A MODIFIED BUFFERED ADAPTIVE ALGORITHM FOR ROUTING IN OPTICAL BENES NETWORK

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Tanvir Kaur
(800931021)

Under the supervision of:
Mrs. Rinkle Rani
Assistant Professor
CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004
June 2011

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*A Modified Buffered Adaptive Algorithm for Routing in Benes Network*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs. Rinkle Rani* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Tanvir

(Tanvir Kaur)

800931021

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Rinkle

(Mrs. Rinkle Rani)
Assistant Professor,
Computer Science and Engineering Department.

Countersigned by:

(Dr. Maninder Singh)
Head,
Computer Science and Engineering Department,
Thapar University,
Patiala.

(Dr. S. K. Mohapatra)
Dean (Academic Affairs),
Thapar University,
Patiala.

Acknowledgement

I would like to express my deepest appreciation to Mrs. Rinkle Rani, my mentor and thesis supervisor for her constant support and motivation. She had been instrumental in guiding me throughout the thesis with her valuable insights, constructive criticisms and interminable encouragement.

I am also thankful to Dr. Maninder Singh, Head, Computer Science and Engineering Department and Mr. Sumit Miglani, P.G. Coordinator for their constant support and encouragement.

I express my thanks to my family for their support and affection and for believing in me always.

In the end, I would like to thank all the faculty members and staff of the department and my friends who directly or indirectly helped me in the completion of this thesis.

Tanvir

Tanvir Kaur

(800931021)

Abstract

One of the main issues with optical networks is the problem of blocking that affects the performance of the network. Several approaches have been established in order to turn a blocking network into a non-blocking one. These approaches can be classified as strictly non-blocking, wide sense non-blocking and rearrangeably non-blocking. Out of all these approaches the rearrangeably non-blocking networks has been discussed in this thesis.

Benes network is a long established rearrangeable network to connect large switching elements. This network has the characteristic of providing multiple paths for same output request. For providing different paths several routing algorithms have been proposed in the literature.While designing the routing algorithms, attention must be paid towards the performance of the network in delivering the data packets to the required destination without blocking the network in between the different stages of the network. However in the existing partially adaptive routing algorithm, discussed in this thesis, blocking of the network was taking place whenever multiple data packets tried to use the network simultaneously.

In this thesis, a new routing algorithm named Modified Buffered Adaptive Routing algorithm has been proposed. The proposed algorithm uses buffers to improve the network performance by removing the conflicts occurring with the existing partially adaptive routing algorithm and by handling multiple inputs at a time. The results have been shown on how the proposed algorithm improves the performance in terms of removal of routing conflicts.

Table of Contents

List of Figures

Chapter 1 Introduction

1.1 Multistage Interconnection Networks

MINs consist of more than one stage of small interconnection elements called switching elements and links interconnecting them. MINs are used in multiprocessing systems to provide cost-effective, high-bandwidth communication between processors and/or memory modules. A MIN normally connects N inputs to N outputs and is referred as an NxN MIN. The parameter N is called the size of the network. Figure 1 illustrates the structure of MIN which shows the connection between p inputs and b outputs, and connection between these is via number of stages. Multistage interconnection network is actually a compromise between crossbar and shared bus networks of various types of multiprocessor interconnections networks as they:

- Attempt to reduce cost.
- Attempt to decrease the path length.

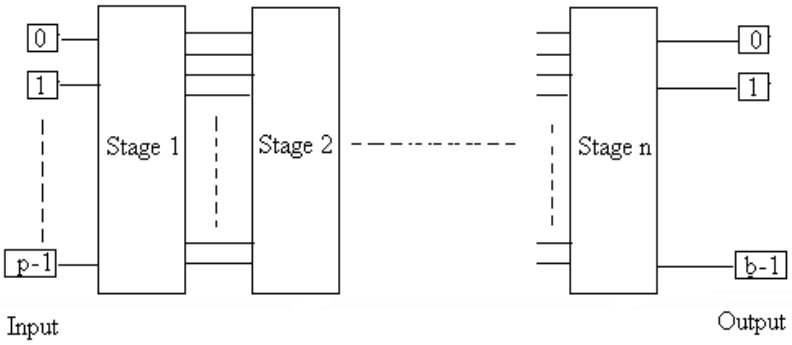


Figure 1: A Multistage Interconnection Network

1.2 Optical Multistage Interconnection Networks

Optical Multistage Interconnection networks (OMINs) are popular in switching and communication application. In these types of networks optical signal is converted

to/from electrical signal at the network input/output. This advantage makes signal transmission in optical network faster. Optical networks provide higher capacity and reduced cost for fewer applications such as internet, video and multimedia interactions. Figure 2 shows the different types of multistage interconnection networks. An OMIN can be implemented with either free-space optics or guided wave technology. It uses the time division multiplexing. To exploit the huge optical bandwidth of fiber, the wavelength division multiplexing (WDM) technique can also be used. With WDM the optical spectrum is divided into many different logical channels, and each channel corresponds to a unique wavelength. Two types of guided wave optical switching systems can be used. The first is a hybrid approach in which optical signals are switched, but the switches are electronically controlled. As such the speed of the electronic switch control signals can be much less than the bit rate of the optical signals being switched. So, with this approach there is a big speed mismatch due to the high speed of optical signals. The second approach is all-optical switching. This has removed the problem that occurred with the hybrid approach. But, such systems will not become practical in the future and hence only hybrid optical MINs are considered. In hybrid optical MINs, the electronically controlled optical switches, such as lithium neonate directional couplers, can have switching speeds from hundreds of picoseconds to tens of nanoseconds.

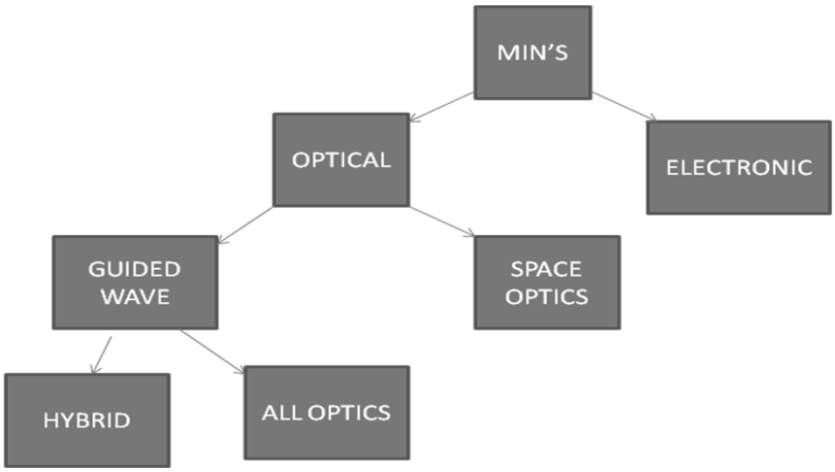


Figure 2: Types of Multistage Interconnection networks

1.2.1 Various Optical MINs

a) **Crossbar Network:** It is the squared wide-sense non-blocking network without crossover. It has N^2 switches [12].
b) **Clos Network:** It is a 3-stage network with 2 stages of r n*m switches and one middle stage of m r*r switches.
c) **Double Crossbar Network:** It consists of two crossbar like structures placed on top of each other. It is non-blocking network.
d) **N-stage Planar Network:** It is rearrangeable non-blocking network. It has first order switch crosstalk but no crossover. Its path length is N.
e) **Banyan Network:** It has best results on number of switch elements ((N logN)/2). It is a blocking network [2, 10].

Out of all the optical networks, the Banyan networks are attractive for serving as the optical switch architectures due to their nice properties of small depth and absolutely signal loss uniformity. Banyan structure is one of the candidates for serving as the directional-coupler (DC)-based switching systems that can switch signals at the rate of several terabits per second.

One of the main properties of Banyan is that only one path from a given input to a given output exists and different input-output pairs share interstage links as shown in Figure 3, therefore blocking can occur when different input-output connections are requested to be set-up in circuit-switching applications.

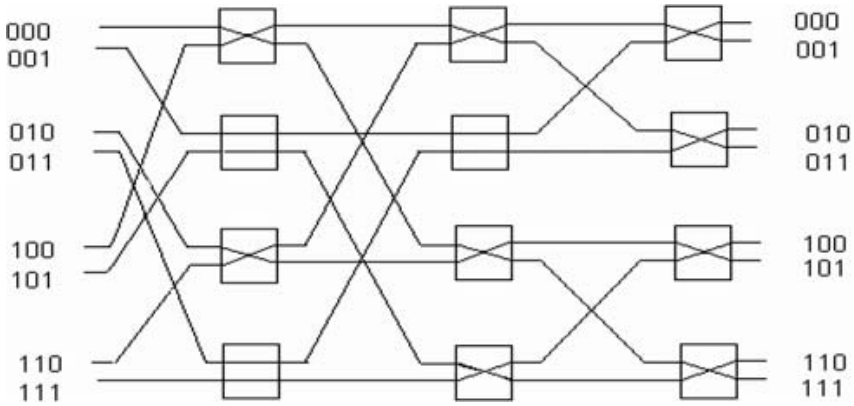


Figure 3: 8×8 Banyan Network

1.3 Switching in Optical Networks

In optical networks, circuit switching is used. Packet switching is not possible with OMINs. If packet switching is used, the address information in each packet must be decoded in order to determine the switch state. In a hybrid MIN, it means it require conversions from optical signals to electronic ones, which could be very costly. For this reason, circuit switching is usually preferred in optical MINs. So we assume that circuit switching is used.

1.4 Problems in Optical MINs

The electronic MINs and the optical MINs have many similarities, but there are some fundamental differences between them such as the optical-loss during switching and the crosstalk problem in the optical switches.

1.4.1 Path Dependent Loss

Path dependent loss means that optical signals become weak after passing through an optical path. In a large MIN, a big part of the path-dependent loss is directly proportional to the number of couplers that the optical path passes through. Hence, it depends on the architecture used and its network size which means that if the optical signal has to pass through more number of stages or switches, the path dependent loss will be more.

1.4.2 Optical Crosstalk

Optical crosstalk occurs when two signal channels interact with each other. There are two ways in which optical paths can interact in a switching network. The channels carrying the signals could cross each other. Alternatively, two paths sharing a switch could experience some undesired coupling from one path to another within a switch. Crosstalk problem is more dangerous than the path-dependent loss problem with current optical technology. Thus, the switch crosstalk is the most significant factor that reduces

the signal-to-noise ratio and limits the size of a network. The first-order crosstalk can be eliminated by ensuring that a switch is not used by two input signals simultaneously. Once the major source of crosstalk disappears, crosstalk in an optical MIN will have a very small effect on the signal-to-noise ratio and thus a large optical MIN can be built and effectively used in parallel computing systems. In order to reduce the crosstalk effect, three approaches, space dilation, time dilation and wavelength dilation, have been proposed. In both space and time dilation, crosstalk can be eliminated by ensuring that only one signal pass through an SE at a time. In other words, only one input and one output of an SE is used at any time instance. In a space dilation approach, an N × N OMIN is dilated into a network that is essentially equivalent to a 2N × 2N network [9], [11], [23], [24]. The space dilation trades the hardware cost that is more than 2 times of regular OMINs to achieve the same permutation capability. In time dilation approach, a set of permutation connections is partitioned into subsets so that the connections in each subset can be established simultaneously without crosstalk and the subsets can be used to form a sequence of configurations for the set of connections. Such a subset is called a crosstalk-free (CF) partial permutation. In the wavelength dilation approach, the crosstalk between two signals passing through the same SE is suppressed by ensuring two wavelengths to be far apart by routing [19], [20] or by using wavelength converters [16], [17] which limits the efficiency of bandwidth utilization and/or increases cost and complexity.

1.4.3 Blocking

A request is a pair of input and output requesting a connection; once routed, a request turns into a connection, which is a path link-disjoint to all other connections but sometimes the simultaneous connection of more than one terminal result in a confliction in the use of network communication links, which is referred to as blocking. In order to make it a non-blocking additional stages (horizontal or vertical) have to be added.

Chapter 2 Literature Review

2.1 Blocking in Optical Networks

When the simultaneous connections of more than one terminal results in a confliction in the use of network communication links, then it is referred to as blocking in optical networks. Since the Banyan networks have a unique path from a given input to a given output and different input-output pairs share inter stage links therefore blocking can occur when different input-output connections are requested to be set up in circuit switching applications and this makes them a suitable example of a class of blocking networks. On the other hand, there is a class of non-blocking networks in which it is possible to route from any source to any destination, in presence of other established source-destination routes, provided no two sources have same destination.

These are further classified into three levels:-

- **Strictly Non-blocking:** A network is SNB [26] if the permutation can be realized by edge-disjoint paths without rearranging active connections, any idle path can be used for each new connection request. In other words, any permutation can be dynamically realized.

- **Wide-sense non blocking (WSNB)**: A network is WSNB [26] if the permutation can be realized by edge-disjoint paths without rearranging active connections subject to the condition that a selected path is used for each new connection request. In other words, any permutation can be dynamically realized with the help of a wise algorithm.

- **Rearrangeably non blocking (RNB) or rearrangeable**: A network is said to be RNB [26] if any permutation can be realized by edge-disjoint paths when the

entire permutation is known. In other words, any permutation can be statically realized.

2.2 Methods to remove blocking

Banyan networks are attractive for serving as the optical switch architectures due to their nice properties of small depth and absolutely signal loss uniformity. One of the main properties of Banyan is that only one path from a given input to a given output exists and different input-output pairs share interstage links; therefore blocking can occur when different input-output connections are requested to be set-up in circuit-switching applications. Since a network is blocking, collisions may occur and certain packets can be lost so there is a need to remove blocking in order to increase the performance of the network.

2.2.1 Horizontal Expansion [33]

In this technique multiple paths are created between an input-output pair in which reverse of the first x stages of a regular NxN network is appended to the back of the network.

2.2.2 Vertical Stacking

In this technique multiple paths are created by vertical stacking of the networks by adding extra stages vertically.

Ideally the general scheme for building non-blocking networks is a combination of horizontal expansion and vertical stacking both of the optical networks as shown in Figure 4. The main focus has been on determining the minimum number of stacked copies (planes) required to ensure non-blocking. But the resulting horizontally expanded and vertically stacked optical networks usually take either a high hardware cost or a large network depth to guarantee the non-blocking property. Hence there is a need to find a

compromise between the hardware cost, network depth and the blocking probability of the network.

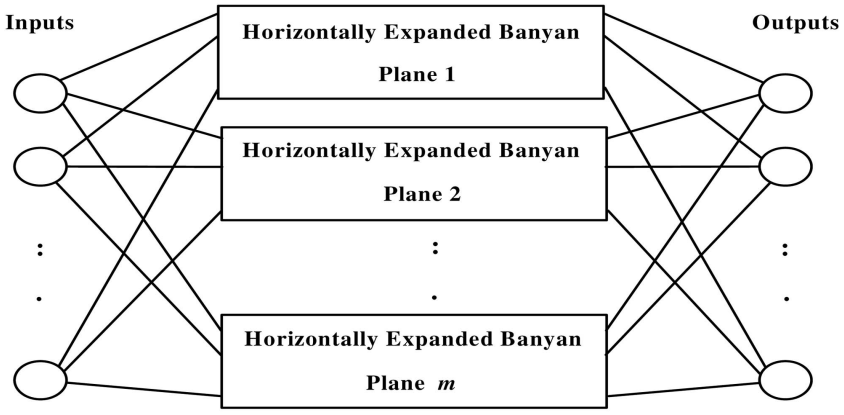


Figure 4: A combination of Vertical Stacking and Horizontal Expansion [33]

2.3 Strictly Non-blocking Conditions for MIN

2.3.1 Banyan networks

- A replicated Banyan of size N×N with SE b×b with K planes is strict-sense nonblocking [38] if and only if:

$K \geq n_c + 1$, where

$n_c = 2(b^{n-1/2}-1)$, n-odd.

$n_c = b^{n-2/2}(b+1) - 2$, n-even.

The number of replicated planes needed for the Banyan in order to achieve the non blocking property has been analyzed using this formula.

- Another necessary and sufficient condition that make a N x N network made by vertical replication of horizontally-expanded banyan networks each with n + m stages ($0 < m < n - 1$) a strictly non-blocking network is given. It is specified by the number K of networks vertically stacked that is:

$K \geq 32\ 2^{(n-m)/2} + m-1$, when $(n + m)$ is even.

[18]

$K \geq 2^{(n-m+1)/2} + m-1$, when $(n + m)$ is odd.

2.3.2 Shuffle Exchange networks

1. The maximum number of paths that can be blocked in the internal shells of a Shuffle Exchange network with s stages is found to be equal to that of recursive banyan networks with $n + m = s$ stages and thus given by:

$N_b = 32\ 2^{(s/2)} - 2^{(s-n+1)}$, when s is even

$N_b = 2^{(s+1)/2} - 2^{(s-n+1)}$, when s is odd

Where N_b = no of blocked paths.

a) When $s = n+1$ stages:

An N x N network made by vertical replication of K Shuffle Exchange networks with $s = n + 1$ stages is strictly non blocking if and only if:

$K \geq 32\ 2^{(n-1)/2}$, when n is odd

$K \geq 2^{n/2}$, when n is even.

b) When $s = n + 2$ stages:

$N_b = 32\ 2^{(n+2)/2} - 8$, when n is even

$N_b = 2^{(n+3)/2} - 8$, when n is odd

c) When $s = n + 3$ stages:

$N_b = 32\ 2^{(n+3)/2} - 16$, when n is even

$$N_b = 2^{(n+4)/2} - 16 \text{ , when n is odd}$$

2.4 Routing in Rearrangeable Networks

In (2n-1) stage rearrangeable networks, the routing time for any arbitrary permutation is $\Omega(n^2)$ compared to its propagation delay O(n) only. So the attempt is to identify the sets of permutations, which are routable in O(n) time in these networks. Therefore, in all these cases, the time needed to realize an arbitrary permutation in rearrangeable networks is dominated by the set-up time. However, in a Benes network, many useful permutations, often required in parallel processing environments are found to be self-routable. The self-routable (SR) permutations are classified into four categories namely:

(i) Top-Control Routable set of permutations (TCR)
(ii) Bottom-Control Routable set of permutations (BCR)
(iii) Least-Control Routable set of permutations (LCR)
(iv) Highest-Control Routable set of permutations (HCR)

An algorithm was developed that will detect whether any N×N permutation P belongs to any of the four classes, and if yes, it also generates the appropriate control vectors for routing P. This algorithm can be implemented on a multi-processor system with a time complexity O(n). Therefore, this algorithm enables us to route all the permutations in the union of TCR, BCR, LCR and HCR in O(n) time. By this algorithm a much larger class of permutations was identified to be self-routable in Benes network than it has been reported earlier. The exact number of permutations covered by each of these classes is found for N=4 and 8. These experimental results help us to have an idea about the total number of permutations self routable in Benes network by any of these self routing strategies. In self-routing algorithms, the setting of a switch is done locally using the destination tags of its two inputs. For any (2n-1) stage rearrangeable MIN, which is a concatenation of two full-access unique-path MINs, with the central stage

being common (as it is in the Benes network), the routing through the last n-stages, must follow the normal destination tag routing scheme. Figures 5-8 show examples of the routing schemes for different permutations on a Benes network.

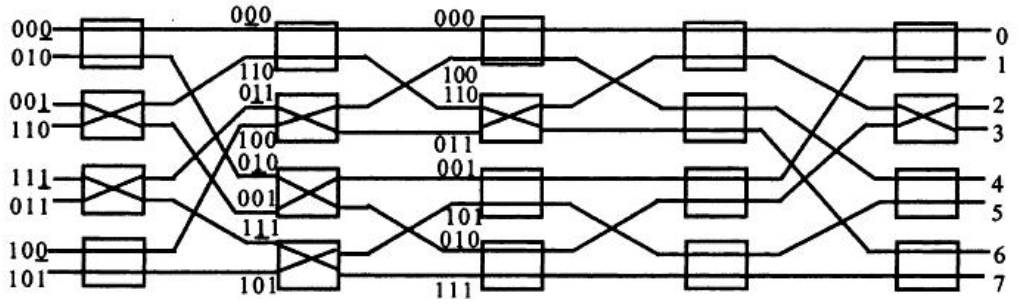


Figure 5: Routing of permutations: (0 2 1 6 7 3 4 5) by top-control-routing [6]

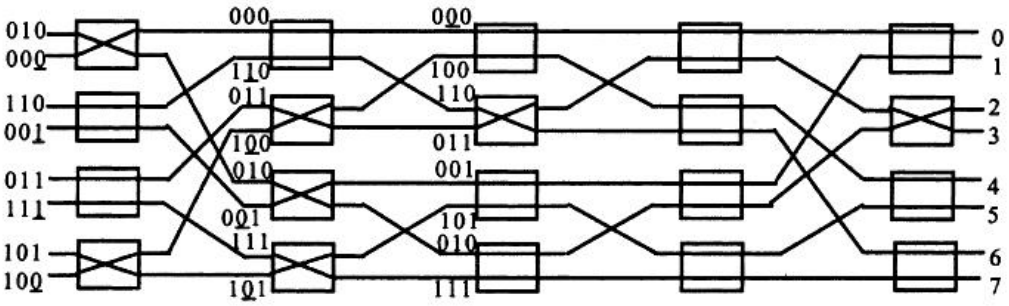


Figure 6: Routing of permutations: (0 2 1 6 7 3 4 5) by bottom-control-routing [6]

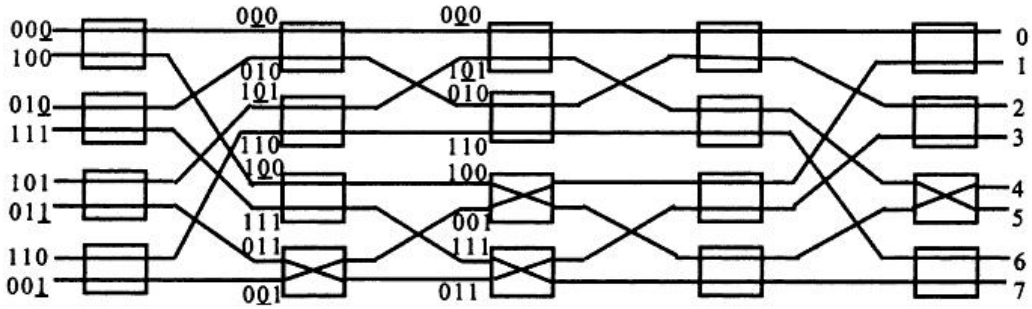


Figure 7: Routing of permutations: (0 4 2 7 5 3 6 1) by least-control-routing [6]

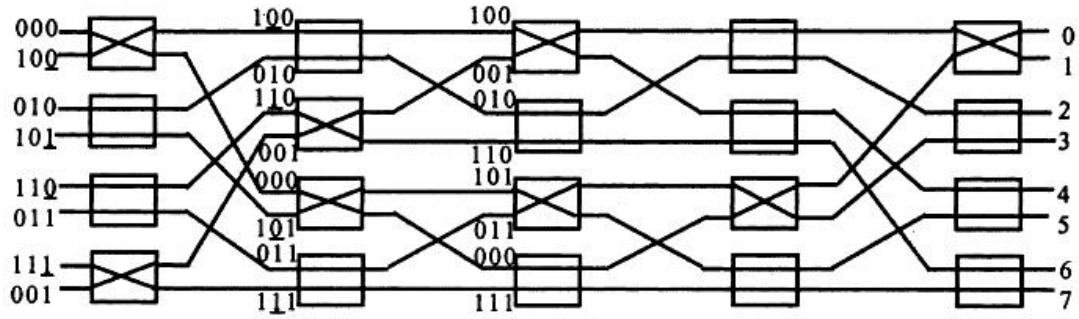


Figure 8: Routing of permutations: (0 4 2 7 5 3 6 1) by highest-control-routing [6]

2.5 Routing algorithms for Rearrangeably Non-Blocking (Benes Network)

Routing algorithms are needed to find a path that packets travel between any pair of source and destination nodes. All of these algorithms run in time $O(N\log_2 N)$ in a single processor.

2.5.1 Matrix based Method for Routing in Benes Networks [3]

The algorithm starts by setting a switching element in to straight through position at stage 0 for any one of its input for a particular output request. The algorithm continues this process for all the switching elements in that stage until we encounter any conflict of values in the cells. For any conflict, we change the switching elements setting by identifying cells sub-matrix. The algorithm then proceeds to the next stage and continue setting up the switching element in that stage. The algorithm is as follows:

A. Phase 1 (Setting the Switching Elements)

Input: Input Permutation $P_{0:(N-1)}$.

Output: Routing tags for stage i, $0 \leq i \leq (\log N-2)$.

Step 1: Create 2i sub-matrices where $0 \leq i \leq (\log N- 2)$ and i identifies the stages of the network.

Step 2: At stage i where $0 \leq i \leq (\log N-2)$, index the rows of sub-matrices from 0 to N/2 sequentially. Create row sized columns for each sub-matrices. Each row takes values for two inputs for the same switching element (SE) at the input stage and each column identifies requested switching element port for specific inputs.

Step 3: At state 0, start with input 0, identify the requested output switching element. In the row of input port put a 0 at the column for requested switching element. For input 1, put a 1 at the same row but in the column that points to the requested output switching element.

Step 4: Continue Step 3 for remaining input/output requests of the permutation. For each row and column only one 0 and 1 can be used. In case of a situation with multiple 00s or 10s the sub-matrix enters into a blocking state. To break the blocking state, goto phase 2.

B. Phase 2 (Rearrangements Inside Sub-matrices)

Input: Blocking State in a sub-matrix.

Output: Rearrangement of 00s and 10s to unblock a blocked request.

Step 1: In the blocked row, rearrange the values between cells in that row.

Step 2: In case of any new blocking introduced because of last rearrange, change the value of the column other than the one affected by the rearrange. If there is no blocking, then the request is unblocked, goto Phase 1 to set switching elements for remaining requests else goto step 1.

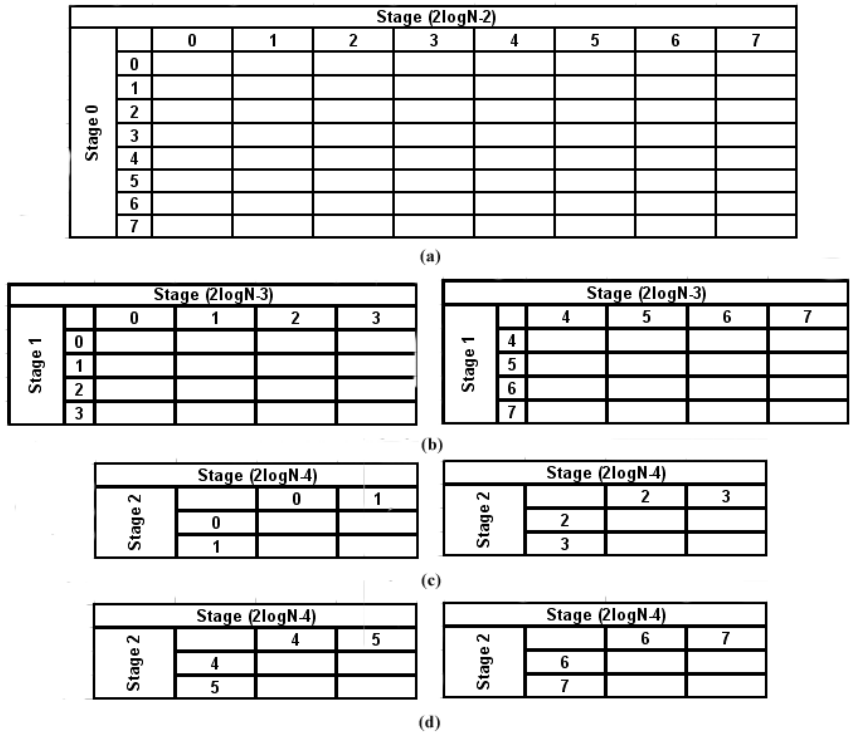


Figure 9: Sub-matrices format for a 16 × 16 Benes Network [3]

2.5.2 Dynamic Path Selection algorithm [4]

A. Phase

Input: Input Permutation $P_{0:(N-1)}$.

Output: Set the switching element at stage 0 and (2logN− 2).

Step 1: Start with the first switching element, SE [k, 0], $0 \le k \le (N/2) - 1$, set the switching element to STATE [k, 0] = 0. Set the STATE of the output switching element SE (P(i)), $0 \le i \le (N - 1)$, to 0 or 1 depending on the requested output. For the request, if Sub network = 0 and requested Port = 0, set STATE [r, (2logN − 2)] = 0, $0 \le r \le (N/2) - 1$ else set to 1.

Step 2: Check if any port remains at SE (P(i)), after Step 1.

If

yes then goto Step 3.

Else

goto Step 4.

Step 3: Start with the remaining port of SE (P(i)), $0 \leq i \leq (N - 1)$. Check the state of the corresponding switching element at stage 0. If STATE [k, 0] = NULL, $0 \leq k \leq (N/2) - 1$, set the switching element to either 0 or 1 depending whether the connection is coming from upper or lower subnetwork and the requested port is even or odd.

Step 4: If

Any port remains at the input switching element after step 3, then set up paths between that port i and corresponding output port P(i). Set the state of the output switching element if the state was NULL.

Step 5: Continue Step 2 to Step 4 till all the connections forms a logical connected graph with at maximum N/2 complete cycles. This connecting graph will indicate that all the switching elements in stage 0 and stage $(2\log N - 2)$ are set.

B. Phase 2

Input: Input Permutation $P_{0 : (N-1).}$

Output: Set the state of the switching elements from stage 1 to $(2\log N - 3)$.

Step 1: Start with switching element, SE [0, 1], check the the status variable for SE [0, 1]. If STATE [0, 1] = NULL, set it to 0 or 1 depending on the value of input port. If PORT = 0, set to 0, else 1 and go from stage s to (s + 1), $1 \leq s \leq (\log N - 2)$. If the SE [0, 1] was set before, then use the unused port.

Step 2: From stage $(\log N - 1)$, use first $(\log N - 2)$ MSB bits of the output port for establishing path from $(\log N - 1)$ to $(2\log N - 3)$.

Step 3: If there is any blocking,

call Phase 3.

Else

goto Step 4.

Step 4: If

Any port remains at the switching element last used in step 2 at stage (2logN − 3) Use that port for reverse routing.

Go to stage (logN−2) from (2logN−3) by following the switching element setting rules used in step 1 and from (logN −2) to stage 1 apply step 2.

Else

Go to stage 1 and start with any available switching element with STATE [k, 1] = NULL, $0 \leq k \leq (N/2) - 1$. Phase 2 can set up paths for all input-output requests if no blocking arises inside the switching elements between stage (logN−2) and (2logN−3). In case of any blocking, call Phase 3 and resolve the conflict by selecting alternative paths for the request and continue its execution till it reaches the end of the entire request.

C. Phase 3

Input: Blocking between stages (logN −2) and (2logN − 3).

Output: Alternative blocking free paths.

Step 1: For forward routing follow Step2, else follow Step 3

Step 2: Blocking at stage k, $(\log N - 1) \leq k \leq (2\log N - 3)$, go back to switching element SE [i, j] at stage j using link pattern Lj , $0 \leq i \leq (N/2) - 1$ and $1 \leq j \leq (\log N - 2)$ and choose the alternative port of that switching element.

If

Alternative port is not free, go back to next step and choose the alternative port of the switching element at that stage and continue routing.

Else

Go to Step4.

Step 3: Blocking at stage k, $1 \leq k \leq \log N$, go back to switching element SE [i, j] at stage j using link pattern Lj ,$(\log N-1) \leq i \leq (2\log N-3)$ and $\log N \leq j \leq (2\log N-3)$ and choose the alternative port of that switching element.

If

Alternative port is not free, go back to next step and choose the alternative port of the switching element at that stage and continue routing.

Else

Go to Step4.

Step 4: In case where it is not possible to find any available path for the request, drop the request and start with any other available request at stage 1.

2.5.3 Self routing algorithm [25]:

The self- routing algorithm routes a tag as follows:

- If the i-th bit is connecting bit for a stage, then the i-th bit of the tag called the routing bit is used for routing through that stage, it is routed to the upper(lower) routing line of the switch if the routing bit is 0(1).
- When the routing bits of both the tags at a switch are the same then there exists a conflict in setting up the switch since both them specifies the same output line.

Algorithm:

- For the first n-1 stages, the switches are set up such that input line with smaller destination tag value is routed according to its routing-bit.
- For the next n stages, the switches are set up using the above 2 steps.
- In case of any conflict, the algorithm gives priority to the tag having the smallest value.

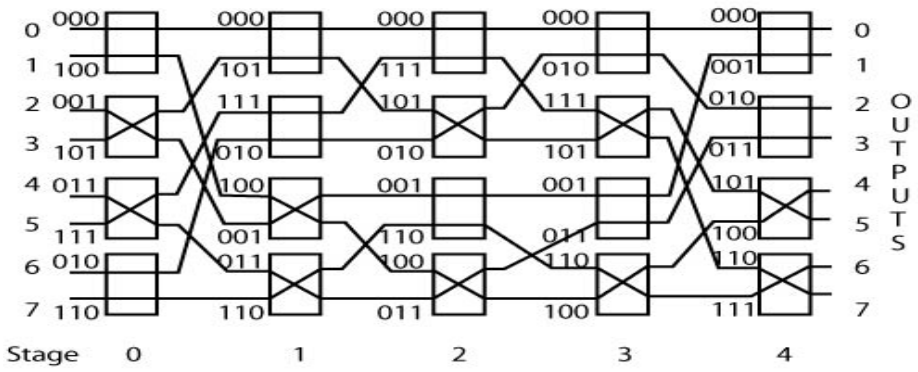


Figure 10: Routing a linear Permutation [25]

Chapter 3 Problem Statement

The most important issue clouding the future of optical interconnection networks is their practicality. Numerous ad hoc approaches have been developed which demonstrate that optical interconnection networks can be designed but the reduction of such networks to economically viable units is very difficult. In addition to it there occurs a problem of blocking in the optical networks which is affecting the performance of these networks. Therefore more emphasis should be on the development of non-blocking networks.

The strictly non-blocking conditions for the optical networks usually cause a lot of hardware cost so a new class of networks came into being called the rearreangeable network in which there was no need of any extra planes and a conflict free path can be established by rearrangement of the existing connections. Therefore the problem of blocking was solved by the rearrangement of the connections. For this rearrangement of the connections we require the routing algorithms that aim to seek a path that data packets travel between any pair of source and destination nodes. Several routing algorithms have been developed for different rearrangeable networks. In this thesis a most popular type of rearrangeable network called the Benes Network has been taken and several routing algorithms have been studied in order to develop a non-blocking network.

In this thesis, a partially adaptive routing algorithm for routing in the Benes Network has been undertaken and studied. The main problem with this existing algorithm was that it only worked when one data packet was to be routed at a time but failed when multiple data packets were needed to be routed simultaneously. This resulted in the routing conflicts when multiple data packets tried to use the same path which in turn affected the performance of the network.

To avoid the above problem **"A New Modified Buffered Routing Algorithm"** has been proposed in this thesis. The proposed algorithm uses the concept of buffers for temporary storage of the data packets at different stages of the Benes Network. Whenever any path is utilized by one data packet and the other data packet tries to use

the same data path then the information of that particular data packet gets stored in the buffer maintained at that stage and it can wait for the time till the path becomes free. Hence in the next cycle when the path becomes free then the data packet stored in the buffer can utilize that path and move onto the next stage without blocking the network. This would resolve the problem of the routing conflicts that were arising earlier when the existing algorithm was used. A simulator in C language has been developed to illustrate the following features:

a) To show the working and the problems of the Partially Adaptive Routing algorithm.
b) To show the working of the newly proposed Modified Buffered Adaptive Routing Algorithm.

Chapter 4 Modified Buffered Adaptive Routing Algorithm

4.1. Introduction

As the level of complexity of digital systems increases, the problem of interconnecting subunits is receiving increasing attention. In order to cope with the immense diversity of applications and traffic volume, interconnecting optical networks are needed but one of the main issues in optical networks is the problem of blocking which should be considered in order to fully exploit the benefits of the high speed optical networks. A network is considered blocked, if a given pair of idle terminals cannot be connected. A network can be non-blocking in three senses, these are:

- If a request can always be routed under the condition that existing connections are allowed to be rearranged (rerouted).
- If rearrangement of present calls unblocks the requested call (non-blocking in the wide sense)
- If a network has no blocking states (non-blocking in the strict sense).

The high degree of connection capability in strictly non-blocking and wide sense non blocking networks is at a high hardware cost whereas the rearrangeably non-blocking networks are usually constructed with lower hardware cost and can establish a conflict free path for the connection from any idle input to any idle output if the rearrangement of the existing connections is allowed.

4.2 Benes Network

The Benes network is one of the choices for optical networks because of its simple topology and easy scalability with low degree. An N×N Benes network is a well-known rearrangeable multistage interconnection network (MIN), with (2n-1) stages ($n=\log_2 N$), that can connect its N inputs to its N outputs in all possible ways thus enabling non-blocking communication between any pair of idle terminals through reassigning the

exiting links. An N×N Benes network, which is also called B(k), can be constructed by adding one stage including N / 2 (N = 2k) switching elements at the left and the right of two copies of B(k-1) respectively. B(1) is the smallest and simplest Benes network, which is just the 2×2 switching element. The interconnection pattern between any two stages could be defined by three types of permutations. They are shuffle, sub-shuffle and reverse-shuffle permutation. A 8×8 Benes network (B(3)) constructed from 4×4 Benes network (B(2)) is shown in Figure 11.

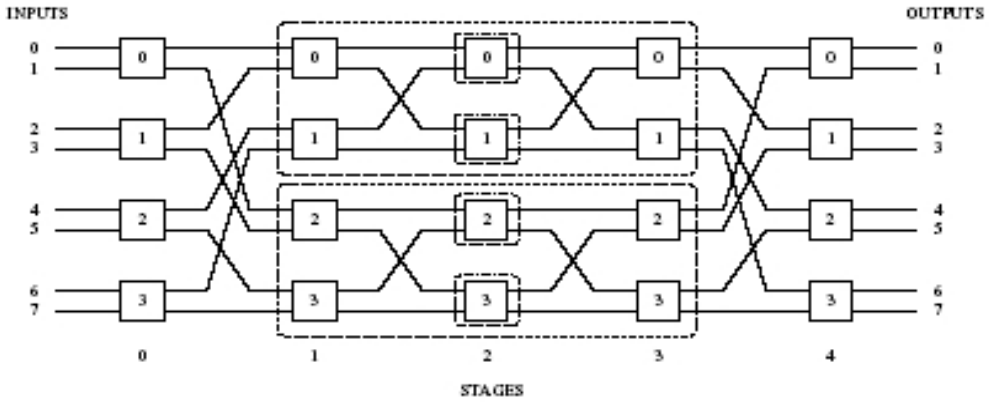


Figure 11: 8×8 Benes Network

4.3 Routing in Benes Network

In order to select a path from source to destination address among all the possible choices, there is a need of the routing algorithms. Routing algorithms are classified into deterministic and adaptive. In deterministic routing, which is also called obvious routing, the path always depends on the source and destination address completely without considering traffic and congestion of network. However, adaptive routing could remodify routes dynamically according to present conditions of the network at the expense of a more complicated logic.

4.3.1 Deterministic Routing

In a well-known deterministic algorithm routing of the packets depends on the routing label. The routing label, which is formed by the destination address, contains 2k −1 bits. The corresponding bit of the routing label at each stage is called routing bit. Operation of switching node is determined by the value of the routing bit. If the value is zero, packet is routed to the upper output line of the switching node, otherwise to the lower output line. Using this routing technique, only one path can be used between the given pair of source and destination nodes. But this deterministic routing could not provide good performance when the network is congested. In contrast, the adaptive routing algorithm could obtain better performance at the same network conditions by using alternative routing paths. As the packet injection rate increases, the advantage of applying adaptive routing is more obvious.

4.4 Partially Adaptive Routing Algorithm for Benes Network

The algorithm enables 2k−1 different paths for packets traveling the Benes network between a particular pair of source and destination nodes. Packets can select any one of these paths routed to the final destination according our routing algorithm. This approach reduces the possibility of congestion effectively. The partially adaptive routing algorithm is described as follows:

The partially adaptive routing algorithm for N×N Benes network:

Index Terms:

a) Current node: the current node is given by : C(position, stage)
b) Destination node: the destination node is represented by : D(position, 0)
c) N = 2k.
d) p ($1 \leq p \leq k$) is an integer.

Routing Procedure:

If

The local node address (y > = 0 && < k -1)

then

Select any one of the two output ports of the current switching node.

Else

If

(local node address y == (2k - p))

{

If

(destination node address x/ 2^{p-1}%2 == 0)

then

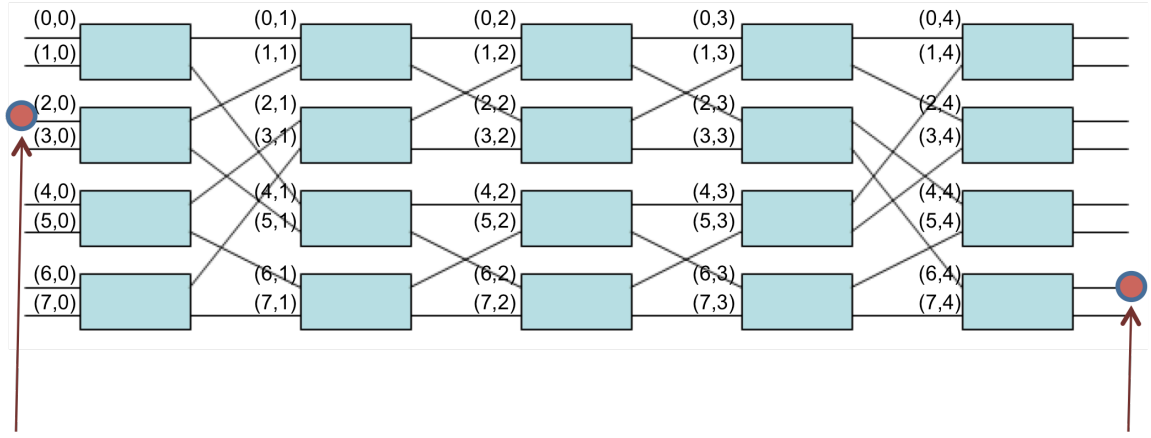
Select the upper output port of the current switching node.

Else if

The destination node address x/ 2^{p-1} % 2 == 1) then select the lower output port of the current switching node.

}

To illustrate this routing technique, example of packets routed from (2, 0) to (6, 4) in 8×8 Benes network is given in Figure 12.



Initial position of data bit

Final intended position of data bit (after routing)

Figure 12: Routing of packets in 8×8 Benes network

1. According to the partially adaptive routing algorithm, a data packet's address is represented as (x, y). x denotes the input switching element and y denotes the current stage. In the above example, initially the data is at the switching node (2, 0) where 0 is the current stage in the network and 2 is its input.

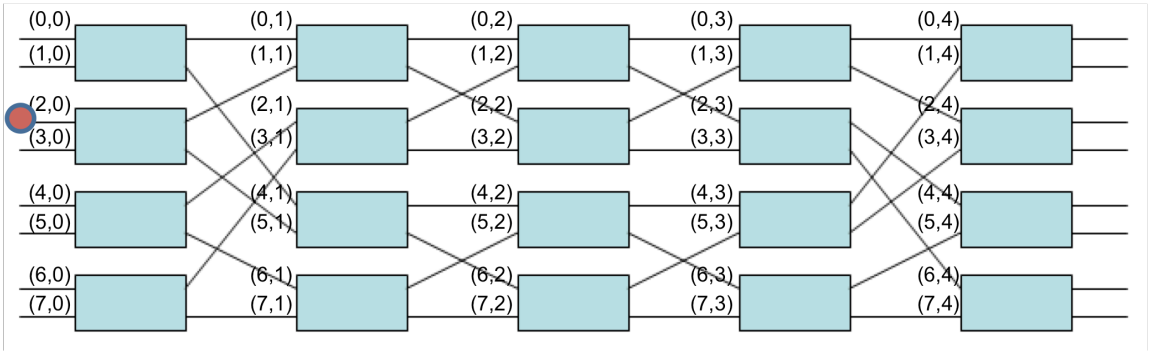


Figure 13: At stage 0, data is at (2, 0)

2. Going into the next stage, since the y coordinate of the data packet's address i.e y >= 0 and < k-1(i.e. 2), any one of the two output ports of the current switching node can be selected. By default (as per the implementation) the upper output port as shown in the Figure given below has been selected.

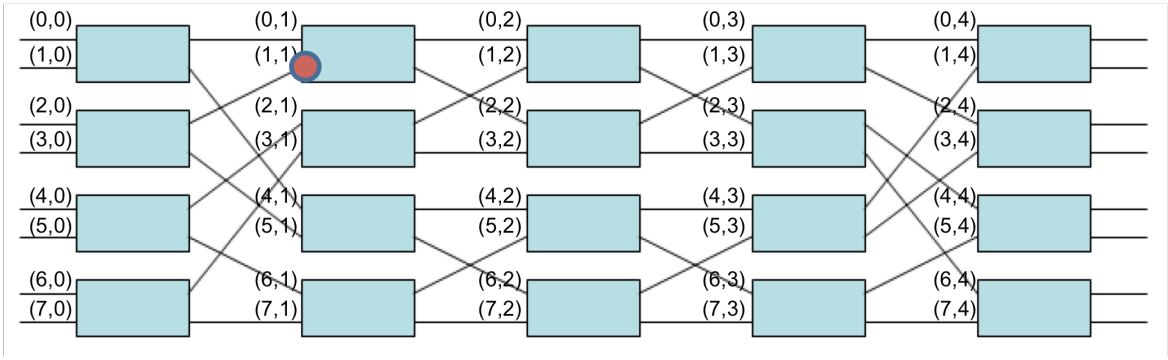


Figure 14: At stage 1, data is at (1, 1)

3. Repeating the same step again, data packet moves to the next stage again selecting the upper port of the switching node as shown in Figure 15.

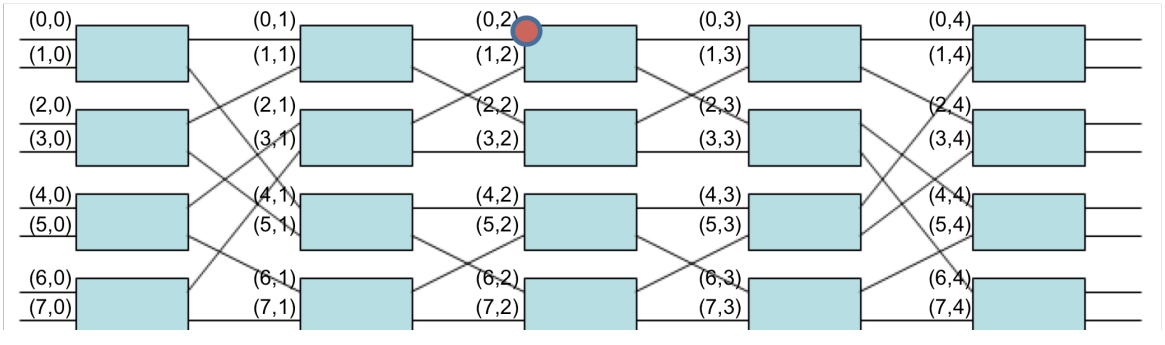


Figure 15: At stage 2, data is at (0, 2)

4. Now after reaching stage 2, the condition y >= 0 and y < k-1 fails since the value of y > k-1. So according to the else part of the algorithm, the value of $x/ 2^{p-1}\%2$ comes out to be equal to 1, hence the lower port of the node is selected and the data packet now moves to stage 3 as shown in Figure 16.

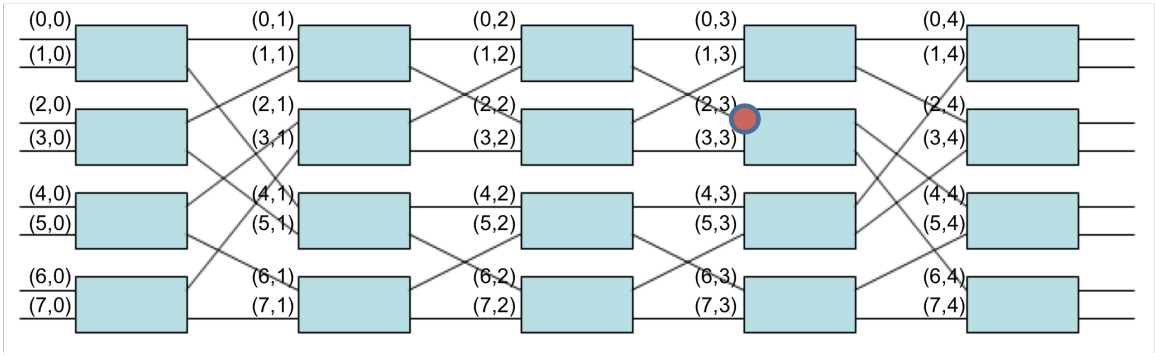


Figure 16: At stage 3, data is at (2, 3)

5. Repeating the same procedure again, the data packet uses the lower port to reach the next stage as shown in Figure 17.

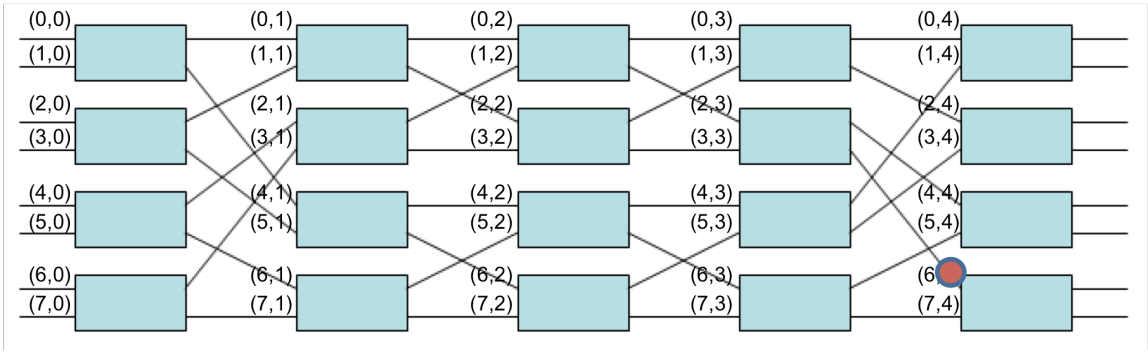


Figure 17: At stage 4, data is at (6, 4)

6. At stage 4, following the same procedure, the data packet uses the upper port since the value of $x/ 2^{p-1}\%2$ comes out to be equal to 0 and it reaches the intended final position as shown in Figure 18.

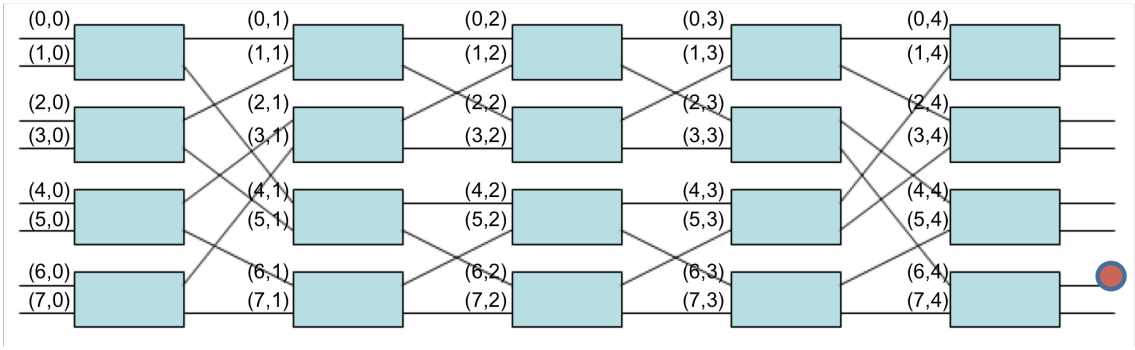


Figure 18: Data is at the final position

Limitations of the algorithm:

However this particular algorithm had a limitation and it was not suitable when multiple inputs needed to be realized at the same time and was only applicable for routing one data packet at a time. Therefore in order to fully exploit the benefits of the network performance of the Benes network there was a need to route multiple data packets simultaneously at a single instant of time. This limitation of the algorithm is illustrated below by taking an example of three data packets that are needed to be routed simultaneously:

Data at (2, 0) to be sent to (6, 4)

Data at (6, 0) to be sent to (5, 4)

Data at (7, 0) to be sent to (1, 4)

1. Initially the data are at (2,0), (6,0), (7,0) as shown in the Figure 19.

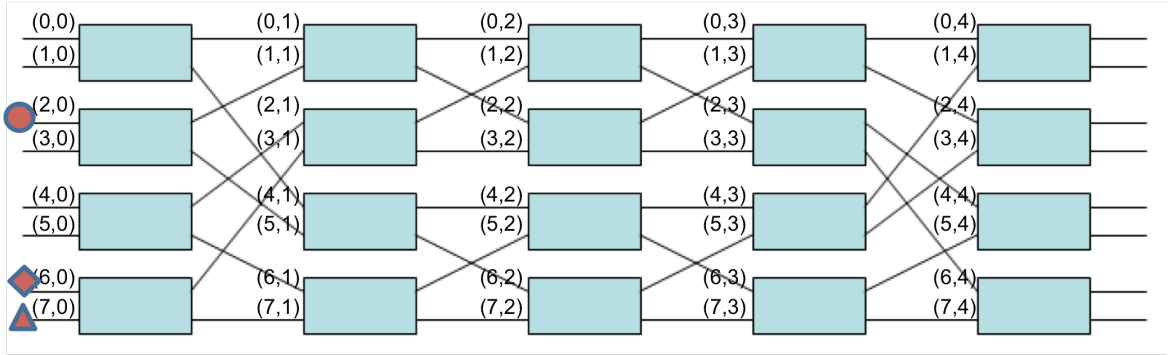


Figure 19: At stage 0, data are at (2,0), (6,0), (7,0)

2. According to the algorithm, the data packets adopt the upper port of the nodes and consequently move to the next stage as shown in the Figure 20.

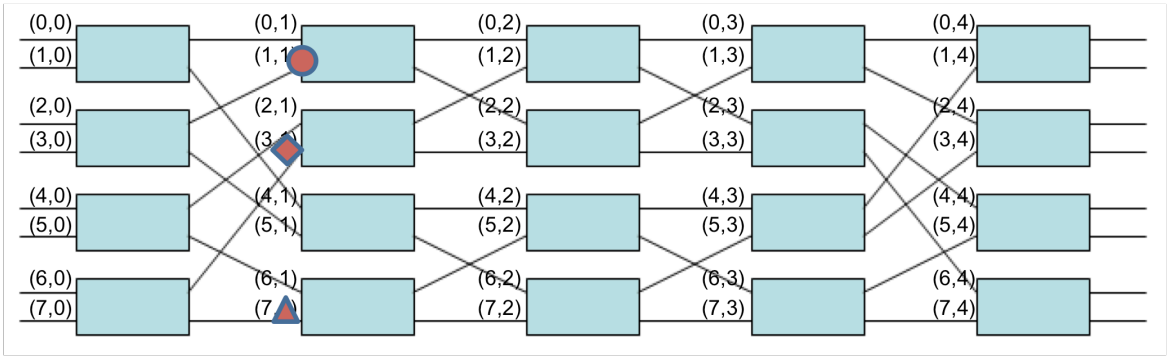
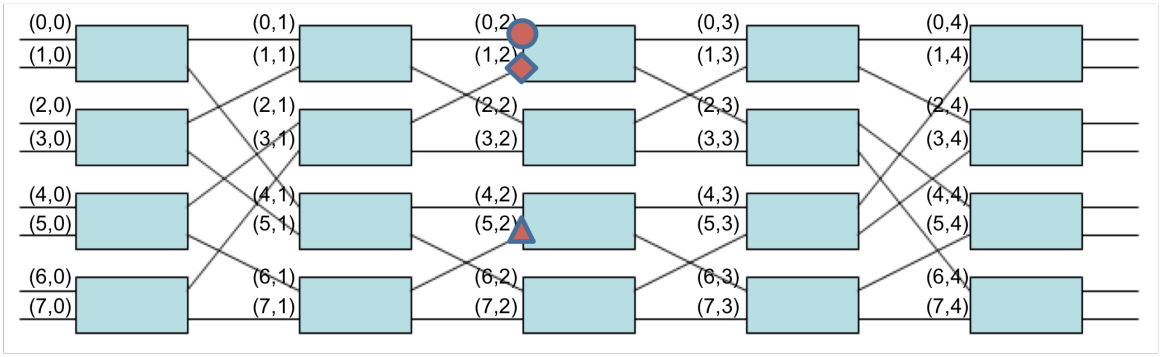


Figure 20: At stage 1, data are at (1,1), (3,1), (7,1)

3. Following the same step again, the data packets are routed onto the next stage.

Figure 21: At stage 2, data are at (0,2), (1,2), (5,2)



4. The else part of the algorithm here comes into practice and the packets are routed onto the next stage. According to the algorithm, the first data packet utilizes the lower output port and reaches the next stage at (2,3) while the second data packet also follows the same path and reaches at (2,3) by adopting the lower output port. This causes a routing conflict and ultimately results in the blocking of the network which proves to be a big limitation of the algorithm. The third data packet follows the upper output port to move to the next stage as shown in the Figure 22.

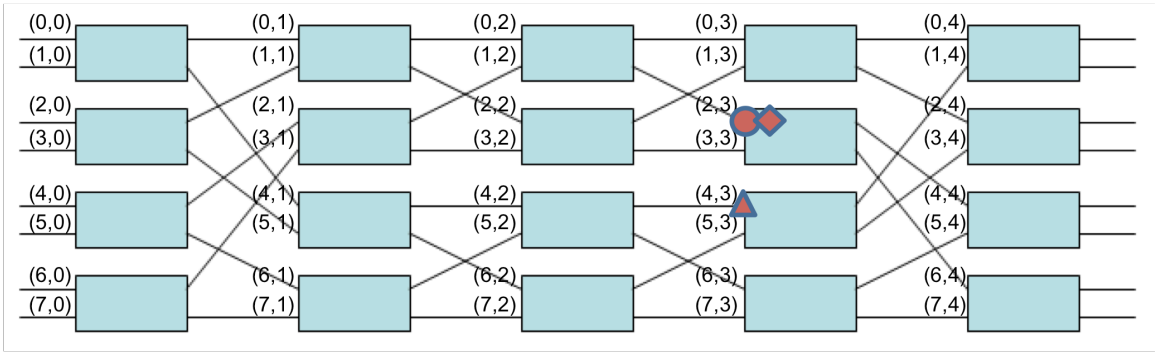


Figure 22: A routing conflict arises when the data packets are at stage 3 at (2, 3), (2,3) and (4, 3)

If the algorithm would have worked properly and there would not have been any kind of routing conflict ie. the algorithm would easily have worked for multiple inputs at a time then the following route as shown in Figure 23 would have been followed by the data packets.

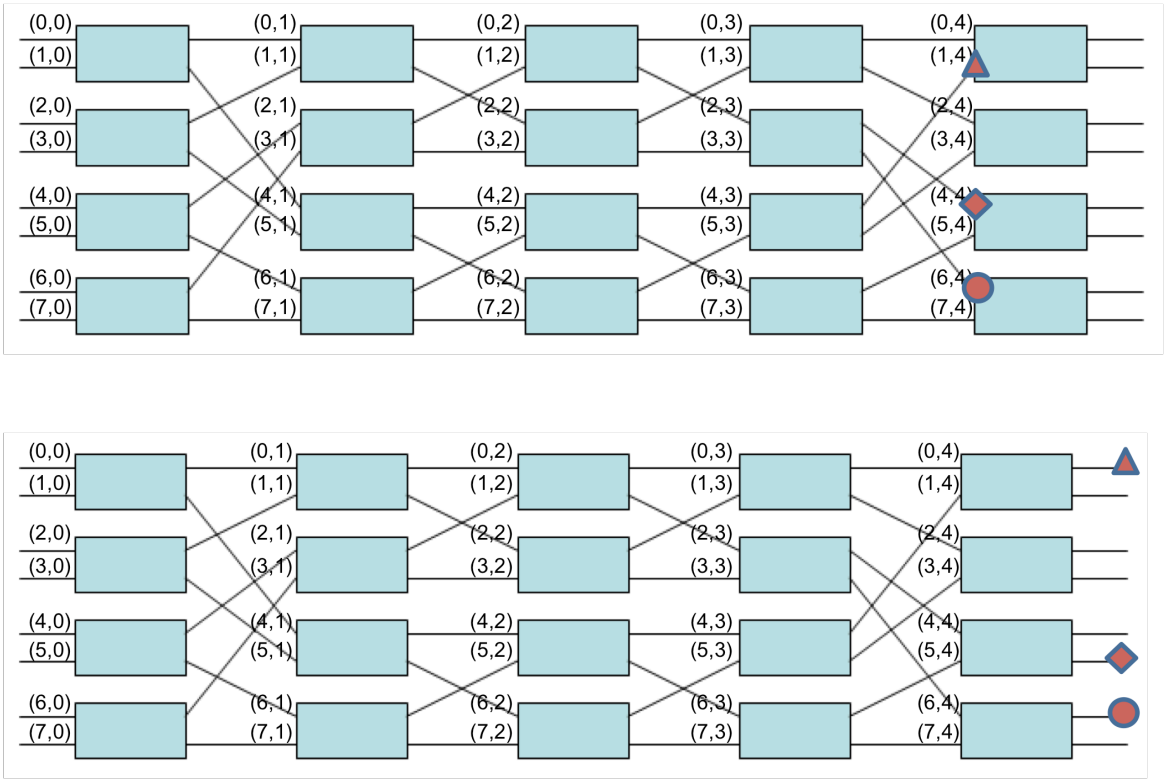


Figure 23: The actual route followed by the data packets in the absence of conflict

4.5 A Modified Buffered Adaptive Routing Algorithm

Modified Buffered Adaptive Routing Algorithm for N×N Benes network:

Index Terms:

a) Current node: the current node is given by : C(position, stage)
b) Destination node: the destination node is represented by : D(position, 0)
c) N = 2k.
d) p ($1 \leq p \leq k$) is an integer.
e) Buffer(x, y): the buffer at x position and y stage

Routing Procedure:

If

local node address(y >= 0 && < k - 1)

then

select any one of the two output ports of the current switching node.

else

If

(local node address y == (2k - p))

{

If

(destination node address x / 2^{p-1} % 2 == 0)

then

select the upper output port of the current switching node

{

If

The upper output port is already occupied

then

move data packet to buffer (current x, current y)

}

else if

The destination node address x/ 2^{p-1}%2==1)

then

select the lower output port of the current switching node.

If

{

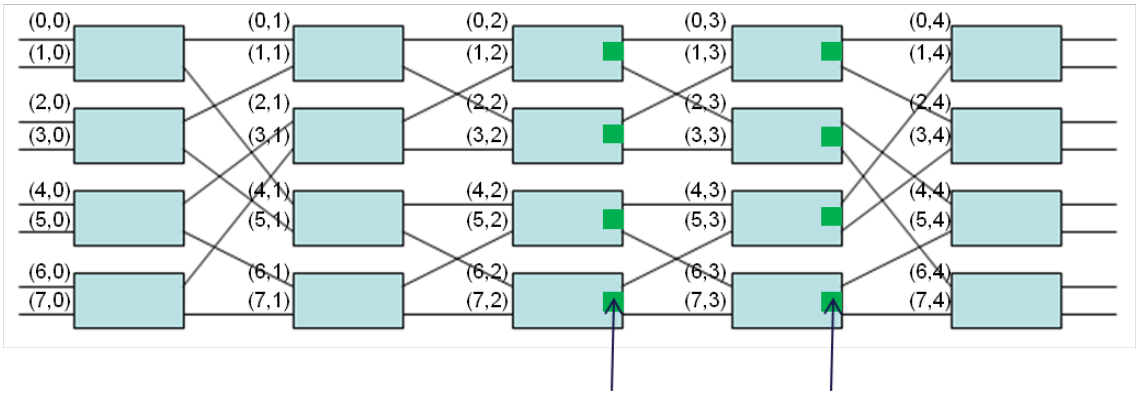
The lower output port is already occupied

then

move data packet to buffer (current x, current y)

}

For all buffers!= NULL at stage y-1, move data packets for each according to step 6. To enable the routing of multiple data packets using the proposed algorithm, there was a need to make some enhancements to the network hardware. This includes an addition of buffers to stage 2 and stage 3 switching nodes as shown in Figure 24 below.



Addition of buffers to stage 2 and 3

Figure 24: New Modified Buffered Approach for routing

Routing done using the modified algorithm is illustrated below (for the same data set as used in the illustration of previous algorithm's limitation.)

Data at (2,0) to be sent to (6,4)

Data at (6,0) to be sent to (5,4)

Data at (7,0) to be sent to (1,4)

1. Initially the data are at (2,0), (6,0), (7,0) as shown in the Figure 25.

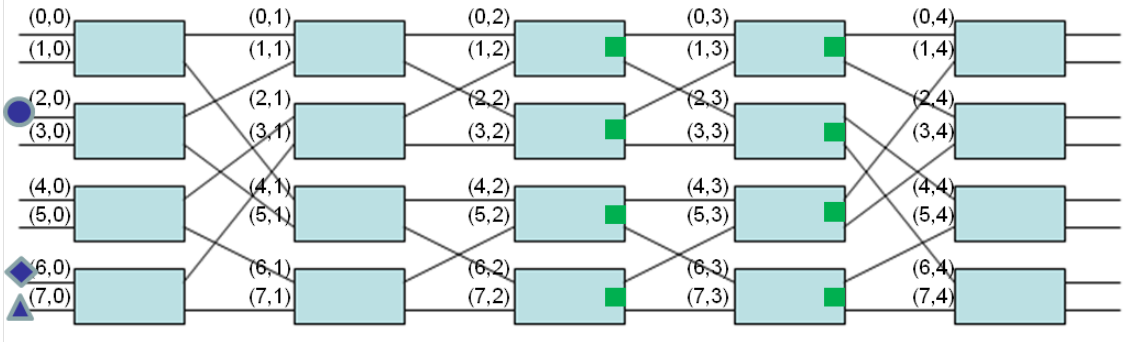


Figure 25: Data at (2,0), (6,0), (7,0)

2. According to the algorithm, the data packets adopt the upper port of the nodes and consequently move to the next stage as shown in the Figure 26.

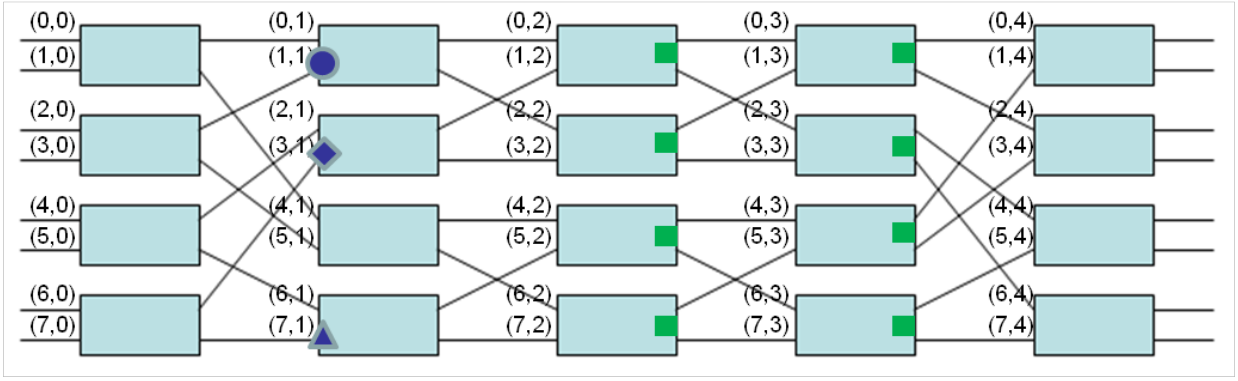


Figure 26: Data at (1,1), (3,1), (7,1)

3. Following the same step again, the data packets are routed onto the next stage as shown in the Figure 27.

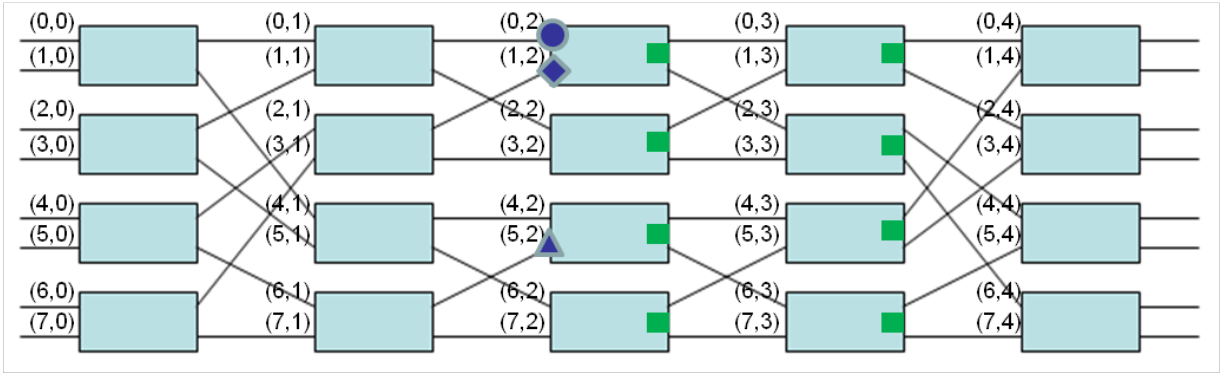


Figure 27: Data at (0,2), (1,2), (5,2)

4. The else part of the algorithm now comes into practice and the packets are routed onto the next stage. According to the algorithm, the first data packet utilizes the lower output port and reaches the next stage at (2,3). The second data packet also wants to use the same path that is already occupied by the first data packet so instead of blocking the network, the routing information of the second data packet waits in the buffer maintained at the second stage till the path becomes free and then moves onto the next stage as shown in the Figure 28 given below.

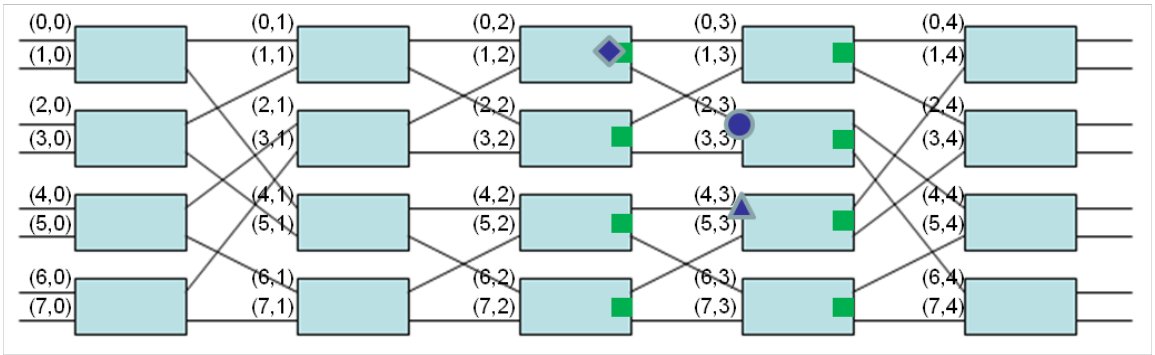


Figure 28: Data at (2,3), buffer(1,2), (4,3)

5. In the next step, the first data packet will move to the next stage and will free the path that it was earlier utilizing. The second data packet now will move to (2,3) position and free the buffer. The third data packet moves to the next stage accordingly as shown in the Figure 29.

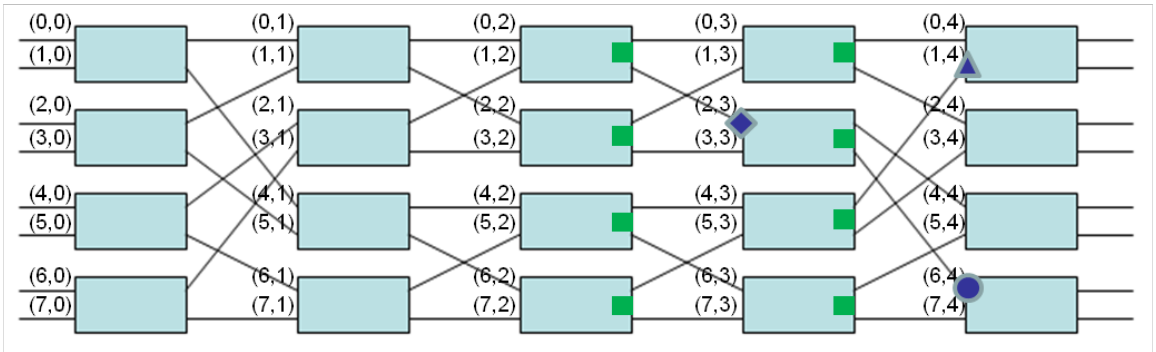


Figure 29: Data at (6,4), (2,3), (1,4)

6. In the next step, both the first and the third data packet will reach to their required destination while the second data packet gets delayed by one cycle as illustrated in the Figure 30 given below.

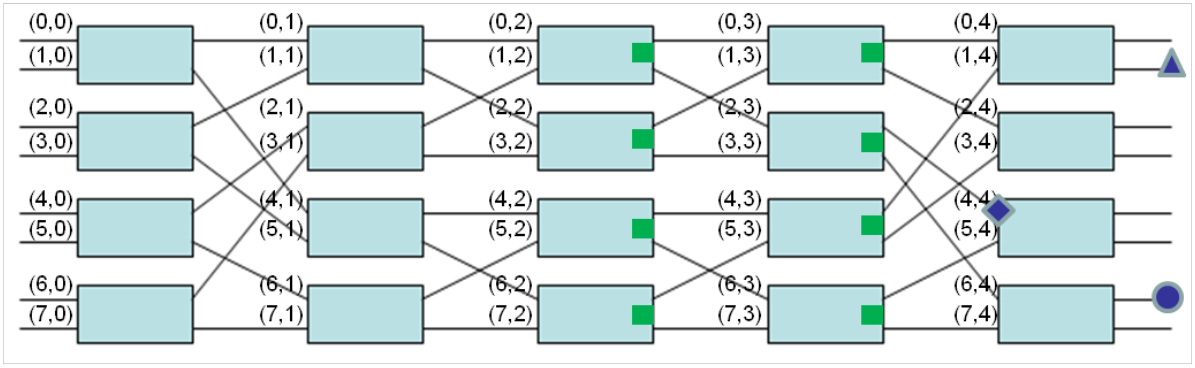


Figure 30: Data at 6, (4,4), 1

7. Finally in the last step, the second data packet also reaches its required destination as shown in the Figure 31.

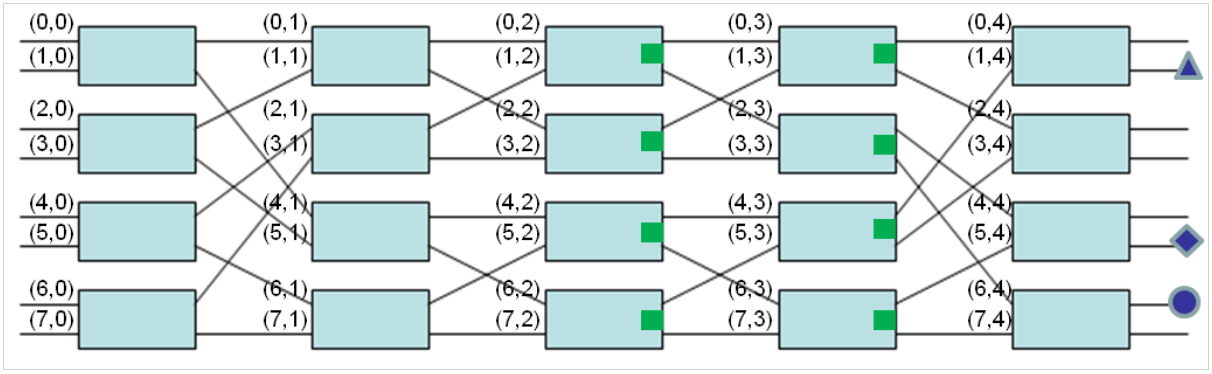


Figure 31: Data at 6, 5, 1

Chapter 5 Experimental Results

5.1. Introduction

This section summarizes the results of the experiment, conducted on the existing and proposed algorithms. Both the routing algorithms enable different paths between any particular pair of source node and the destination node for Benes network architecture, the difference being in the number of inputs realized by the particular algorithm at a particular time. The existing algorithm was suitable for handling only one input at a time thereby affecting the performance of the network. So in order to increase the network performance of the Benes network, a new algorithm is proposed which could handle multiple inputs at a time.

5.2. Benes network architecture

Benes network is a rearrangeable network which enables non-blocking communication between any pair of idle terminals through reassigning the exiting links. Besides, Benes network is also a well-known multistage network which is also called B(k) and can be constructed by adding one stage including N / 2 (N = 2k) switching elements at the left and the right of two copies of B(k-1) respectively. It is composed of 2×2 basic switching elements in a recursively method which could be setup in any two possible states i.e. the bar state or the cross state.

Figure 32 represents the architecture of the Benes network with 8 inputs and 8 outputs, where $8 = 2^k$ (k = 3). Besides, the network consists of $2\log_2 8$-1 stages which come out to be equal to 5 stages starting from 0 to 4. Each stage contains 8/2=4 switching elements. Each switching node is denoted by a set of integer coordinate (position, stage), where position is the spot in one stage and stage is the stage in the network. the position

is numbered from zero to 8/2−1=3 (top to bottom) and the stage is numbered from zero to 2k −1 = 4 (left to right). The input nodes which connect with the first stage and the output nodes which connect with the last stage are not in the stage ordering.

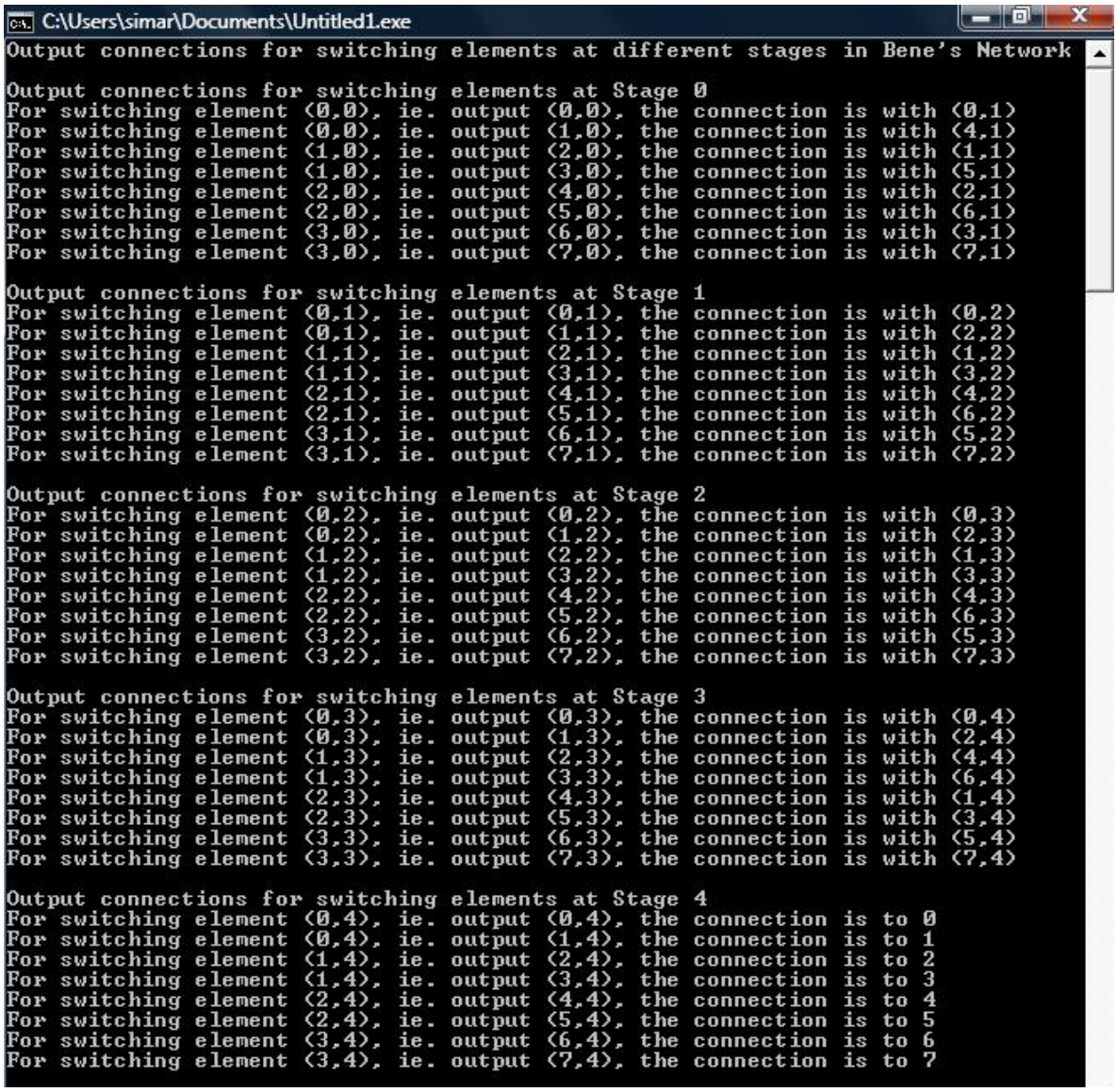


Figure 32: The basic connections of switching elements in a 8×8 Benes Network

5.3. Partially Adaptive Routing Algorithm for Benes Network

The partially adaptive routing algorithm aims to seek a path that packets travel between any pair of source and destination nodes. This algorithm however could

remodify routes dynamically according to the present conditions of the network in contrast to the deterministic routing algorithms. Hence it reduces the possibility of congestion effectively.

But the major limitation with this method was that it only worked with one input at a time and was not suitable for working with multiple inputs at a time which means that when multiple requests were made, there occurred a routing conflict which caused a strong impact on the performance of the network as the routing algorithms play a very important role in the performance of the Benes network.

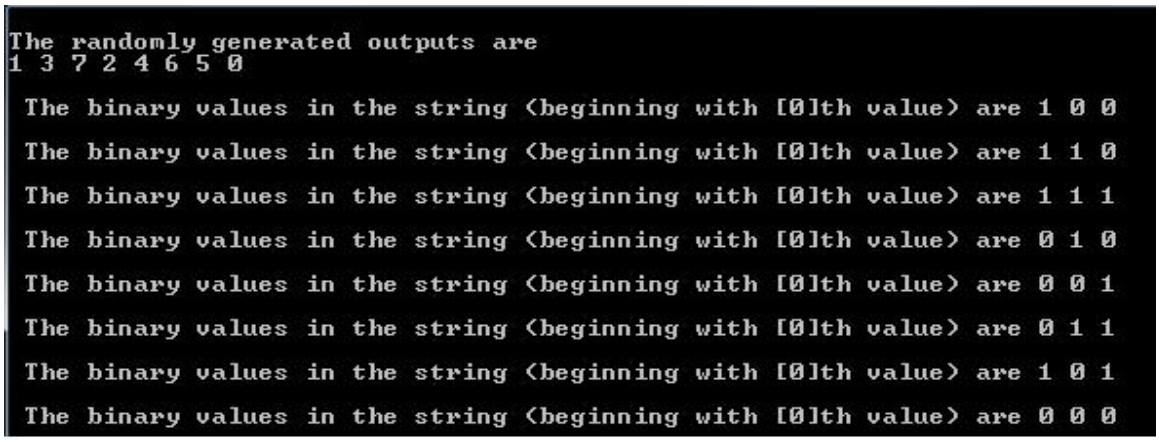


Figure 33: A randomly generated permutation for routing in a Benes Network

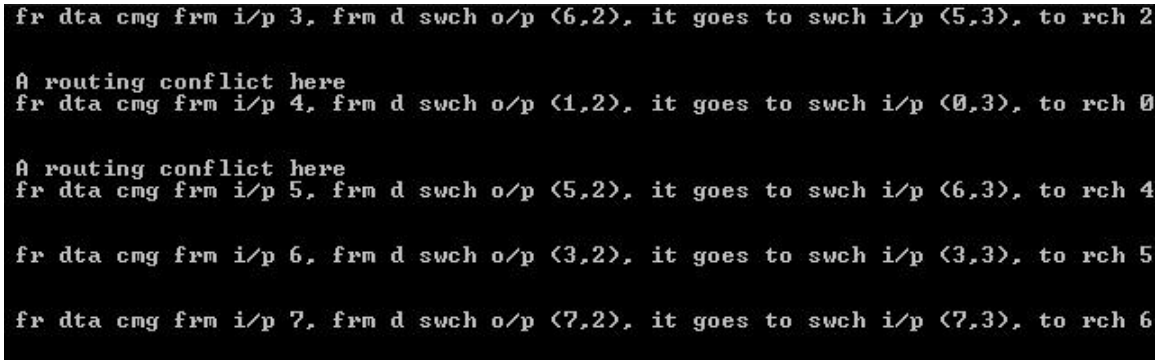


Figure 34: Routing conflict due to Partially Adaptive Routing algorithm's limitation

5.4 Modified Buffered Adaptive Routing Algorithm

The proposed algorithm uses the concept of buffers in order to remove the limitation of the existing algorithm so that multiple inputs are routed simultaneously. This means that whenever two inputs try to route through the same path and there occurs a routing conflict, then this can be avoided by allowing only one input at a time to use the path and storing the other input information in the buffer maintained at the switching node of a particular stage at which the conflict arises. With this approach, multiple inputs can be routed at the same time and thus can be helpful in improving the performance of the network.

Figure 35 shows how the packets are routed between the source and the destination pairs and also explains the concept of buffers used at different stages in the network. A randomly generated output permutation is considered and the paths are defined that the packets travel between the input and the output pairs according to the algorithm. In case of any routing conflict, the routing information is stored in the buffers for temporary basis so that a path is available for a particular input at a particular time and at the same times multiple inputs are realized.

```
The data are at stage 0
for data cmg from input 0, frm d swch o/p (0,0), it goes to swch i/p (0,1)
for data cmg from input 1, frm d swch o/p (1,0), it goes to swch i/p (4,1)
for data cmg from input 2, frm d swch o/p (2,0), it goes to swch i/p (1,1)
for data cmg from input 3, frm d swch o/p (3,0), it goes to swch i/p (5,1)
for data cmg from input 4, frm d swch o/p (4,0), it goes to swch i/p (2,1)
for data cmg from input 5, frm d swch o/p (5,0), it goes to swch i/p (6,1)
for data cmg from input 6, frm d swch o/p (6,0), it goes to swch i/p (3,1)
for data cmg from input 7, frm d swch o/p (7,0), it goes to swch i/p (7,1)
```

Figure 35: The routing of data packets at stage 0 of Benes Network

```
The data are at stage 1
for data cmg from input 0, frm d swch o/p (0,1), it goes to swch i/p (0,2)
for data cmg from input 1, frm d swch o/p (4,1), it goes to swch i/p (4,2)
for data cmg from input 2, frm d swch o/p (1,1), it goes to swch i/p (2,2)
for data cmg from input 3, frm d swch o/p (5,1), it goes to swch i/p (6,2)
for data cmg from input 4, frm d swch o/p (2,1), it goes to swch i/p (1,2)
for data cmg from input 5, frm d swch o/p (6,1), it goes to swch i/p (5,2)
for data cmg from input 6, frm d swch o/p (3,1), it goes to swch i/p (3,2)
for data cmg from input 7, frm d swch o/p (7,1), it goes to swch i/p (7,2)
```

Figure 36: The routing of packets at stage 1 of Benes Network

```
The data are at stage 2
fr dta cmg frm i/p 0, frm d swch o/p (0,2), it goes to swch i/p (0,3), to rch 1
fr dta cmg frm i/p 1, frm d swch o/p (4,2), it goes to swch i/p (4,3), to rch 3
fr dta cmg frm i/p 2, frm d swch o/p (2,2), it goes to swch i/p (3,3), to rch 7
fr dta cmg frm i/p 3, frm d swch o/p (6,2), it goes to swch i/p (5,3), to rch 2
fr dta cmg frm i/p 4, frm d swch o/p (1,2), it goes to swch i/p (2,3), to rch 4
fr dta cmg frm i/p 5, frm d swch o/p (5,2), it goes to swch i/p (6,3), to rch 6
A routing conflict, dis data wil move to the buffer in stage 3
fr dta cmg frm i/p 6, frm d swch o/p (3,2), it goes to swch i/p (3,3), to rch 5
A routing conflict, dis data wil move to the buffer in stage 3
fr dta cmg frm i/p 7, frm d swch o/p (7,2), it goes to swch i/p (5,3), to rch 0
```

Figure 37: The routing of packets at stage 2 of Benes Network

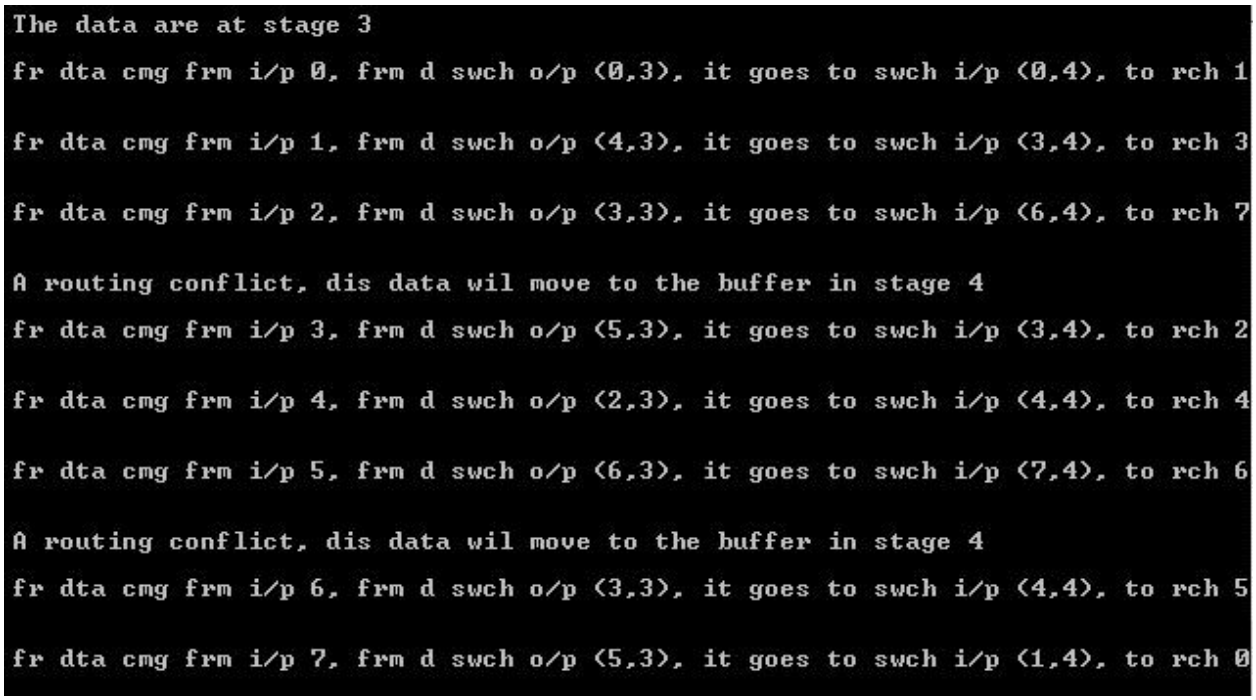


Figure 38: The routing of packets at stage 3 of Benes Network

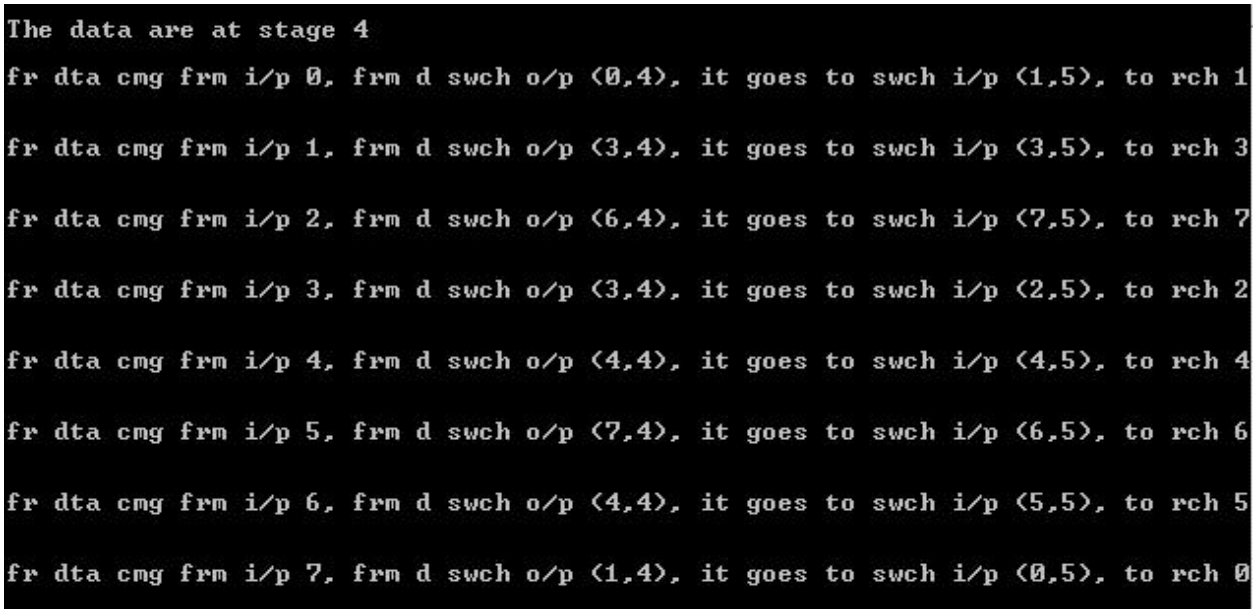


Figure 39: The routing of packets at stage 4 of Benes Network

Chapter 6 Conclusion

6.1 Conclusion

1. The strictly non blocking conditions for the networks turn the networks into non blocking ones but incur a lot of cost so there occurs a need for rearrangeable networks (Benes network) and different types of routing algorithms are devised for them.
2. The major problem with the existing routing algorithm for the Benes network was the occurrence of the routing conflict whenever multiple inputs were needed to be realized at a single instant of time.
3. A new modified buffered algorithm has been developed to solve this problem that is capable of handling multiple inputs simultaneously and thus removing the occurrence of any type of routing conflict that could cause an adverse effect on the performance of the network.

6.2 Summary of Contributions

1. The routing procedure of existing partially adaptive routing algorithm for Benes Network has been simulated.
2. The problem of the routing conflict arising in the existing algorithm has been practically implemented.
3. A new modified routing algorithm has been proposed and developed for the Benes network.
4. A gap analysis between the existing and the proposed algorithm has been done and practically implemented.

6.3 Future Research

1. Despite the fact that the rearrangeable networks cause a very less hardware cost as compared to the strictly non-blocking networks but still a delay problem occurs before a given request is routed in case of rearrangeable networks. This delay problem can be worked upon and removed.
2. The new modified buffered routing algorithm although removed the problem of the routing conflicts arising when multiple inputs were routed simultaneously but still it caused a delay as the data packets had to wait in the buffer whenever the path was used up by the other data packet. This time delay can be evaluated.
3. The comparison can be made between the existing and the proposed algorithm on the basis of time delay.

References

[1] Abed F. and Othman M., "Efficient Window Method in Optical Multistage Interconnection Networks", in *Proceedings of IEEE International Conference on Networks*, 2007, pp. 181-185.

[2] Busi I. and Pattavina A., "Strict-Sense Non-Blocking Conditions for Shuffle Exchange Networks with Vertical Replication", in *Proceedings of IEEE INFOCOM, The Conference on Computer Communications*, vol.1, no.7, pp. 126-133, 1998.

[3] Chakrabarty A., Collier M. and Mukhopadhyay S., "Dynamic Path Selection Algorithm for Benes Networks" in *First International Conference on Computational Intelligence*, *Communication Systems and Networks*, 2009, pp. 23-28.

[4] Chakrabarty A., Collier M. and Mukhopadhyay S., "Matrix-Based Nonblocking Routing Algorithm for Benes Networks", in *Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns,* 2009, pp. 551-556.

[5] Clos C., "A study of Non-Blocking Switching Networks", *Bell Systems Technical Journal*, vol. 32, no.4, pp. 406-424, 1953.

[6] Das N., Mukhopadhyaya K. and Dattagupta J., "O(n) routing in Rearrangeable Networks", *Journal of Systems Architecture*, vol. 46, no. 5, pp. 529-542, 2000.

[7] Hasan C., "Rearrangeability of (2n − 1)-Stage Shuffle-Exchange Networks," *Society for Industrial and Applied Mathematics,* vol. 32, no. 3, pp. 557-585, 2003.

[8] Hinton H., "A Non-Blocking Optical Interconnection Network using Directional Couplers", in *Proceedings of IEEE, Global Telecommunications Conference*, 1984, pp. 885-889.

[9] Hwang F.K., "Choosing the Best $\log_k$(N,m,P) Strictly Non blocking Networks", *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 1045-1050, April 1998.

[10] Jiang X., Shen H., Khandker M. and Horiguchi S., "Blocking Behaviors of Crosstalk-free Optical Banyan Networks on Vertical Stacking," *IEEE Transactions on Networking*, vol. 11, no. 6, pp. 982-993, December 2003.

[11] Jiang X., Shen H. and Horiguchi S., "Blocking Probability of Vertically Stacked Optical Banyan Networks Under Random Routing," *in Proceedings of IEEE GLOBECOM*, vol.6, no.5, pp. 876-943, 2003.

[12] Kaixin R., Naijie G.U., "Permutation Capability of Optical Cantor Network", in *Proceedings of 8th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2007, pp.398-404.

[13] Kim M.K. and Maeng S.R., "On the Correctness of Inside-Out Routing Algorithm", *IEEE Transactions on Computers*, vol. 46, no. 7, pp. 820-823, July 1997.

[14] Lea C.T. and Shyy D.J., "Tradeoff of Horizontal Decomposition Versus Vertical Stacking in Rearrangeable Nonblocking Networks", *IEEE Transactions on Communications,* vol. 39, no. 6, pp. 899-904, June 1991.

[15] Lea, C.T. and Shyy, D.J., "Log_2 (N,m,p) Strictly Nonblocking Networks", *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1502-151, 1991.

[16] Lee C.Y. and Yavuz O.A., "A Fast Parallel Algorithm for Routing Unicast Assignments in Benes Networks" *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 3, pp. 314-328, March 1995.

[17] Lu E. and Zheng S.Q., "Parallel Routing Algorithms for non Blocking Electronic and Photonic Switching Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no.8, pp. 562-572, 2005.

[18] Lu E. and Zheng S.Q., "High-Speed Crosstalk-Free Routing for Optical Multistage Interconnection Networks", in *IEEE International Conference on Computer Communications*, 2003, pp. 249-254.

[19] Nassimi D. and Sahni S., "A Self-Routing Benes Network and Parallel Permutation Algorithms", *IEEE Transactions on Computers*, vol. C-30, no. 5, pp. 332-340, May 1981.

[20] Nassimi D. and Sahni S., "Parallel algorithm to set up the Benes permutation network", *IEEE Transactions on Computers*, vol. 31, no. 2, pp. 148-154, 1982.

[21] Opferman D.C. and Tsao-Wu N.T., "On a Class of Rearrangeable Switching Networks", *The Bell System Technical Journal*, vol. 50, no. 5, pp. 112-120, 1981.

[22] Othman M. and Abedi F., "Fast Method to find Conflicts in Optical Multistage Interconnection Networks", *International Journal of the Computer, the Internet and the Management* , vol. 16, no. 5, pp. 18-25, April 2008.

[23] Pan Y., Qiao C. and Yang Y., "Optical Multistage Interconnection Networks: New challenges and approaches", *IEEE Communications Magazine*, vol. 37, no. 2, pp. 50-56, Feb. 1999.

[24] Qiao C., Melhem R., Chiarulli D. and Levitan S., "A Time Domain Approach for avoiding Crosstalk in Optical Blocking Multistage Interconnection Networks", *Journal of Light*wave *Technology*, vol.12, no. 10, pp. 1854-1862, 1994.

[25] Raghavendra C.S. and Boppana R.V., "On Self-routing in Benes and Shuffle Exchange Networks", *IEEE Transactions on Computers*, vol. 40, no. 9, pp. 1056-1064, 1991.

[26] Rajgopal K., "The KR-Benes Network: A Control optimal Rearrangeable Permutation Network", *IEEE Transactions on Computers*, vol. 54, no. 5, pp. 534-544, May 2005.

[27] Regis B.J., "Optical Switching and Networking Handbook", McGraw-Hill, New York, 2001.

[28] Sawchuck A.A., Jenkins B.K., Raghavendra C.S. and Varma A., "Optical Crossbar Networks," *IEEE Computer Society*, vol. 20, no. 6, pp. 50-60, June 1987.

[29] Seo S.W. and Feng T., "The Composite Banyan Network", *IEEE Transactions on Parallel and Distributed systems*, vol. 6, no. 10, pp. 1043-1054, October 1995.

[30] Vaez M.M. and Lea C.T., "Strictly Nonblocking Directional-Coupler-Based Switching Networks under Crosstalk Constraint," *IEEE Transactions on Communications,* vol. 48, no. 2, pp. 316-323, February. 2000.

[31] Wang J., Pan Y., "Permutation Capability of Optical Multistage Interconnection Networks", in *Proceedings of International Symposium on Parallel and Distributed Parallel processing* , April 1998, pp. 21-28.

[32] Wu C.L. and Feng T.Y., "On a class of Multistage Interconnection Networks", *IEEE Transactions on Computers*, vol. C-29, no. 8, pp. 694-702, August 1980.

[33] Yadav R. and Aggarwal R.R., "Survey and Comparison of Optical Switch Fabrication Techniques and Architectures", *Journal of Computing*, vol. 2, no. 4, pp. 2151-9617, 2010.

[34] Yeh Y.M. and Feng T., “On a class of Rearrangeable networks”, *IEEE Transactions on Computing*, vol. 41, no. 9, pp.1361-1379, Nov. 1992.

[35] Yoon K., “A New Benes Network Control Algorithm”, *IEEE Transactions on Computers,* vol. C-36, no. 6, pp. 768-772, 1987.

[36] Yu C., Jiang X.H., Horiguchi S. and Guo M., “Overall Blocking Behavior Analysis of General Banyan-based Optical Switching Networks”, *IEEE Transactions on Parallel and Distributed Systems,* vol. 17, no. 9, pp. 1037-1046, 1990.

[37] Zadedyurina O., Ofek O. and Pattavina A., “Space and Time Blocking versus Cost in All-Optical Banyan Networks”, in *Proceedings of IEEE Conference on Networks*, 2008, pp. 5338-5343.

[38] Zhang J. and Gu H., “Partially Adaptive Routing Algorithm for Benes Network on Chip”, in *Proceedings of IEEE Conference on Networks*, 2009, pp. 4244-4520.

Paper Published

Kaur T. and Aggarwal R.R., "Blocking Behavior Analysis of Optical Switching Networks", International Journal of Engineering Science and Technology, vol. 3, no. 2, pp. 1542-1550, Feb. 2011.