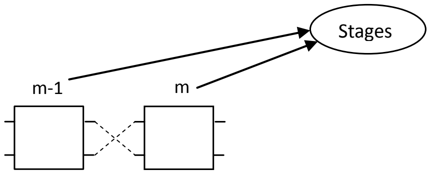

Fig. 4. Connection between the SEs in DBMON

each SE is represented as PS_r. PS_r shows the Pth SE exist in the rth stage of DBMON and $P = 1, 2, \ldots\ldots, 2^{m-1}$ and $r = 1,2, \ldots\ldots, m$. Let us suppose that we have a value $2S_3$. This tells us that we have talked about the second SE, which exists in the third stage in the network. In this network, the connectivity of the SEs of the previous stage with the SE of the imminent stage is shown in Fig. 3 and Fig. 4. In both figures the connecting links are shown by dotted lines and these links are static. Based on the connection pattern of Fig. 3, we can get a DBMON for a large network size. In Fig. 3 the dotted arrows show the switch conflicts and the solid arrows show the link conflicts in DBMON.

3.1 Internal Routing in DBMON

In DBMON, when a request arrives on a SA, then routing will be performed on the basis of its DB. If the DB is 0 then the upper input link of a SE will be active and the message will go through the upper straight way within the SE, as shown in Fig. 5(a). Sometime the DB arrives at the lower input link, at which point the message will go through the lower to upper straight way

(a) USW (b) LUS

Fig. 5. Routing of bit ‘0’ in DBMON

(a) LSW (b) ULS

Fig. 6. Routing of bit ‘1’ in DBMON

within the SE, as shown in Fig. 5(b).

If the DB is 1, then the lower input link of a SE will be active and the message will go through the lower straight way within the SE, as shown in Fig. 6 (a). Sometimes the DB arrives at the upper input link, at which point the message will go through the upper to lower straight way within the SE, as shown in Fig. 6 (b).

In Fig. 3, we have sent the messages from all of the SAs to all their DAs in a single pass and therefore the network suffers the problem of crosstalk.

Note 1: In this research work we have taken SA and DA in a decimal format in order to understand the algorithm. However, the routing of messages is performed on the basis of the binary values of their DAs.

3.2 The Destination Based Scheduling Algorithm

The DBSA algorithm is the generalized form of the ASA [11-13] and RSA [12] algorithms. This algorithm has been designed in order to obtain crosstalk-free routes in every pass. Basically, it does the scheduling of the SA and their corresponding DAs for message transmission. This scheduling is based on their DAs.

In DBSA we have considered three user-defined functions. The function INSERT-BEGINNING (head, SA, DA) takes the SA and its corresponding DA from the user side. The function EVEN-SELECTION (head) schedules the addresses for the first pass. This function only takes the addresses that have even DAs. Subsequently, the function ODD-SELECTION (head) schedules the addresses for a second pass. This function only takes the addresses that have odd DAs. Furthermore, the running time of the algorithm is calculated and is given as:

$$\mathrm{T(n)} = c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8.(n+1) + c_9.n + c_{10}.n + c_{11}.n + c_{12}.n + c_{13}.(n+1) + c_{14}.n + c_{15}.n + c_{16}.n + c_{17}.n$$

$$= c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8.n + c_8 + c_9.n + c_{10}.n + c_{11}.n + c_{12}.n + c_{13}.n + c_{13} + c_{14}.n + c_{15}.n + c_{16}.n + c_{17}.n$$

$$= (c_8 + c_9 + c_{10} + c_{11} + c_{12} + c_{13} + c_{14} + c_{15} + c_{16} + c_{17})n + (c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 + c_{13})$$

Algorithm_DBSA

Input: Network Size and DAs
Output: Scheduled DA and their SAs for first and second pass

Begin	***Cost***	***Time***
INSERT-BEGINNING (head, SA, DA)		
1. ptr.sinfo = SA	c_1	1
2. ptr.dinfo = DA	c_2	1
3. **if** LIST IS EMPTY	c_3	1
4. ptr.next = NULL	c_4	1
5. head = ptr	c_5	1
6. **else**		
7. ptr.next = head	c_6	1
8. head = ptr	c_7	1
End INSERT-BEGINNING (head, SA, DA)		
EVEN-SELECTION (head)		
9. **while** head. next!=NULL	c_8	$n+1$
10. **if** head.dinfo%2==0	c_9	n
11. head.sinfo	c_{10}	n
12. head.dinfo	c_{11}	n
End if		
13. head=head.next	c_{12}	n
End while		
End EVEN-SELECTION (head)		
ODD-SELECTION (head)		
14. **while** head!=NULL	c_{13}	$n+1$
15. **if** head.dinfo%2!=0	c_{14}	n
16. head.sinfo	c_{15}	n
17. head.dinfo	c_{16}	n
End if		
18. head=head.next	c_{17}	n
End while		
End ODD-SELECTION (head)		
End		

We can express this running time as $an+b$ where a and b are constants that depend on the statement costs c_i. Thus, it is a linear function of n. Hence the complexity of the DBSA algorithm is O (n)**.**

3.3 The Path Information Algorithm

Before passing the scheduled addresses, the path information is given to each and every SA and its DA. The Path Information Algorithm (PIA) gives the path information. The PIA shows that if the DA is even, then the function EVEN-DESTINATION (N) will be called upon to provide the path to each DA and its SA. If the DA is odd, then the function ODD-DESTINATION (N) will be called upon to provide the path to each DA and its SA. The specialty of the PIA is that it allocates a universal SE to a DA (e.g., suppose it allocates a third SE to DA 4, then in

Algorithm_PIA

Input: Network Size

Output: Destination Addresses with their Path Information

Begin	***Cost***	***Time***
EVEN-DESTINATION(N)		
1. **for** j=0 to N && k=0 to N/2	c_1	$n+1\ \&\&\frac{n}{2}+1$
2. DA[j]=j	c_2	$n\ \&\&\frac{n}{2}$
3. SE[k]=k+1	c_3	$n\ \&\&\frac{n}{2}$
End for		
End EVEN-DESTINATION(N)		
ODD-DESTINATION(N)		
1. **for** m=1 to N && n=0 to N/2	c_4	$n\ \&\&\frac{n}{2}+1$
2. DA[m]=m	c_5	$n-1\ \&\&\frac{n}{2}$
3. SE[n]=n+1	c_6	$n-1\ \&\&\frac{n}{2}$
End for		
End ODD-DESTINATION(N)		
End		

every stage the message will go through the third SE of DBMON from its SA to the DA).

Again the running time of the PIA is the sum of the running times of each step. Hence, the total running time here is:

$$
\begin{aligned}
T(n) &= c_1(n+1\ \&\&\frac{n}{2}+1) + c_2(n\ \&\&\frac{n}{2}) + c_3(n\ \&\&\frac{n}{2}) + c_4(n\ \&\&\frac{n}{2}+1) + c_5(n-1\ \&\&\frac{n}{2}) \\
&\quad + c_6(n-1\ \&\&\frac{n}{2}) \\
&= c_1(n+1) + c_2(n) + c_3(n) + c_4(n) + c_5(n-1) + c_6(n-1) \\
&= (c_1+c_2+c_3+c_4+c_5+c_6)n + (c_1-c_5-c_6)
\end{aligned}
$$

We can express this running time as $an+b$ where a and b are constants that depend on the statement costs c_i. Thus, it is a linear function of n. Hence, the complexity of the PIA is O (n)**.**

The total Complexity of the proposed algorithm is = Complexity (DBSA) + Complexity (PIA)

$$= O(n) + O(n) = 2O(n) = O(n)$$

4. PERFORMANCE EVALUATION PARAMETERS

The performance of DBMON and CFMON depends on the two parameters that are listed below.

4.1 Average NOP

Average NOP means the total number of passes that an algorithm takes to make the network crosstalk-free.

4.2 Average ET

The average ET depends on the average NOP. The average ET can be calculated as follows:

$$\mathrm{ET} = [\left(\mathrm{NOP} \times \mathrm{t} \times \left(\frac{\mathrm{N}}{\mathrm{M_S}}\right)\right] \quad (1)$$

In equation (1), t is the time and here it is assumed to be 1 ms for a single pass. M_S is the minimum network size and here it is 8. Now suppose an algorithm takes 5 NOP to make the network crosstalk-free and $N = 16$, then to get the average ET we will put the value of NOP and N in equation (1) hence, the average ET is 10 ms.

5. Performance Comparison of DBMON and CFMON

To compare the performance of DBMON and CFMON [18], we have taken an example. Let the source and destination addresses are as follows:

Source Address	Destination Address
0	**7**
1	**3**
2	**6**
3	**2**
4	**1**
5	**5**
6	**0**
7	**4**

After applying the DBSA algorithm on Example 1 above, we have obtained the following outputs:

First Pass:

Source Address	Destination Address
2	**6**
3	**2**
6	**0**
7	**4**

Second Pass:

Source Address	Destination Address
0	**7**
1	**3**
4	**1**
5	**5**

After selecting the addresses, we need to pass them through DBMON and we can then obtain the path information from the PIA, which is as follows:

First Pass:

Destination Address	Universal SE
0	**1**
2	**2**
4	**3**
6	**4**

Second Pass:

Destination Address	Universal SE
1	**1**
3	**2**
5	**3**
7	**4**

After getting the results from the DBSA and PIA we have obtained the first pass of DBMON. Fig. 7 shows that in each case the message is going through the universal SE of the network. (e.g., From SA 6 to DA 0, the message is going through the first SE in every stage and in the first pass DBMON does not has any switch and link conflicts.) In the same way we can see the second pass through DBMON.

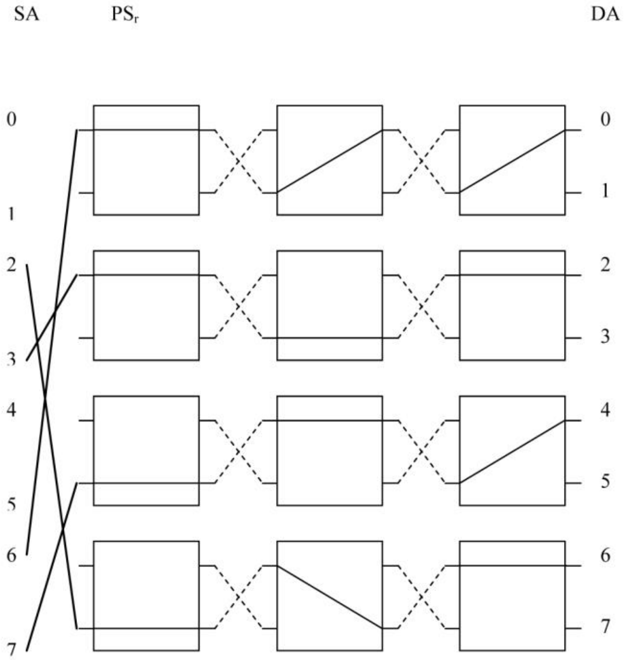

Fig. 7. First Pass of the DBSA through DMBON

By applying the CFMON algorithm on example (1), we will get the following results:

First Pass:

Source Address	Destination Address
0	**7**
2	**6**
4	**1**
6	**0**

In the first pass, CFMON will face the problem of a switch conflict in the last stage, as the DAs 7 and 6 are connected to the fourth SE of the network and DAs 1 and 0 are connected to the first SE of the network.

Second Pass:

Source Address	Destination Address
1	**3**
3	**2**
5	**5**
7	**4**

In the second pass, CFMON will face the problem of a switch conflict in the second and third SEs of the last stage as the DAs 3 and 2 are connected to the second SE of the network and DAs 5 and 4 are connected to the third SE of the network.

5.1 Comparison Based on the Average Number of Passes

Although CFMON [18] can do the message transmission in two passes, however it fails in some exceptional cases, as shown in Example 1. It also requires a maximum of four passes for every network size in order to make the network crosstalk-free. Whereas, DBMON only requires two passes in every case for every network size, as shown in Fig. 8.

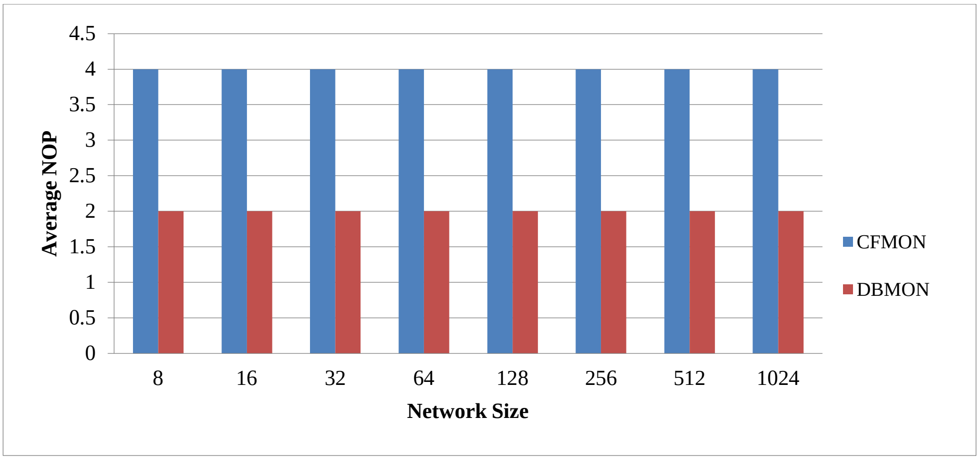

Fig. 8. Average NOP on Various Network Sizes

5.2 Comparison Based on Average Execution Time

Based on equation (1) and Fig. 8, we obtained the following equations:

$$ET_{CFMON} = \left(4 \times t \times \frac{N}{8}\right) \quad (2)$$

$$ET_{DBMON} = \left(2 \times t \times \frac{N}{8}\right) \quad (3)$$

For CFMON, the value of NOP is 4 and for DBMON the value of NOP is 2. It shows that DBMON takes less execution time than the CFMON [18]. CFMON_ET and DBMON_ET are the execution times of CFMON [18] and DBMON.

Based on Fig. 8 and Fig. 9, we can say that DBMON performs better than the CFMON in terms of NOP and average ET.

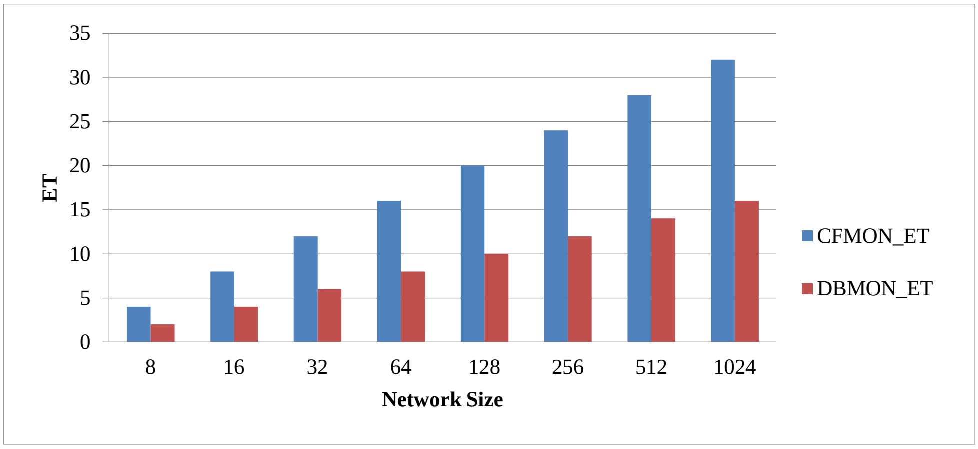

Fig. 9. The ETs of CFMON and DBMON

6. Conclusion

In OMINs, especially in OON, crosstalk is a critical challenge. In this paper, we have addressed this problem. Regarding this concern, we have proposed a new scheduling algorithm (DBSA) and we have applied this algorithm to a new network (DBMON). DBSA and PIA makes the DBMON strong and efficient for message transmission. Furthermore, results show that it is less costly and less time consuming than CFMON. In the future, our goal is to design an algorithm for OON that provide crosstalk-free communication between all SAs to all DAs with the minimum number of passes and in the minimum amount of execution time.

References

[1] Nitin, RajanVaish and Utkarsh Srivastava, "On a Deadlock and Performance Analysis of ALBR and DAR Algorithm on X-Torus Topology by Optimal Utilization of Cross Links and Minimal Lookups,"

Journal of Supercomputing, 59 (3), DOI: 10.1007/s11227-010-0524-x, 2010, pp.1252-1288.

[2] Nitin and Durg Singh Chauhan, "Stochastic Communication for Application Specific Networks-on-Chip," Journal of Supercomputing, 59 (2), DOI: 10.1007/s11227-010-0472-5, 2010, pp.779-810.

[3] Nitin, S. Garhwal and N. Srivastava, "Designing a Fault-tolerant Fully-chained Combining Switches Multi-stage Interconnection Network with Disjoint Paths," Journal of Supercomputing, 55 (3), DOI 10.1007/s11227-009-0336-z, 2009, pp.400-431.

[4] Nitin and Durg Singh Chauhan, "Comparative Analysis of Traffic Patterns on k-ary n-tree using Adaptive Algorithms based on Burton Normal Form," Journal of Supercomputing, 59 (2), DOI: 10.1007/s11227-010-0454-7, 2010, pp.569-588.

[5] Ved Prakash Bhardwaj and Nitin, "A New Fault Tolerant Routing Algorithm for Advance Irregular Augmented Shuffle Exchange Network, Proceedings of the 14th IEEE International conference on Computer Modeling and Simulation," Emmanuel College, Cambridge, UK, March 28-30, 2012, pp.505-509.

[6] Ved Prakash Bhardwaj and Nitin, "A New Fault Tolerant Routing Algorithm for Advance Irregular Alpha Multistage Interconnection Network," Advances in Intelligent and Soft Computing, Springer-Verlag, ISSN: 1867-5662, 2012, pp.979-987.

[7] Ved Prakash Bhardwaj and Nitin, "A New Fault Tolerant Routing Algorithm for IASEN-2," Proceedings of the 2nd IEEE International Conference on Advances in Computing and Communications, Kerala, INDIA, 2012, pp.199-202.

[8] C.W. Chen, "Design Schemes of Dynamic Rerouting Networks with Destination tag Routing for Tolerating Faults and Preventing Collisions," Journal of Supercomputing, 38, 2006, pp.307-326.

[9] S. Kaur, Anantdeep and D. Aggarwal, "Effect of Crosstalk on Permutation in Optical Multistage Interconnection Networks," Journal of Computing, 2 (4), 2010, pp.100-104.

[10] F. Abed and M. Othman, "Fast Method to Find Conflicts in Optical Multistage Interconnection Networks," International Journal of The Computer, The Internet and Management, 16 (1), 2008, pp.18-25.

[11] Ved Prakash Bhardwaj, Nitin and Vipin Tyagi, "An Algorithmic Approach to Minimize the Conflicts in an Optical Multistage Interconnection Network," Communications in Computer and Information Science Series, Springer-Verlag, Vol.191, ISSN: 1865:0929, 2011, pp.568-576.

[12] Ved Prakash Bhardwaj, Nitin and Vipin Tyagi, "Minimizing the Switch and Link Conflicts in an Optical Multi-stage Interconnection Network," International Journal of Computer Science Issues, 8 (4) 1, 2011, pp.206-214.

[13] Ved Prakash Bhardwaj and Nitin, "Applying Address Selection Algorithm on Nonblocking Optical Multistage Interconnection Network," Proceeding of IEEE World congress on Information and Communication Technologies, 2011, pp.694-698.

[14] T.D. Shahida, M. Othman and M.K. Abdullah, "Fast Zerox Algorithm for Routing in Optical Multistage Interconnection Networks," IIUM Engineering Journal, 11 (1), 2010, pp.28-39.

[15] M.A. Al-Shabi and M. Othman, "A New Algorithm for Routing and Scheduling in Optical Omega Network," International Journal of The Computer, the Internet and Management, 16 (1), 2008, pp.26-31.

[16] M. Abdullah, M. Othman and R. Johari, "An Efficient Approach for Message Routing in Optical Omega Network," International Journal of The Computer, the Internet and Management, 14 (1), 2006, pp.50- 60.

[17] Farzaneh Abed and Mohamed Othman, "Efficient Window Method in Optical Multistage Interconnection Networks," Proceedings of the 2007 IEEE International Conference on Telecommunication and Malaysia International Conference on Communications, Penang, Malaysia.

[18] Sonia Kaur and Neeraj Shukla, "Reducing Crosstalk in an Optical Multistage Interconnection Network," International Journal of Computer Applications (0975-8887), 46 (14), 2012, pp.8-13.

Ved Prakash Bhardwaj

Ved Prakash Bhardwaj was born on June 08, 1986, in Jaipur, India. He received the B. Engg. in Computer Sc. & Engineering from Rajasthan Institute of Engineering & Technology (RIET), Bhankrota, Jaipur in 2008. In 2010, he received the M.Tech. in Computer Science and Engineering from Jaypee University of Information Technology (JUIT),Waknaghat, Solan-173234, Himachal Pradesh, INDIA. Currently, he is pursuing Ph.D. in computer science from Jaypee University of Information Technology. His research domain is interconnection networks.

Dr. Nitin

Dr. Nitin is Ex First Tier Bank Professor, University of Nebraska at Omaha, NE, USA. His permanent affiliation is with Jaypee University of Information Technology (JUIT), Waknaghat, Solan-173234, Himachal Pradesh, INDIA as a Associate Professor in the Department of Computer Science & Engineering and Information & Communication Technology. He joined Jaypee University of Information Technology in July 2003. He was born on October 06, 1978, in New Delhi, INDIA.

In July 2001, he received the B.Engg. in (CSE [Hons.]) from B.R. Ambedkar University, Agra and M.Engg. in (SE) from Thapar Institute of Engineering and Technology, Patiala, INDIA in March 2003. In September 2008, he received his Ph.D. in (CSE) from JUIT, INDIA. He has completed his Ph.D. course work from University of Florida, Gainesville, FL, USA. Finally, he received the D.Sc. in (CSE) from Uttarakhand Technical University, Dehradun, INDIA, in May 2013. The half of the D.Sc. work is executed at University of Nebraska at Omaha, USA.

He is a IBM certified engineer and Senior Member-IEEE and IACSIT, Life Member of IAENG, and Member-SIAM and ACIS. He has 125 research papers in peer reviewed International Journals and Transactions, Book Chapters, Symposium, Conferences and Position. His research interest includes Social Networks especially Computer Mediated Communications and Flaming, Interconnection Networks and Architecture, Fault-tolerance and Reliability, Networks-on-Chip, Systems-on-Chip, and Networks-in-Packages, Wireless Sensor Networks. Currently he is working on Parallel Simulation tools, BigSim using Charm , NS-2 using TCL. He is the Co-founder of High-end Parallel Computing and Advanced Computer Architecture Lab at JUIT. He is Associate Editor of Journal of Parallel, Emergent and Distributed Systems, Taylor and Francis, UK. He is referee for the Journal of Parallel and Distributed Computing, Computer Communications, Computers and Electrical Engineering, Mathematical and Computer Modelling, Elsevier Sciences. WSEAS Transactions, The Journal of Supercomputing, Springer and International Journal of System Science, Taylor and Francis.