We are IntechOpen,
the world's leading publisher of Open Access books
Built by scientists, for scientists

5,900
Open access books available

145,000
International authors and editors

180M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

The Development of Crosstalk-Free Scheduling Algorithms for Routing in Optical Multistage Interconnection Networks

Mohamed Othman, and Tg Dian Shahida Raja Mohd Auzar
Department of Communication Technology and Network
Universiti Putra Malaysia
Malaysia

1. Introduction

Advances in electro-optic technologies have made optical communication a promising networking alternative to meet the ever increasing demands of high-performance computing communication applications for high channel bandwidth, low communication latency and parallel processing as well. 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. A major problem in OMIN is optical crosstalk. It is caused by coupling two signals within a Switching Element (SE). Crosstalk problem in a switch is the most prominent factor, which reduces the signal-to-noise ratio and restricts the size of a network.

Various methods to decrease the undesirable effect of crosstalk have been proposed, that apply the concept of dilation in either the space, time or wavelength domains. With the space domain approach, additional SE(s) and links are used to certify that at most only one input and one output of every SE will be active at any given time. With the time domain approach, two connections will be activated at different time slots if they share the same SE in any stage of the network. The last approach, the wavelength domain, different wavelengths are used for routing active connections by ensuring two wavelengths entering an SE to be far apart by routing or using wavelength converters. Whenever the limitation of the network size is reached, the time domain method may be used as a feasible way to trade the maximal bandwidth available to each particular input and output pair for enhanced connectivity. Again, it is useful when future technology let the transmission rate to expand faster than the network size or when the cost of expanding the bandwidth of each connection becomes as "cheap" as the cost of building a network of twice its original size.

The chapter covers the development of crosstalk-free scheduling algorithms for routing in an OMIN. The interest is on how to efficiently schedule messages using the time domain approach in order to avoid crosstalk. In the time domain approach, messages to be sent to the network are distributed into several groups for routing at different time slots. There are

two phases involved in the algorithm development; permutation decomposition phase to find messages with conflicting path (shares an SE) in the network and mapping these conflicts into some array before it can be selected for scheduling and scheduling algorithm phase to schedule messages according to some order for routing in the OMIN.

We begin this chapter with a brief introduction of MIN and OMIN, particularly the Optical Omega Network (OON) topology that will be used to represent OMIN, and the problem associated with routing permutations in OON in general and crosstalk as the main prominent factor that effect the performance of OON and OMIN at large. Next, we will also discuss commonly used approaches to solve crosstalk in OON including the strength and weaknesses of each approach. At the end of this chapter, we present recent research in crosstalk-free scheduling and their development based on the time domain approach.

2. Optical Multistage Interconnection Network

Multistage Interconnection Network (MIN) are a class of dynamic interconnection network that connects input devices to output devices through a number of switch stages, each stage consists of a set of SEs arranged in cascaded order, where each switch is a crossbar network (Feng, 1981). Frequently proposed as interconnection schemes in multiprocessor systems or in high bandwidth network switches, MIN has assumed importance in recent times, because of their cost-effectiveness. While crossbar networks have the advantage of establishing connections between every input port to any free output port, it requires N^2 switches to construct the network where N is the network size. MIN requires only $N(\log_2 N)/2$ switches for the same N (Dally, *et al.*, 2004 and Abdullah, *et al.*, 2006).

Depending on the interconnection scheme employed between two adjacent stages and the number of stages, various MINs have been proposed. MIN can be one-sided, where both inputs and outputs are on the same side, or two-sided, which usually have an input side and output side on opposite sides of the network. The two-sided MIN are generally divided into three classes, called blocking, rearrangeable and nonblocking network. In blocking networks, simultaneous connections of more than one input and output pair may result in link contention. Examples of blocking MIN are Baseline, Cube and Omega networks. Rearrangeable networks are a class of MIN where all possible connections can be performed between inputs and outputs by rearranging existing connections so that a connection path for a new input/output pair can always be established. The Benes network is an example of rearrangeable networks. The non-blocking class of MIN is a network that can handle all possible connections without blocking. The Clos network is an example of this class of MIN.

As optical technology advances, there are considerable interests in using optical technology for implementing interconnection networks and switches. As far as speed is concerned, with the introduction of optical switching, it is most feasible to apply MIN in the architecture for building electro-optical switches with a capacity at the rate of terabits per second. An OMIN can be implemented with either free-space optics or guided wave technology (Yang, *et al.*, 2000). It involve the switching of optical signals, rather than electronic signals as in conventional electronic systems. A hybrid OMIN can be built from electro-optic SEs such as common Lithium Niobate (LiNbO$_3$) directional coupler with two inputs and two outputs (Yang, *et al.*, 2000 and Lu, *et al.*, 2003). OMIN has been an attractive solution that offers a combination of high bandwidth, low error probability, and large transmission capacity in the design of high-speed communication networks and switches (Lu, *et al.*, 2003).

3. Optical Omega Network

An Optical Omega Network (OON) topology has altogether N inputs, N outputs and n stages where $n=\log_2 N$. Each stage has $N/2$ SEs with each SE has two inputs and two outputs connected in a certain pattern (Wu, *et al.*, 1980; Feng, 1981 and Dally, *et al.*, 2004). The inter-stage connection pattern in an Omega network is of shuffle-exchange connection pattern (Wu, *et al.*, 1981). To connect the source address to the destination address, the address is shifted one bit to the left circularly in each connection such as source to the first stage, one stage to the next stage. For instance, to connect between each stage in an 8 x 8 optical Omega network, each connection is shuffle-exchanged as shown in Figure 1(a).
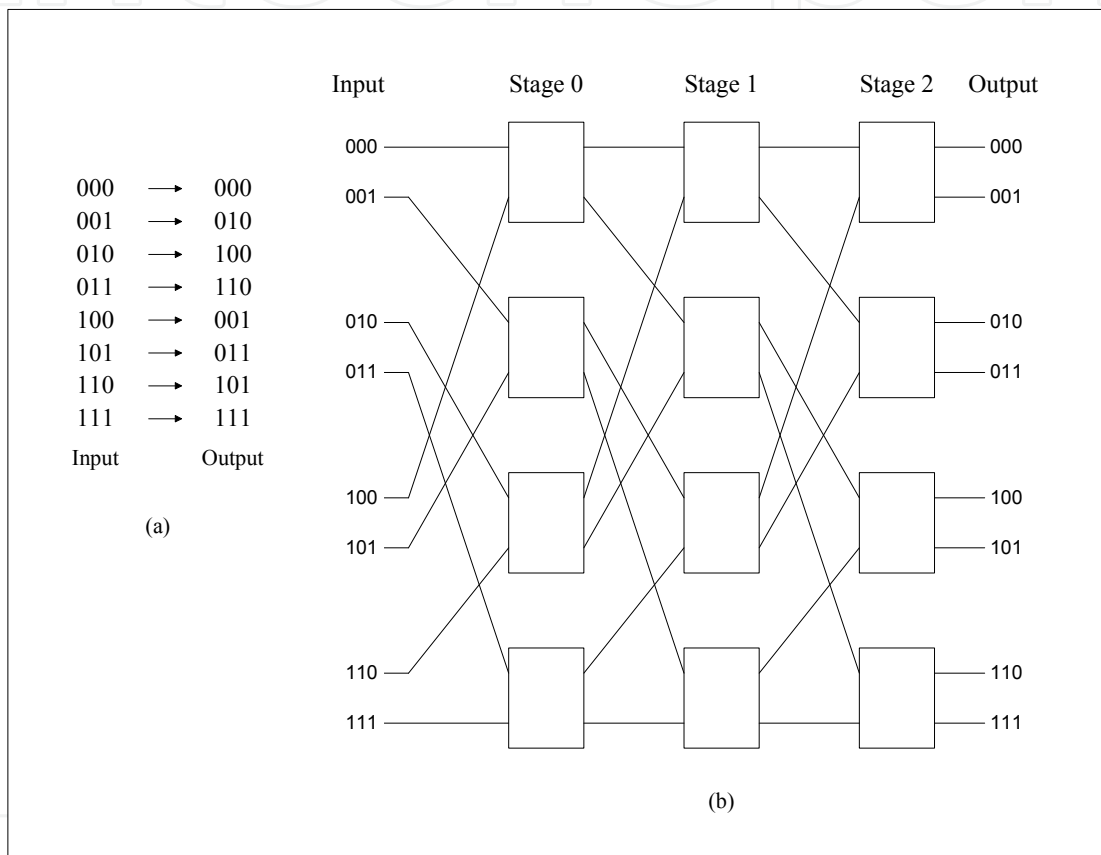

Fig. 1. (a) Shuffle-Exchange Inter-Stage Connection Pattern, and (b) An 8 x 8 Optical Omega Network

The shuffle-exchange connections have to be considered when scheduling a permutation for routing in the OON. The inter-stage connection pattern determines the routing mechanism in a network. It also limits the number of messages that can be routed simultaneously in a single time slot or pass, since no two signals are allowed to share an SE at any given time or crosstalk will occur. Figure 1(b) illustrates the general layout of the Omega network topology. OON is topologically equivalent to many other topologies such as the Baseline, Butterfly and Cube networks (Wu, *et al.*, 1980 and Feng, 1981). Since many other topologies are equivalent to the Omega network topology, performance results obtained for the Omega network are also applicable to other OMIN topologies (Abdullah, 2005).

Suppose an n-bit binary numbers from 0 to $N-1$ (where $n=\log_2 N$ and N is the network size) is used to label the addresses of N input or output ports from top to bottom of the OON, the shuffle-exchange interconnection connects output port $s_0 s_1 s_2 \ldots s_{n-1}$ from stage i to the input port $s_1 s_2 \ldots s_{n-1} s_0$ of stage $i + 1$, $0 \le i < n - 1$. Every stage of switches in the OON is preceded by the shuffle-exchange interconnection including the N source inputs connected to the switches of the first stage. The switching connections in each SE can be of either straight or cross connection.

To route a message in an OON, the destination tag which is binary equivalent of the destination address, $(d_{n-1} d_{n-2} \ldots d_1 d_0)$ is used. The ith bit d_i is used to control the routing at the ith stage counted from the right with $0 \le i \le n - 1$. If $d_i = 0$, the input is connected to the upper output. Otherwise, if $d_i = 1$, it is connected to the lower output. In other words, message routing can be achieved simply by relaying messages to either the upper switch output link or the lower output link of the SEs according to the destination address. This unique characteristic of the OON are often referred to as self-routing (Chau, *et al.*, 2005 and Chao, *et al.*, 2007).

4. Optical Crosstalk in Optical Omega Network

Although electronic MIN and OMIN have many similarities, there are some fundamental differences between them. An SE can be connected using two types of logic states, namely the straight or cross connection as shown in Figure 2(a). Depending on the amount of voltage at the junction of the two waveguides, optical signals carried on either of the two inputs can be coupled to either of the two outputs. Figure 2(b) shows an illustration of the optical crosstalk occurrence within an SE connected with straight logic state. In the event of optical crosstalk occurrence, a small fraction of the input signal power may be detected at another output disregard of the actual signal injected to the appropriate output port. Consequently, the input signal will be distorted at the output due to loss and crosstalk accumulated along the connection path.
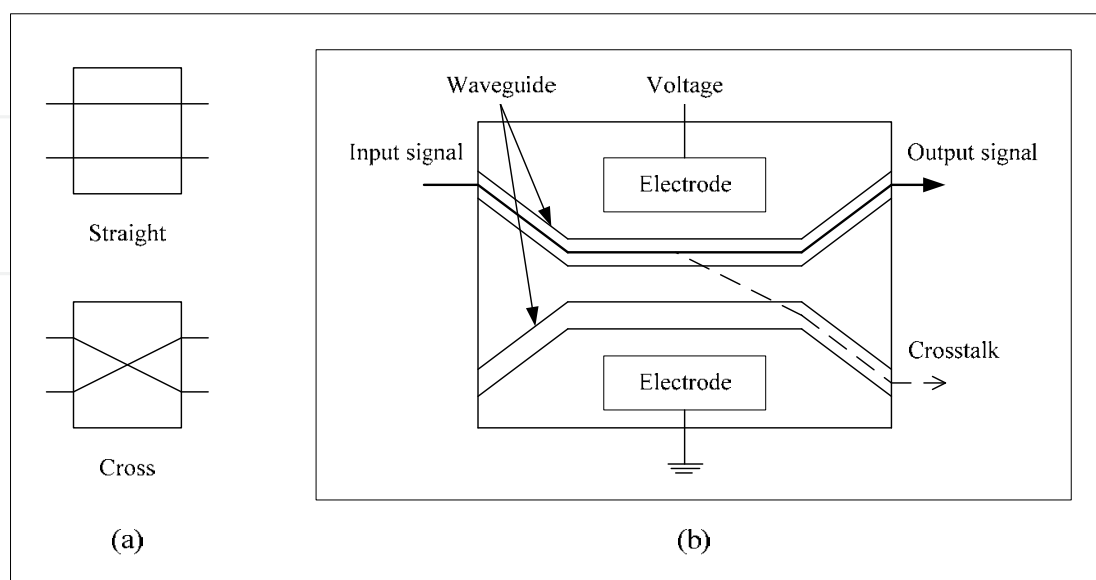

Fig. 2. (a) Straight or Cross Logic State of a 2x2 SE, and (b) Optical Crosstalk Effect in an Electro-Optic SE

Because routing in OON make use of both SE configuration shown in Figure 2(a), optical crosstalk has been the major drawback in achieving the most of network performance when routing permutations simultaneously. Therefore, it is not possible to route more than one message simultaneously, without optical crosstalk, over an SE in OON. Reducing the effect of optical crosstalk has been a challenging issue considering trade-offs between performance and hardware and software complexity. To solve optical crosstalk, many scheduling algorithms have been proposed for routing in OMIN based on a solution called the time domain approach, which divides the N optical inputs into several groups such that crosstalk-free connections can be established. In this chapter, we propose a solution that can further optimize and improve the performance of message scheduling for routing in OON using the time domain approach.

5. Time Domain Approach

In order to avoid crosstalk in OONs, several approaches based on network dilation have been proposed. The three approaches include the the space domain (Qiao, 1996), time domain (Qiao, et al., 1994) and wavelength domain (Lu, et al., 2004) dilation.

Space domain approach duplicates and combines a MIN to avoid crosstalk within individual SE. Using this approach, an N x N network is dilated into a network that is essentially equivalent to a $2N$ x $2N$ network, but only half of the input and output ports used for routing. Based on this approach, a dilated Benes network has been proposed (Padmanabhan, et al., 1987), where up to N connections can be established without sharing any SE. However, it uses more than double of the number of switches required for the same connectivity.

The wavelength domain (Vaez, et al., 1998) implements an approach where the crosstalk between two signals passing through the same SE is suppressed. It is done by ensuring two wavelengths to be far apart by routing (Sharony, et al., 1993) or by using wavelength converters (Qin, et al., 2001). Thus, limits the efficiency of bandwidth utilization and/or increases cost and complexity (Lu, et al., 2003 and Yang, et al., 2004).

The time domain approach avoids optical crosstalk via network partitioning across the time dimension. The approach solves the crosstalk problem by allowing only one input and output link to be active at a time within each SE. A set of permutation connection is partitioned into several scheduling groups called semi-permutations in such a way that the entries within each group are crosstalk-free (Yang, et al., 2000 and Lu, et al., 2003). Each group is routed to its corresponding destination independent of the other groups in a different time slot. The main advantage of the time domain approach is that it does not involve additional cost of having more SEs as well as the cost for wavelength conversion as does the space and wavelength domain approaches.

5.1 Time Domain Approach Framework
Because routing messages simultaneously across the OON causes crosstalk, it is important to make sure a permutation is decomposed and scheduled in crosstalk-free order for routing messages. The general framework of the time domain approach consists of two phases including permutation decomposition in the first stage and message scheduling in the second as illustrated in Figure 3.

In permutation decomposition phase, the primary goal is to identify conflicts among all messages in the network. Permutation decomposition involves source and destination address generations, building the conflict matrix in order to discover message conflict for the permutation. In the message scheduling phase, based on the array of conflict generated, the messages are sorted and selected for scheduling into crosstalk-free groups called passes for routing to the intended destination, one group per time slot. Typically, messages are selected for scheduling according to some order unique to a particular scheduling algorithm. The output of each scheduling algorithm is the execution time for scheduling permutations and number of passes to route a permutation. Both the execution time and the number of passes are used to measure the performance of time domain scheduling algorithms.
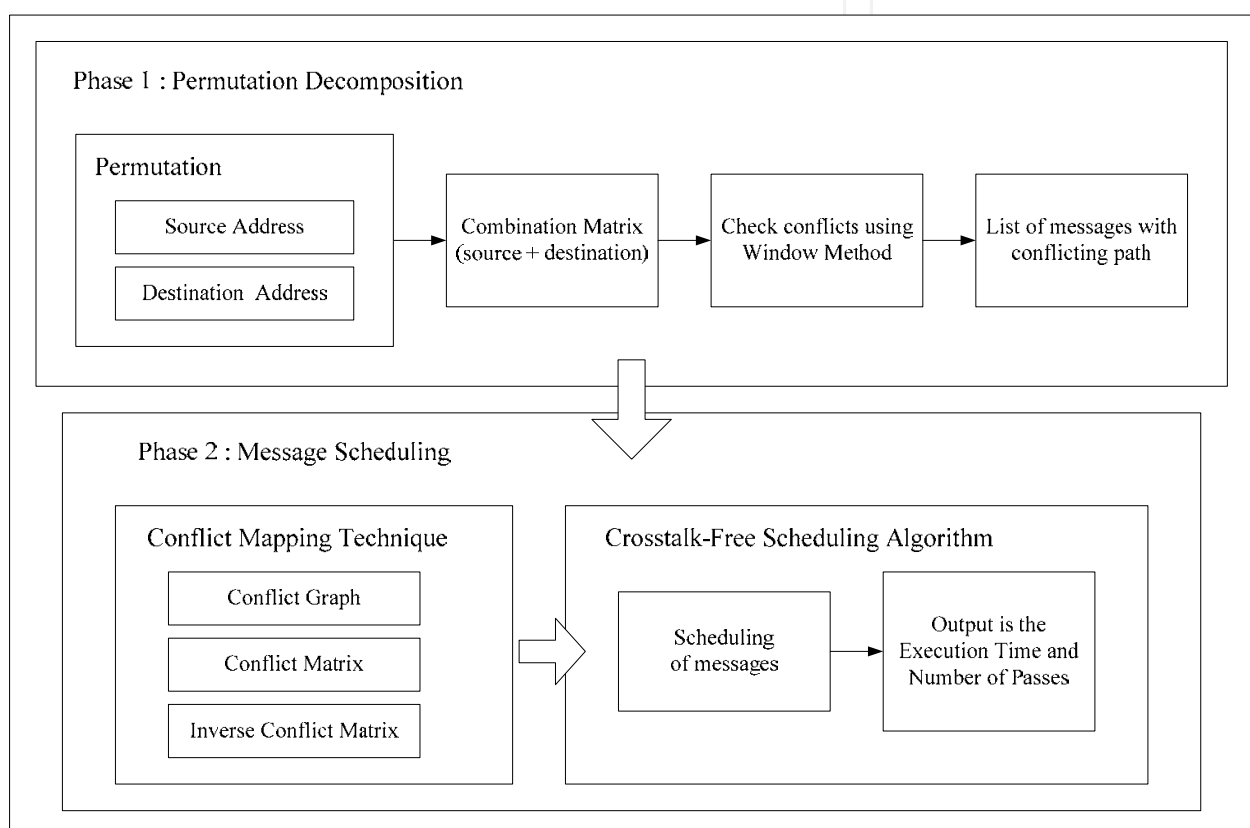
Fig. 3. Time Domain Approach Framework

5.2 Permutation Generation

Before a permutation can be divided into its crosstalk-free subsets, the source and destination addresses of the permutation are randomly generated. A permutation refers to a one-to-one mapping from a source node to a destination node in the OON. The network size, N is defined as a base-2 integer, 2^n where $n=log_2N$, ranging from the smallest size 4 to the largest size 1024 that represents the number of source nodes and destination nodes of the network. In each permutation generation, each source is connected to a random destination via unique one-to-one mapping. For example, when $N = 16$ it specifies that there are 16 unique source nodes mapped to 16 unique random destination nodes. This arbitrary source and destination address of the permutation generated is then combined in its binary form to build a combination matrix.

5.3 The Combination Matrix

To build the combination matrix, each source and destination address pair of a permutation will be represented separately in their n-digit binary structure, where $n = \log_2 N$. Then, both source and destination addresses from the pair are combined; with the source address put on the left followed by the destination address on the right. For example, combining the address of source '0' and destination '5' will result in binary 000101 with the first three digits from the left represent the source address while the last three digits represent the destination address. The combination matrix is useful when identifying conflicts in OON.

5.4 Conflict Discovery

5.4.1 Window Method

Based on the combination matrix, conflict patterns are checked using some pattern-checking method. Window Method (WM) is one example of a pattern-checking method where the combination matrix is divided into windows of the same size; and if any two messages have the same bit pattern between them in any of the windows, then it implies conflict between the message pair. Thus, the two messages must not be scheduled in the same group. In WM, an optical window of size $m - 1$ where $m = \log_2 N$ and N is the size of the optical network is applied to the columns of the combination matrix, from left to right excluding the first and the last columns. For each optical window, the bit pattern for each message is compared for similarity with the bit pattern of the rest of the other $N - 1$ message(s) sequentially starting from message 0 to $N - 1$. If the bit pattern is the same, it will be mapped into the array of conflict pattern.

5.4.2 Improved Window Method

Sequentially comparing bit patterns among all messages in each optical window was found to be time consuming especially when the network size, N is large and the number of optical window increases. To reduce the execution time contributed by the WM, the Improved WM (IWM) was proposed that eliminates checking for conflicts in the first optical window (Abdullah, 2005). This is because the first optical window has the same conflict pattern where the first $N/2$ inputs in sequence uses the same SEs as the second half of the other $N/2$ inputs. Therefore, inputs 0 to ($N/2 - 1$) will have conflict with inputs $N/2$ to ($N - 1$), which is always true for any size of network, N. The execution time is reduced approximately by $1/S$ as compared to the standard WM where S is the number of stages (Abdullah, 2005; Abdullah, *et al.*, 2005). In addition, the parallel IWM was developed on shared memory multiprocessors and the results shown drastic improvement in execution time (Othman, *et al.*, 2008).

5.4.3 Bitwise Window Method

Based on comparative analysis performed in (Abed, *et al.*, 2007), it was concluded that the time spent for identifying conflicts is very high compared to routing the messages. Table 1 shows the execution time of WM compared to the time executed for scheduling and routing.

Network Size	Routing + WM	WM	Routing
8	0.032	0.031	0.001
16	0.078	0.063	0.015
32	0.219	0.204	0.015
64	1.031	1.000	0.031
128	4.797	4.656	0.141
256	25.329	24.187	1.142
512	110.750	108.906	1.844
1024	519.922	499.046	20.876

Table 1. WM Execution Time (*ms*) (Abed, *et al.*, 2007)

Based on the analysis, (Abed, 2007) then proposed the Bitwise Window Method (BWM) that significantly reduces the execution time of the WM. In the new BWM, each (n - 1)-bit binary optical window of the standard WM where n=$\log_2 N$ and N is the network size, be transformed into its equivalent decimal representation using bitwise operations. As a result, the number of columns used to compare each message for similar bit pattern is reduced to n, instead of ($n^2 – n$) for an N x N Omega network. Figure 4 illustrates the transformation steps for each optical window in BWM implementation.
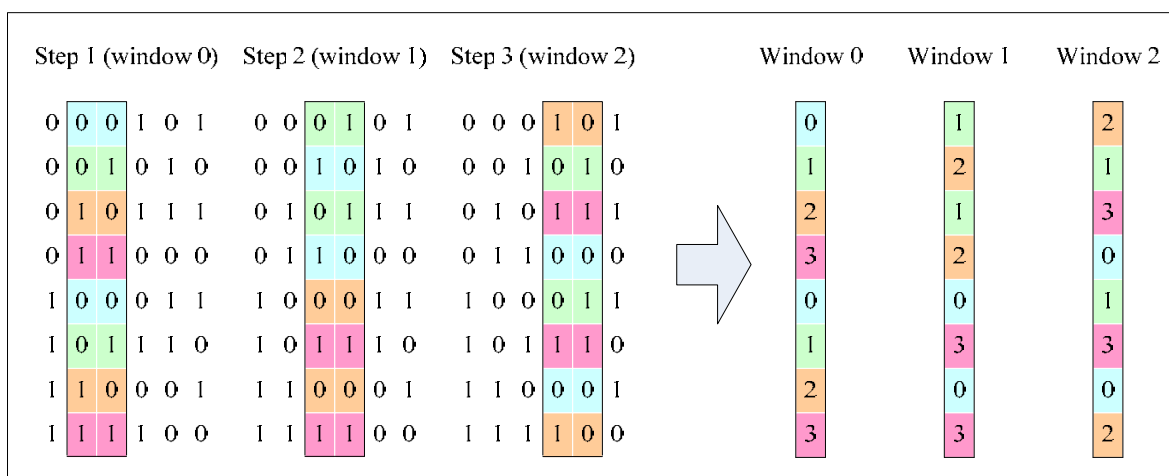

Fig. 4. Optical Window Transformation in BWM

5.5 Conflict Mapping

5.5.1 Conflict Graph

The conflict graph is one of the foremost technique proposed to map conflicts discovered using WM. By definition, the conflict graph of an N-permutation π (where N is the network size) is the graph $G(V, E)$ where V is a set of vertices $\{v_0 v_1 v_2 \ldots v_{N–1}\}$ and E is a set of edges $\{(v_0, v_1),\ldots,(v_i, v_j),\ldots,(v_{N-2}, v_{N-1})\}$ (Shen, *et al.*, 2001). Each vertex, $V = \{v_0 v_1 \ldots v_{N–1}\}$ in the conflict graph represents a source node's address i.e. v_0 for source 000, v_1 for source 001 and so on for all nodes in the network. In the conflict graph, any two vertices v_i and v_j are connected by an edge, E to indicate conflict, if and only if they share a common SE at certain stage of the network.

5.5.2 Conflict Matrix

Another conflict-mapping technique that can be used to map conflict pattern identified using WM is called the conflict matrix, proposed by (Al-Shabi, 2005). The conflict matrix is defined as a square matrix, M with matrix size of $N \times N$ where N is the network size. Similar to the conflict graph, the conflict matrix can be generated based on the results from the WM, IWM or BWM. However, if there is an edge from some vertex i to some vertex j in the conflict graph, then the element $M_{i,j}$ in the conflict matrix is set to a value 1 to indicate conflict between the intersected messages, otherwise it is set to a value 0.

The conflict matrix is illustrated in Figure 5. Since the message 000 has conflict with messages 010, 100 and 111, elements $M_{000,010}$, $M_{000,100}$ and $M_{000,111}$ are set to the value 1 to indicate conflict in the conflict matrix. The rest of the intersections for message 000 i.e. the intersections between message 000 and messages 001, 011, 101 and 110 are set to 0 value, which means that these messages will not cause crosstalk with the message 000 during routing in the network.

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

Fig. 5. The Conflict Matrix

6. Message Scheduling Algorithms

Among the algorithms developed by previous researchers to perform scheduling of the messages into crosstalk-free groups for routing in OON include the standard four Heuristic algorithms; Sequential Increasing, Sequential Decreasing, Degree Ascending and Degree Descending algorithm, Simulated Annealing (SA) algorithm, Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm, Remove Last Pass (RLP) algorithm, Zero algorithm, Improved Zero (IZero) algorithm and Bitwise-Based algorithm. To evaluate the performance of the time domain scheduling algorithm, researchers have used two main parameters; the total execution time for scheduling permutations and the total number of passes to route a permutation (Katangur, *et al.*, 2002; Katangur, *et al.*, 2004; Chau, *et al.*, 2005; Abdullah, *et al.*, 2006 and Al-Shabi, *et al.*, 2008). Based on these parameters, we listed a summary of the performance analysis of the algorithms below:

- Among the four Heuristic algorithms, the Degree Descending algorithm gives the best result while the Degree Ascending algorithm gives the worst result in terms of

the number of passes obtained for a permutation (Katangur, *et al.,* 2002; Katangur, *et al.,* 2004 and Al-Shabi, *et al.,* 2008). In terms of the execution time, the Sequential Increase and Sequential Decrease algorithms gives best result compared to Degree Ascend and Degree Descend algorithms (Al-Shabi, *et al.,* 2008, Othman, *et al.,* 2008). It can be summarised that scheduling the messages based on their order of degrees achieved best results in the number of passes but consumes more time to calculate the degree for each message in the network. On the other hand, both Sequential Increase and Sequential Decrease algorithms achieve best in execution time but higher in the number of crosstalk-free passes obtained to route a given permutation.

- The SA algorithm is one of the time domain algorithms that perform best in terms of the number of passes for message routing without crosstalk. It is the second best among all time domain algorithms where the number of passes obtained by the SA algorithm result closely to that of the GA (Katangur, *et al.,* 2002) but slightly higher than the RLP algorithm (Chau, *et al.,* 2005). However, as much as the reduction in the execution time compared to the GA as demonstrated in (Katangur, *et al.,* 2002), the SA algorithm is also considered as one of the algorithms that consume longer time to compute a solution (Al-Shabi, *et al.,* 2008).

- Due to the complex procedures involved in the algorithm, the GA takes a very long time to calculate the number of passes especially for large network sizes and therefore considered less appropriate for message routing particularly in OON by previous researchers (Katangur, *et al.,* 2002; Chau, *et al.,* 2005 and Al-Shabi, *et al.,* 2008).

- The ACO algorithm successfully reduces the number of passes when limited crosstalk is allowed in the network. Unfortunately, when zero crosstalk is concerned, the number of passes is higher than the rest of the other algorithms (Katangur, *et al.,* 2004).

- RLP algorithm gives the best result when the number of passes is considered (Chau, *et al.,* 2005). However, the algorithm consumes longer execution time than other time domain algorithms. Apart from the algorithm's dependency to other algorithm to obtain the initial solution, the RLP algorithm also involves complex procedures when making scheduling decisions (Shahida, *et al.,* 2008d, Shahida, 2009).

- Based on the experimental results shown in (Al-Shabi, 2005; Al-Shabi, *et al.,* 2005 and Al-Shabi, *et al.,* 2008), Zero algorithms perform best in terms of the execution time for scheduling the permutations and result very closely to the Degree Descending algorithm in terms of the number of passes generated to route a permutation. However, it was found in (Abed, 2007) that crosstalk may still occur between messages scheduled in the same pass of a particular time slot using the original Zero algorithm.

- Improved the weaknesses found in the original Zero algorithm, IZero algorithm performed slightly higher in terms of its execution time for scheduling permutations compared to the original algorithm while maintaining the same result in the total number of passes to route a permutation (Abed, *et al.,* 2008; Shahida, *et al.,* 2008b; Shahida, *et al.,* 2008c).

- All Bitwise-Based algorithms have shown to successfully reduce the execution time of the original algorithm (Abed, 2007; Abed, *et al.,* 2007 and Abed, *et al.,* 2008), except that the number of passes obtained by the new algorithm is the same as before it is implemented using the Bitwise approach. Among all Zero-based algorithms, none of the previous researchers improved the performance of the algorithm in terms of the number of passes to route a permutation. Furthermore, since the Bitwise-Based algorithms used the same framework and procedures from the original algorithm, the BIZero algorithm is also not crosstalk-free.

7. Fast Zero Algorithm

Fast Zero (FastZ) algorithm (Shahida, *et al.,* 2008a; Shahida, *et al.,* 2008b and Shahida, *et al.,* 2008c), is among the latest time domain scheduling algorithm proposed to optimally minimize the execution time of Zero-based algorithms. Although various developments have been made towards the original Zero algorithm including the IZero and Bitwise IZero algorithms to refine the algorithm in different aspects, none of these developments affects the steps involved in the algorithm itself to improve the execution time of the algorithm. Through analysis, Zero-based algorithms involve a lot of iterative procedures in the algorithm's function for the summation of rows or columns, finding the intersections to refine the conflict matrix, multiple conflict-checking to ensure no conflicts when adding successful intersections to a group, reducing the matrix and calculating new summation for the reduced matrix.

FastZ algorithm consist of three algorithms namely Fast ZeroX (FastZ_X), Fast ZeroY (FastZ_Y) and Fast ZeroXY (FastZ_XY) algorithms. FastZ algorithm is also combined with the RLP algorithm to reduce the total number of passes obtained for a particular permutation and will be described in section 7.4.

7.1 Permutation Decomposition

Based on the time domain approach, scheduling depends very much on the pattern of conflicts among the messages. Conflict-mapping technique i.e. the conflict graph provides an easy access to refer conflicts between messages in the network before scheduling the messages. For instance, to determine if a message is routable in a particular group is as simple as referring to the conflict graph and check whether there is an edge connecting the message to the messages that is already scheduled in the same group. An efficient conflict-mapping technique affects the total execution time of an algorithm. Therefore, we proposed another technique called symmetric Conflict Matrix (sCM) to map conflicts in the network discovered using BWM. The new sCM is implemented in FastZ algorithm replacing the conflict matrix.

7.2 Symmetric Conflict Matrix

The sCM is defined as a square matrix, $S_{i,j}$ with matrix size of N x N where N is the network size. The sCM has the same data structure as the conflict matrix to indicate conflict or no conflict between a message pair, only that the entries in the sCM are symmetric to that of the conflict matrix with respect to the matrix's main diagonal of all 0's value. In other words, if an entry $M_{i,j} = 1$ (where $i = 0$ and $j = 4$) in the conflict matrix, see Figure 5, then entries $S_{i,j} = S_{j,i} = 1$ in the sCM as shown in Figure 6.

A great advantage using sCM compared to the conflict matrix is that the sCM provides a complete mapping of all possible conflicts in the network similar to the conflict graph. Scheduling algorithm can be simplified and more straightforward by comparing the intersection value of intersected messages to determine routability thus eliminates time-consuming procedures associated with multiple summation of the conflict matrix, finding intersections, and reducing the conflict matrix in Zero-based algorithms.

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

Fig. 6. The Symmetric Conflict Matrix, sCM

7.3 Message Scheduling

The basis of Zero-based algorithms lies in the Unique Case and Refine functions executed after obtaining the row or column summations of the conflict matrix. However, these procedures are time-consuming, thus contribute to longer execution time to schedule messages for routing in the network. Using sCM, scheduling of messages is more straightforward simply by checking through the intersections between the entries in the sCM, without prior row or column summation of the conflict matrix.

For instance, to check if message 100 is routable in a group already scheduled with messages 000 and 101 based on the sCM in Figure 6, it is sufficient to make sure that $M_{100,000} = M_{100,101} = 0$. In this case, message 100 is not routable in the group since $M_{100,000} = 1$. Therefore, both the Unique Case and Refine functions are no longer necessary and can be removed in FastZ algorithm. Furthermore, the sCM need not be reduced to smaller matrix size since the sCM provides all the information needed for scheduling. Figure 7 shows the general flowchart of the FastZ algorithm.
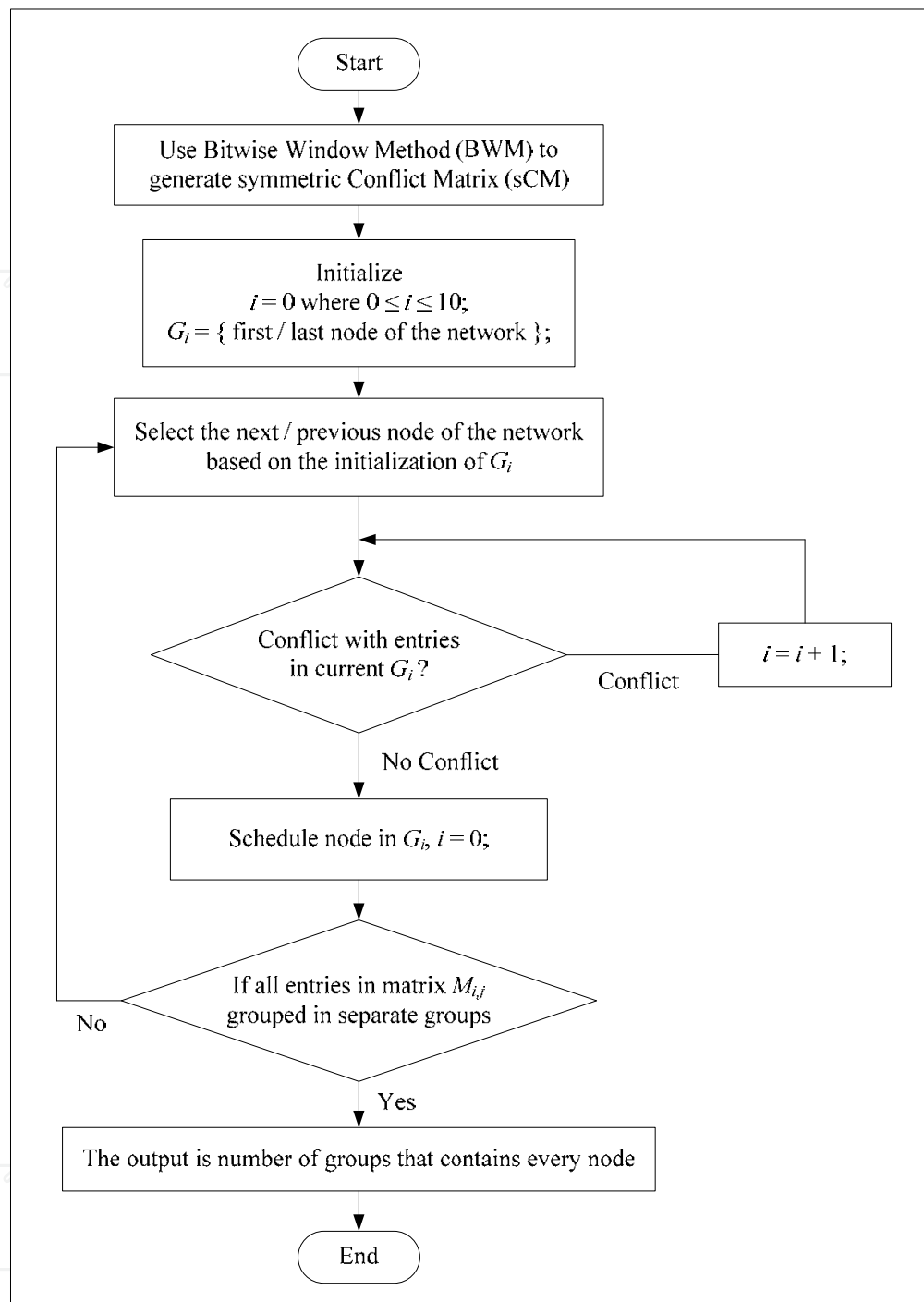
Fig. 7. General FastZ Algorithm Flowchart

FastZ algorithm consists of three algorithms; FastZ_X, FastZ_Y and FastZ_XY algorithms. The difference between these algorithms is in the selection of which message to be added to the first scheduling group during group initialization. For initialization, FastZ_X algorithm selects the first message in the network to the first group. The rest of the messages are selected for scheduling ascendingly, starting from the second message based on the message's source address. On the contrary, FastZ_Y algorithm chooses the last message, N in the network for initialization. FastZ_Y algorithm schedules the other messages

descendingly based on their addresses until all messages are scheduled into crosstalk-free groups. To schedule a permutation in the FastZ_XY algorithm, messages are scheduled using both FastZ_X and FastZ_Y algorithms sequentially. After both results are obtained, FastZ_XY algorithm compares both results and chooses whichever result that has the lowest total number of passes as its final result.

Apart from the algorithm's simplicity in routability and scheduling, FastZ algorithm is also crosstalk-free and therefore solved the weaknesses in Zero-based algorithms. Although IZero algorithm solved optical crosstalk in Zero algorithm, our analysis found that for some permutation involving the IZero algorithm's Unique Case function, crosstalk may still occur between messages scheduled in the same group. Therefore, the IZero algorithms and its predecessors are not crosstalk-free and only applicable for message scheduling in OMIN architecture where limited crosstalk is allowed.

7.4 Fast Zero Algorithm with RLP

We introduced the latest development of Zero-based algorithms; the FastZ algorithm with RLP or FastRLP in short. The algorithm is designed by integrating FastZ algorithm with the RLP algorithm proposed by (Xiao, 2004), with attempt to minimize the total number of passes for routing a given permutation. Based on analysis performed in (Chau, *et al.*, 2005), the RLP algorithm has shown to successfully schedule a permutation with less number of passes, than the Maximal Conflict Number (MCN) required for the permutation.

In FastRLP algorithm, messages are scheduled using FastZ algorithm to obtain initial scheduling groups called the initial solution. Depending on which algorithm used to obtain the initial solution, FastRLP algorithm can be divided into two algorithms. If the FastZ_X algorithm is used to obtain the initial solution, it is referred as the FastXRLP algorithm. Otherwise, if the FastZ_Y algorithm is used, then it is referred as the FastYRLP algorithm. After the initial solution is derived, RLP algorithm is used to remove the last pass by relaying messages to the unused paths of the previous passes. The RLP algorithm is executed if and only if the number of initial scheduling groups generated is more than two. This is because there is not a permutation that can be scheduled for routing in less than two groups without crosstalk in an OON regardless of the network size.

8. Numerical Results and Discussions

This section presents the experimental results obtained from the proposed algorithms. Each of the algorithms is simulated 10,000 times for each execution on different network sizes, N and presented in average for comparative analysis. Performance evaluation will be based on two types of parameters; the execution time and number of passes.

The execution time is defined as the time elapsed between the beginning and the end of its execution on a sequential computer measured in milliseconds (*ms*). The execution time calculated for each algorithm includes the time taken to generate random permutation addresses, execute window transformation, check for conflicts between the messages, mapping conflicts into the sCM and finally schedule the messages into the crosstalk-free groups for each permutation set. Minimum execution time reflects better performance of an algorithm.

Based on the time domain approach, transferring messages from source nodes to the intended destination nodes without crosstalk involves dividing the messages into

independent crosstalk-free groups called passes. These passes can be routed in one group at any given time. Less number of passes implies that more messages can be scheduled in the same pass for routing. Therefore, the number of passes obtained for an algorithm reflects the efficiency of the algorithm in terms of better scheduling strategy employed.

We divided and clustered the results of each algorithm into three categories; ZeroX, ZeroY and ZeroXY since they differ between each other in scheduling. Figure 8 and Figure 9 present the results for ZeroX algorithm, Figure 10 and Figure 11 present the result for ZeroY algorithm, while Figure 12 and Figure 13 present the result for ZeroXY algorithm in terms of the execution time and number of passes.
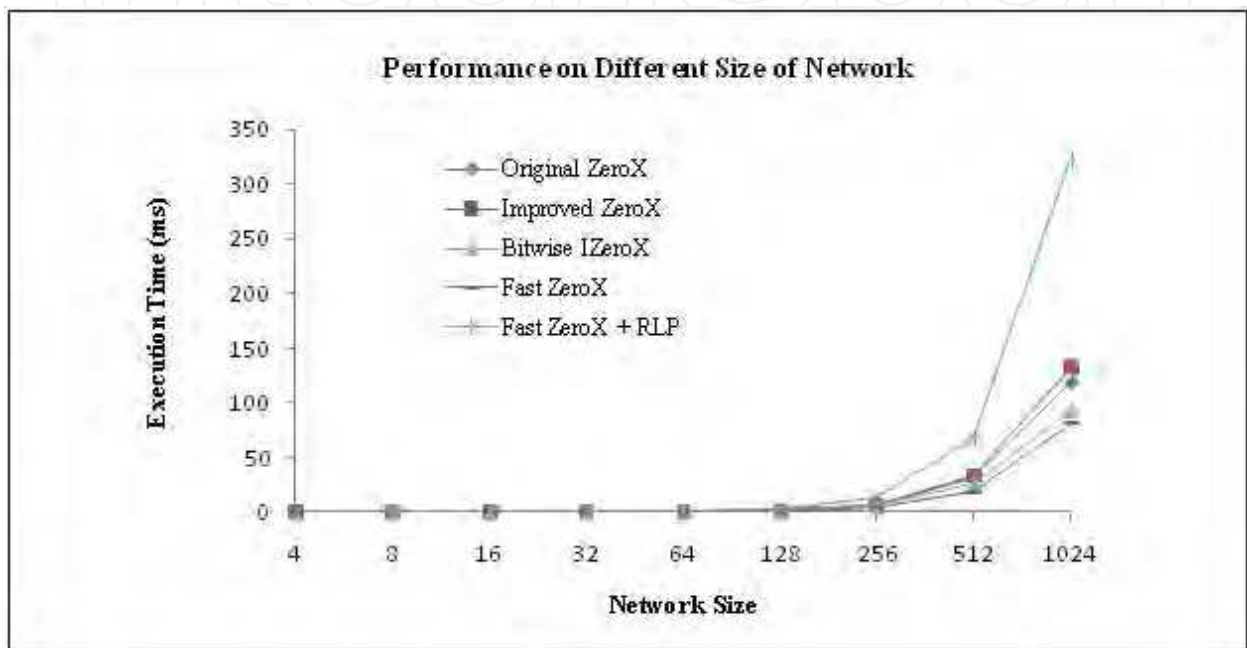
Fig. 8. Execution Time vs. Network Size of the ZeroX Algorithm

When the execution time is considered, it is evident that FastZ algorithm performs the best with the lowest average execution time consistently for all *N*, compared to all Zero-based algorithms (refer Figure 8, Figure 10 and Figure 12). Integrating FastZ and RLP algorithm result in higher execution time especially in large network, *N* = 1024 nodes. This is contributed by the RLP function embedded in the algorithm in order to reduce the number of passes.
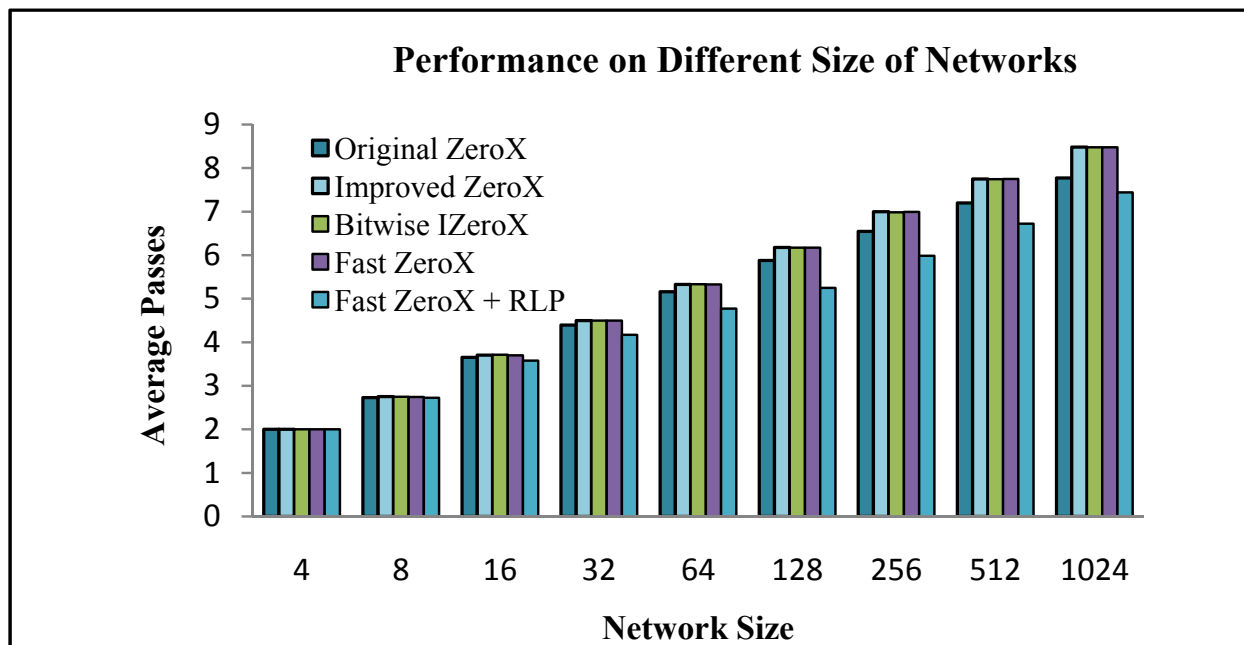
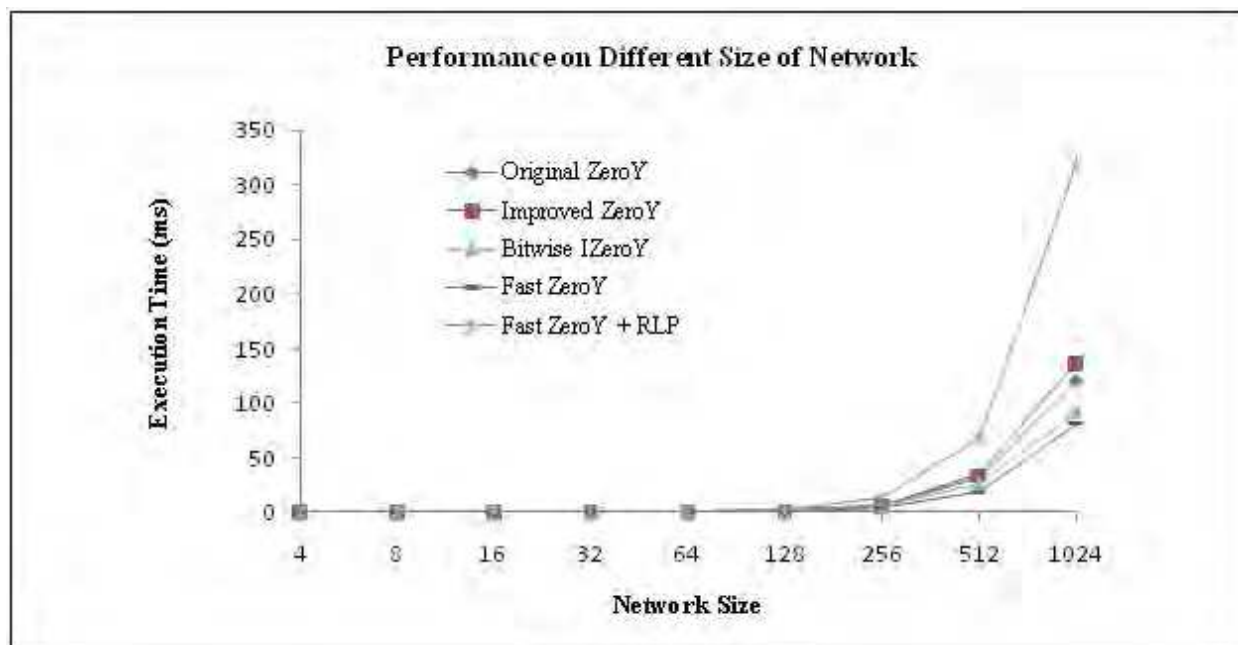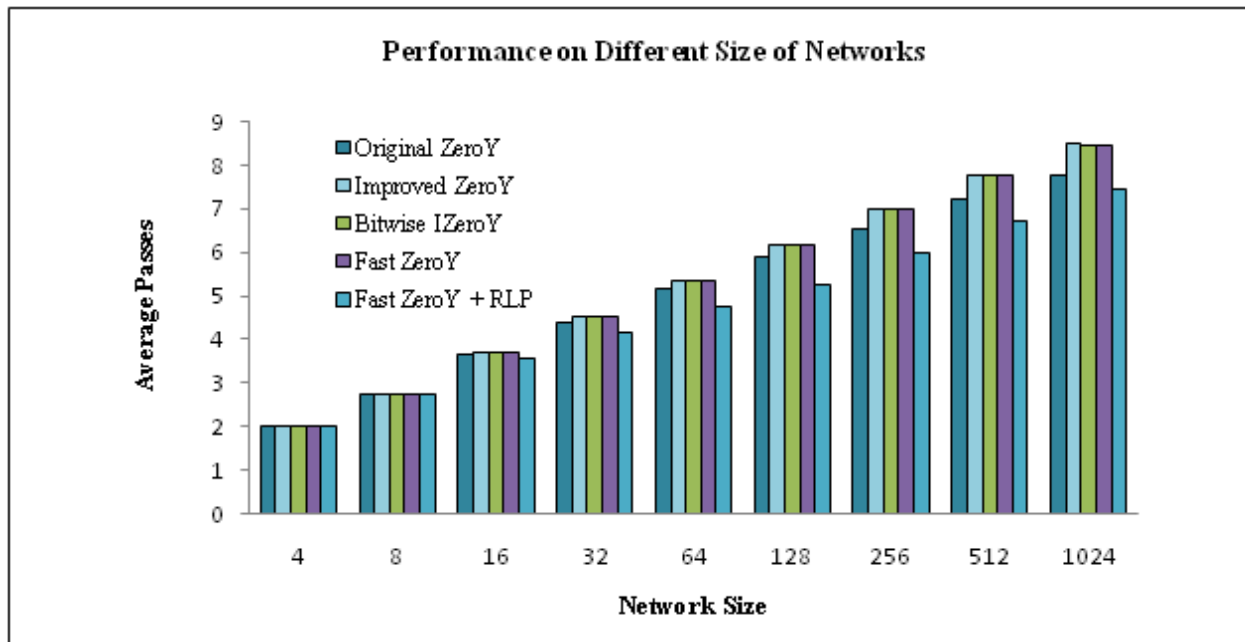Fig. 9. Number of Passes vs. Network Sizes of the ZeroX Algorithm

Fig. 10. Execution Time vs. Network Sizes of the ZeroY Algorithm

In terms of the number of passes generated for permutation routing, FastZ algorithm results closely to the IZero and BIZero algorithms. The increase in the number of passes of the FastZ algorithm compared to the original Zero algorithm was as expected. In terms of the elimination of crosstalk, the results in the number of passes are almost equal to that of the FastZ algorithm. This is mainly because the number of passes generated by FastZ algorithm may be the same as IZero and BIZero algorithms except which message(s) scheduled in each pass may be different when using FastZ algorithm.

Fig. 11. Average Number of Passes vs. Network Sizes of the ZeroY Algorithm
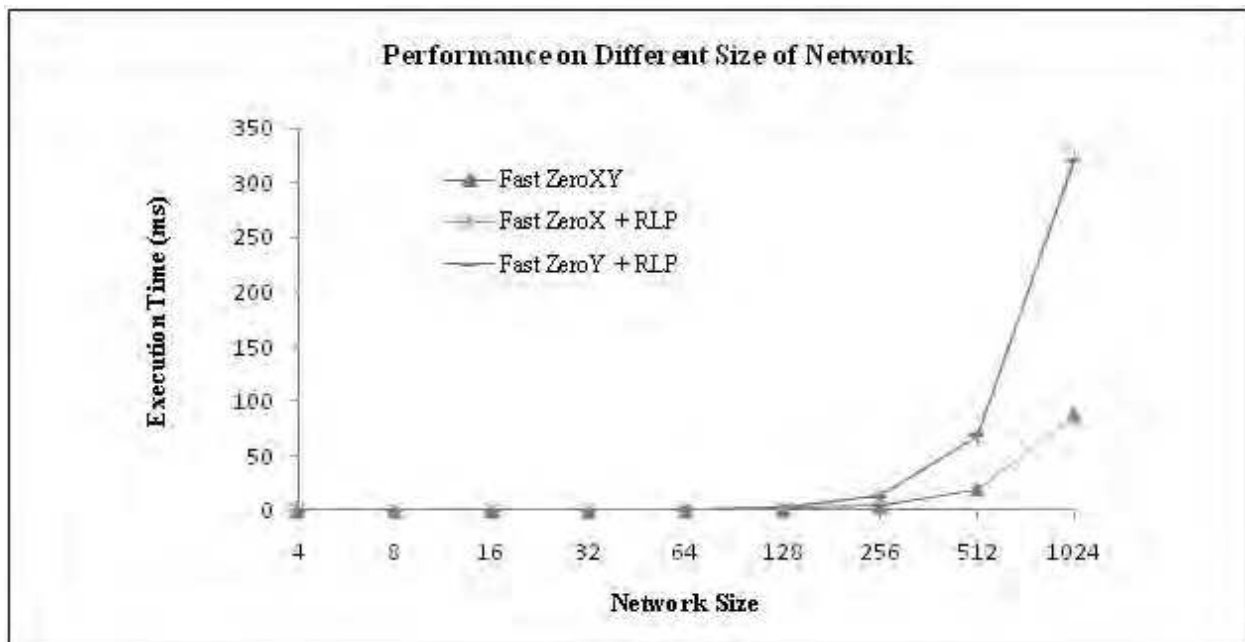

Fig. 12. Execution Time vs. Network Sizes of the ZeroXY Algorithm

Integrating the FastZ algorithm with RLP algorithm known as FastRLP algorithm has shown to successfully reduce the number of passes generated for a permutation starting from $N = 16$ onward (refer to Figure 9 and Figure 11). The results are not as significant for network size with small N (<16) because in the time domain approach no more than one input/output link can be active at any given time. Therefore, the minimum number of passes for these network ranges is limited to two passes for a permutation where in this case the RLP algorithm will not be executed at all.
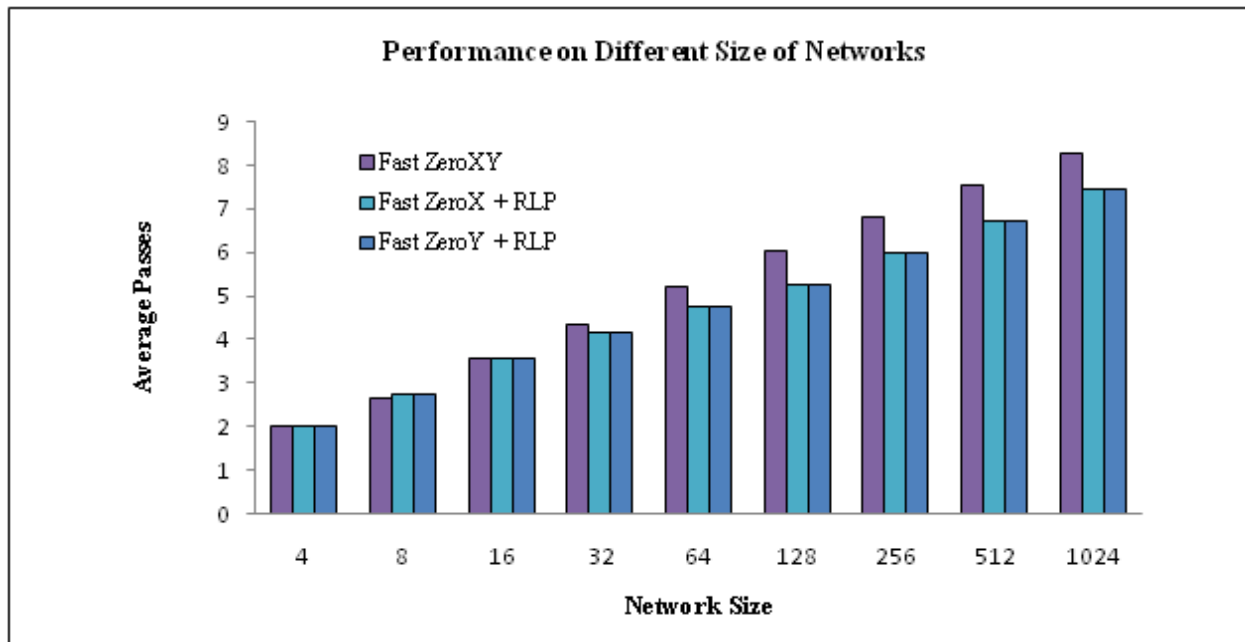
Fig. 13. Average Number of Passes vs. Network Sizes of the ZeroXY algorithm

When compared to FastZ_XY algorithm, FastRLP algorithm reduced the number of passes only when $N > 16$ as shown in Figure 13. This is because in FastZ_XY algorithm, it compares and chooses the minimum number of passes generated between FastZ_X and FastZ_Y algorithms in each execution. It was also proven in (Shahida, *et al.*, 2008a) that FastZ_XY algorithm has less number of passes compared to individual FastZ_X and FastZ_Y algorithm.

9. Conclusion and Future Works

Throughout this chapter, we have presented the development of crosstalk-free scheduling algorithms for routing in OON, a type of OMIN topology. We also presented two latest developments in crosstalk-free scheduling algorithms; FastZ and FastRLP algorithms. Using the proposed sCM to map conflicts in the network, both algorithms have proven to improve scheduling in terms of the execution time as well as the number of passes. Through simulation technique, FastZ algorithm reduced the execution time by 32% compared to previous Zero, IZero and BIZero algorithms without much difference in the number of passes generated. On the other hand, FastRLP algorithm reduced the number of passes by 11% in average compared to all Zero-based algorithms despite significant increase shown in the algorithm's execution time.

In future, we would suggest that the execution time of FastRLP algorithms be reduced using bitwise operations. The idea of sCM can also be applied to any other time domain algorithms to map conflicts identified between the messages in the network. Next, FastZ and FastRLP algorithms can be implemented in parallel to achieve exponential improvement in the algorithm's execution time. Finally, it is worth to consider the design to support for multicast communication in the network. In this case, the multilayer architecture can be incorporated with the single layer design of the OON topology.

10. References

Abed, F. (2007). *Bitwise-Based Routing Algorithms in Optical Multistage Interconnection Networks*, Master Thesis, Universiti Putra Malaysia.

Abed, F. & Othman, M. (2007). Efficient Window Method in Optical Multistage Interconnection Networks. *Proceedings of the IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, pp. 181-185.

Abed, F. & Othman, M. (2008). Fast Method to Find Conflicts in Optical Multistage Interconnection Networks. *International Journal of The Computer, The Internet and Management*, Vol. 16, No. 1, pp. 18-25.

Abdullah, M. (2005). *Efficient Sequential and Parallel Routing Algorithms in Optical Multistage Interconnection Networks*, Master Thesis, Universiti Putra Malaysia.

Abdullah, M.; Othman, M. & Johari, R. (2005). Efficient Parallel Routing Algorithms in Optical Multistage Interconnection Network, *Proceeding of IEEE International Conference on Communication Networks*, pp. 505-509.

Abdullah, M.; Othman, M. & Johari, R. (2006). An Efficient Approach for Message Routing in Optical Omega Network. *International Journal of The Computer, The Internet and Management*, Vol. 14, No. 1, pp. 50-60.

Al-Shabi, M. A. (2005). *Zero Algorithms for Avoiding Crosstalk in Optical Multistage Interconnection Network*. PhD Thesis, Universiti Putra Malaysia.

Al-Shabi, M. A. and Othman, M. (2008). A New Algorithm for Routing and Scheduling in Optical Omega Network. *International Journal of The Computer, The Internet and Management*, Vol. 16, No. 1, pp. 26-31.

Chao, H. J. & Liu, B. (2007). *High Performance Switches and Routers*. Wiley-IEEE Press.

Chau, S. C.; Xiao, T. & Fu, A. W. C. (2005). Routing and Scheduling for a Novel Optical Multistage Interconnection Networks. *Lecture Notes in Computer Science*, Vol. 3648, pp. 984-993.

Dally, W. J. & Towles, B. (2004). *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, San Francisco.

Feng, T. Y. (1981). A Survey of Interconnection Networks. *Journal of Computer*, Vol. 14, pp. 12-27.

Katangur, A. K. (2001). *Message Routing and Scheduling in Optical Multistage Networks Using Simulated Annealing*, Master Thesis, Georgia State University.

Katangur, A. K.; Pan, Y. & Fraser, M. D. (2002). Message Routing and Scheduling in Optical Multistage Networks Using Simulated Annealing. *Proceedings of the International Parallel and Distributed Processing Symposium*, pp. 201-208.

Katangur, A. K.; Akkaladevi, S.; Pan, Y. & Fraser, M. D. (2004). Applying Ant Colony Optimization to Routing Optical Multistage Interconnection Networks with Limited Crosstalk. *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, pp. 163-170.

Lu, E. & Zheng, S. Q. (2003). High-Speed Crosstalk-Free Routing for Optical Multistage Interconnection Networks. *Proceedings of the 12th International Conference on Computer Communications and Networks*, pp. 249-254.

Lu, E. & Zheng, S. Q. (2004). Parallel Routing and Wavelength Assignment for Optical Multistage Interconnection Networks. *Proceeding of the International Conference on Parallel Processing*, pp. 214-221.

Othman, M. & Abed, F. (2008). Fast Four Heuristic Routing Algorithms in Optical Multistage Interconnection Switch Networks. *Research Journal of Information and Communication Technology*, Vol. 2, No. 1, pp. 40-48.

Othman, M.; Abdullah, M. & Johari, R. (2008). Parallel Optical Window Algorithm Applied to Optical Multistage Interconnection Switch Networks. *MathDigest*, Vol. 1, No. 2, pp. 36-39.

Padmanabhan, K. & Netravali, A. N. (1987). Dilated Networks for Photonic Switching. *IEEE Transactions on Communications*, Vol. 35, No. 12, pp. 1357-1365.

Qiao, C.; Melhem, R.; Chiarulli, D. M. & Levitan, S. P. (1994). A Time Domain Approach for Avoiding Crosstalk in Optical Blocking Multistage Interconnection Networks. *Journal of Lightwave Technology*, Vol. 12, No. 10, pp. 1854-1862.

Qiao, C. (1996). Analysis of Space-Time Tradeoffs in Photonic Switching Networks. *Proceedings of the IEEE INFOCOM*, pp. 822-829.

Qin, X. & Yang, Y. (2001). Permutation Capacity of WDM Switching Networks with Limited-Range Wavelength Conversion. *Proceedings of IEEE International Conference on Parallel Processing Workshops on Optical Networks*, pp. 271-276.

Sharony, J.; Cheung, K. W. & Stern, T. E. (1993). The Wavelength Dilation Concept in Lightwave Networks – Implementation and System Considerations. *Journal of Lightwave Technology*, Vol. 11, No. 5/6, pp. 900-907.

Shahida, T. D., Othman, M. & Khazani, M. (2008a). A Fast and Efficient Crosstalk-Free Algorithm for Routing in Optical Multistage Interconnection Networks. *Proceeding of the 5th IFIP International Conference on Wireless and Optical Communications Networks*, pp. 1-5.

Shahida, T. D., Othman, M. & Khazani, M. (2008b). Fast ZeroX Algorithm for Efficient Message Routing in Optical Multistage Interconnection Networks. *Proceedings of the International Conference on Computer and Communication Engineering (ICCCE'08)*, pp. 275-279.

Shahida, T. D., Othman, M. & Khazani, M. (2008c). Fast ZeroY Algorithm for Efficient Message Routing in Optical Multistage Interconnection Networks. *Proceedings of the 3rd International Symposium on Information Technology 2008 (ITSim'08)*, pp. 2369-2374.

Shahida, T. D., Othman, M. & Khazani, M. (2008d). Integrating RLP and Fast Zero Algorithm to Improve Routing Performance in Optical Multistage Interconnection Networks. *Proceedings of the International Symposium on High Capacity Optical Networks and Enabling Technologies 2008 (HONET'08)*, pp. 34-38.

Shahida, T. D., (2009). *Crosstalk-Free Scheduling Algorithms for Routing in Optical Multistage Interconnection Networks*. Master Thesis, Universiti Putra Malaysia.

Shen, X.; Yang, F. & Pan, Y. (2001). Equivalent Permutation Capabilities Between Time-Division Optical Omega Networks and Non-Optical Extra-Stage Omega Networks. *IEEE/ACM Transactions on Networking*, Vol. 9, No. 4, pp. 518-524.

Wu, C. L. & Feng, T. Y. (1980). On a Class of Multistage Interconnection Networks. *IEEE Transactions on Computers*, Vol. 29, No. 8, pp. 694-702.

Wu, C. L. & Feng, T. Y. (1981). The Universality of the Shuffle-Exchange Network. *IEEE Transactions on Computers*, Vol. 30, No. 5, pp. 324-332.

Xiao, T. (2004). *Message Routing and Scheduling in Optical Multistage Interconnection Networks*. Master Thesis, The University of Guelph.

Yang, Y.; Wang, J. & Pan, Y. (2000). Permutation Capability of Optical Multistage Interconnection Networks. *Journal of Parallel and Distributed Computing*, Vol. 60, No. 1, pp. 72-91.

Yang, Y. & Wang, J. (2004). Designing WDM Optical Interconnects with Full Connectivity by Using Limited Wavelength Conversion. *IEEE Transactions on Computers*, Vol. 53, No. 12, pp. 1547-1556.

Vaez, M. M. & Lea C. T. (1998). Space-Wavelength Tradeoff in the Design of Nonblocking Directional-Coupler-Based Networks Under Crosstalk Constraint. *Journal of Lightwave Technology*, Vol. 8, No. 16, pp. 1373-1379.

Trends in Telecommunications Technologies

Edited by Christos J Bouras

The main focus of the book is the advances in telecommunications modeling, policy, and technology. In particular, several chapters of the book deal with low-level network layers and present issues in optical communication technology and optical networks, including the deployment of optical hardware devices and the design of optical network architecture. Wireless networking is also covered, with a focus on WiFi and WiMAX technologies. The book also contains chapters that deal with transport issues, and namely protocols and policies for efficient and guaranteed transmission characteristics while transferring demanding data applications such as video. Finally, the book includes chapters that focus on the delivery of applications through common telecommunication channels such as the earth atmosphere. This book is useful for researchers working in the telecommunications field, in order to read a compact gathering of some of the latest efforts in related areas. It is also useful for educators that wish to get an up-to-date glimpse of telecommunications research and present it in an easily understandable and concise way. It is finally suitable for the engineers and other interested people that would benefit from an overview of ideas, experiments, algorithms and techniques that are presented throughout the book.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mohamed Othman, and Tg Dian Shahida Raja Mohd Auzar (2010). The Development of Crosstalk-Free Scheduling Algorithms for Routing in Optical Multistage Interconnection Networks, Trends in Telecommunications Technologies, Christos J Bouras (Ed.), ISBN: 978-953-307-072-8, InTech, Available from: http://www.intechopen.com/books/trends-in-telecommunications-technologies/the-development-of-crosstalk-free-scheduling-algorithms-for-routing-in-optical-multistage-interconne

INTECH
open science | open minds