A NEW BUFFERED FABRICATION TECHNIQUE FOR AUGMENTED DATA VORTEX SWITCH

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

Master of Engineering

in

Software Engineering

Submitted By:

Ravinder Yadav

(800831010)

Under the supervision of:

Ms. Rinkle Rani Aggarwal
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2010

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"A new Buffered fabrication technique for Augmented Data Vortex switch"**, 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 Ms. Rinkle Rani and refers other researcher's works which are duly listed in the reference section.

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

(Ravinder Yadav)

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

(Ms. Rinkle Rani)
Assistant Professor,
Computer Science and Engineering Department,
Thapar University,
Patiala.

Countersigned by

05/07/2010

(DR. RAJESH BHATIA)
Head,
Computer Science & Engineering, Department ,
Thapar University,
Patiala.

16.7.10

(DR. R.K. SHARMA)
Dean (Academic Affairs),
Thapar Uinversity,
Patiala.

Acknowledgement

Words are often too less to reveals ones deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to Express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis.

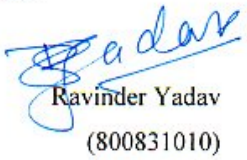
First of all I would like to thank the all mighty god, one who has always guided me to work on the right path of the life. Without Her grace this would never come to be today's reality.

This work would not have been possible without the encouragement and able guidance of my supervisor, Ms. Rinkle Rani, Assistant Professor, Computer Science and Engineering Department, for her enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Her feedback and editorial comments were also invaluable for the writing of this thesis.

I am also thankful to Dr. Rajesh Bhatia, Head, Computer Science and Engineering Department and Dr. Inderveer Channa, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

No words of thanks are enough for my Parents whose support and care makes me stay on earth.

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



Ravinder Yadav
(800831010)

Abstract

Future is the time of high bandwidth communication. Even today, the present technology is not sufficient to cope up with the tremendous demand of bandwidth. Photonic transfer is one of the technologies, which can satisfy the bandwidth requirement as per the current technologies, where electro-optic switches are used in conjunction with the optical media for switching the data packets. In this regard, a switch architecture already has been proposed named as Augmented Data Vortex (ADV) is available. While designing the switch architecture, attention must be paid towards the time delay in delivering the packets during peak load hours. But in this switch architecture less attention has been paid to the peak loads hours time delay. In this thesis a new switch architecture named BADV (Buffered augmented data vortex) has been proposed. The proposed BADV architecture uses buffer to reduce time delay during peak load hours. The performance of BADV has been compared with ADV on the basis of time delay. The result has been shown the proposed architecture improves the performance in terms of time delay.

Contents

List of Figures

Figure **Page No.**

List of Tables

Table **Page No.**

Chapter 1 Introduction

Multistage interconnection networks (MINs) consist of more than one stages of small interconnection elements, called switching elements and links interconnecting them. Multistage networks are described by three characteristic features switching elements, network topology and control structure [1]. A multistage interconnection network is actually a compromise between crossbar and shared bus networks, as shown in the following Table 1, describing the properties of various types of multiprocessor interconnections.

Table 1: Properties of multiprocessor interconnections [2]

Property	**Bus**	**Crossbar**	**Multistage**
Speed	Low	High	High
Cost	Low	High	Moderate
Reliability	Low	High	High
Configurability	High	Low	Moderate
Complexity	Low	High	Moderate

Multistage interconnection networks:

- Attempt to reduce cost.
- Attempt to decrease diameter.

In a multistage interconnection network, as in a crossbar, switching elements are distinct from processors. However, fewer than $O(P^2)$ switches are used to connect P processors. Instead messages pass through a series of switch stages.

Figure 1 illustrates two MINs, which are representatives of a general class of networks. Black circles represent processors and white circles represent crossbar switches. The network can be constructed from unidirectional switches and links, in which case it is folded so that the processors on the left and right are the same. Alternatively, it can be constructed from bi-directional switches and links, in which case processors on the left and right are distinct.

In a unidirectional MIN, all messages must traverse the same number of wires, and so the cost of sending a message is independent of processor location. In effect, all processors are equidistant.

In a bi-directional MIN, the number of wires traversed depends to some extent on processor location, although to a lesser extent than a mesh or hypercube.

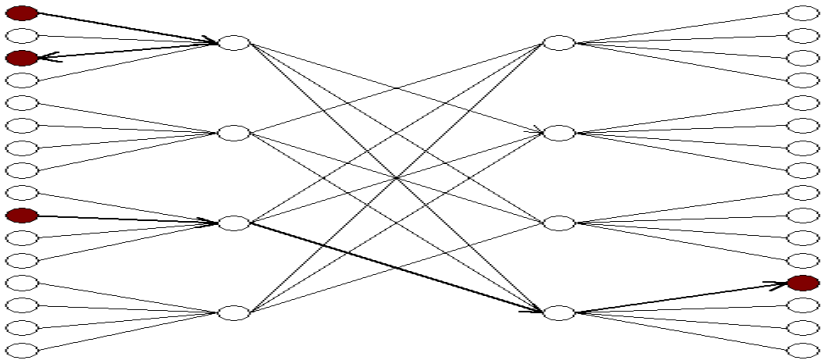


Figure 1: Multistage interconnection networks [4]

1.1. Switching Elements

The switching element may be viewed as a very small network. These switches are the devices having multiple inputs/outputs [10]. The number of inputs/outputs and the input-to-output connections supported within a switch can assume either the straight or the exchange states. A four-function switch box can be in any one of the following four states i.e. straight, exchange, upper broadcast and lower broadcast as shown in Figure 2.

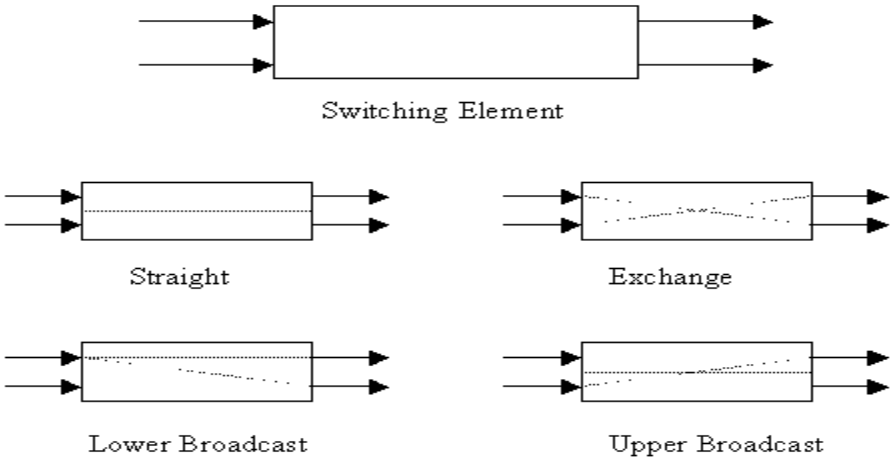


Figure 2: Switching elements [4]

1.2. Classification of Multistage interconnection networks

Multistage interconnection networks can be classified according to different categories. Major classification categories are as follows.

1.2.1. According to number of paths

a.) **Unique path networks:** These networks provide unique path between every source and destination. The failure of any switching element along the path disconnects some source-destination pairs, so adversely affecting the capabilities of existing network.

b.) **Multi path networks:** These provide more than one path between source and destination. In case, there is a failure of one switching element in the path, the request is routed through some alternative path. Multi path multistage interconnection networks can be either static or dynamic. For static networks, if a fault is encountered, then data has to backtrack, the source or some fixed point to select an alternative path in the network. The implementation of backtracking is expensive in terms of the hardware. In dynamic networks, if a fault is encountered in a particular stage, a switching element in preceding stage will re-route data through an alternative available path.

1.2.2. According to number of switches

a.) **Regular networks:** Regular multistage interconnection networks have an equal number of switching elements per stage; as a result they may impose equal time delay to all requests passing through them.

b.) **Irregular networks Irregular:** multistage interconnection networks have unequal number of switching elements as per stage and thus they are inherently multi path in nature. For a given source destination pair, there are different path lengths available.

1.2.3. According to controls

a.) Flip controlled networks: Flip controlled multistage interconnection networks have a common control signal for switching in various switching elements at a given stage. This network is less complicated due to lesser number of control signals but has lesser bandwidth.

b.) Distributed control networks: Distributed control multistage interconnection networks have a separate control signal for every switching element. These have higher bandwidth due to selection of source destination pair at a given time and are quite complex.

1.2.4. According to availability of paths

a.) **Blocking networks:** In blocking network, simultaneous connections of more than one terminal may result a confliction in the use of network communication links. For example Omega network.

b.) Non blocking networks: A network is called none blocking if 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. In other words, a network that can handle all possible connections without blocking is called non-blocking network.

1.3. Optical Multistage Interconnection Networks

Optical Multistage Interconnection networks [OMINs] are popular in switching and communication application. By researcher foremost problems in OMIN is cross talk. Various methods are suggested for avoiding the cross talk. Two major approaches in solving cross talk are Space domain and Time domain [3].

1.3.1. Space Domain

In the space domain approach we use 2Nx2N regular OMIN to provide NxN connections. But half of the input and output are wasted in this approach.

1.3.2. Time Domain

In the time domain approach is to route all the input in several groups i.e. there would be no cross talk between messages in each groups. To find out which message should be kept in which group to avoid the cross talk, a method is used called Window Method and ZeroX [8].

a.) Window Method: It is a technique to find out which message should be kept in which group to avoid cross talk [3],[3]. It can be describe as follows.

For a network size N*N there are N source and N destination address. Each source and destination address is combined to produce a combination matrix, the optical window size is M-1 where $M=\log_2 N$ is the size of network. This window is used in the combination matrix from left to right except first and last column. If the two messages have the same bit pattern will cause the conflict in the network. Hence they must be routed in the different passes. For example: let 8x8 switch is having following permutations.

Table 2: Source and destination address [3]

Src	Dest
000	101
001	001
010	011
011	110
100	000
101	010
110	100
111	111

Now the window size is M-1=2 where $M=\log_2 8=3$ and the number of windows is M=3 say W_0, W_1, W_2.

Step-1 (W_0)

Table 3: Window (W_0) [3]

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

Step-2 (W_1)

Table 4: Window (W_1) [3]

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

In this find the possibilities of confliction are found and arranged them in different groups so as to avoid conflicts. In the window method limitation is the total number of windows required i. e. $M=\log_2 N$.

Step-3 (W_2)

Table 5: Window (W_2) [3]

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

b. ZeroX algorithms: In this method we calculate the conflict matrix at first, then follow the following step for a given conflict matrix [5].

Step1: Compute the sum of all column of matrix M_{ij} and store it in the row N+1.

Step2: Select all the zero entries in the row N+1 and at the matrix M_{ij} and cluster them in the group G.

Step3: Add to group G all column, which are intersected, with all entries in G by zero.

Step4: Check if the group current group G consists of only one entry equal zero then add the successor entry equal to zero in the same row or the G entry.

Step5: Check if there is any other entry in matrix Mij the intersection with current group G equal zero then add it to the current group G.

Step6: Initialize all columns and rows corresponding to all entries in group G to zero.

Step7: Repeat all steps from 1 to 6 until all the entries become zero in conflict matrix.

For example: let 8x8 switch is having following permutations.

Table 6: Source and destination address [5]

Src	Dest
000	101
001	001
010	011
011	110
100	000
101	010
110	100
111	111

The conflict matrix is as follows

Table 7: Conflict matrix [5]

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

After first step we will the sum of all the 8 rows is represented in row 9 as shown in Table 8.

Table 8: Summation in ZeroX [5]

Message	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
000	0	0	0	0	1	1	1	0
001	0	0	0	1	1	1	0	0
010	0	0	0	1	1	0	1	0
011	0	0	0	0	0	0	0	1
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	1
110	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	0	0
Summation	0	0	0	2	3	2	2	3

After the execution of step2, results obtained are G = {000,001,010}. In the next step G will add one more entry i.e. {111}, now G = {000,001,010,111}. In the next step all the entry will be Zero.

Table 9: ZeroX output [5]

Message	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
000	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0
010	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0
Summation	0	0	0	0	0	0	0	0

From the Table 9 there is another group defined as G2, which include elements {011,100,101,100}. These groups are called passes.

1.4. Key issue in the designing of switching

The switching elements are the very sensitive component of the network. While designing the switching elements, attention must be paid toward performance and fault tolerance. In optical switch frailer in one small unit can leads to frailer in whole network. Following are the major issues which should be taken care while designing the switch architecture.

a.) **Optical cross-connects:** A very important application of optical switches is the provisioning of light paths. A light path is a connection between two network nodes that is set up by assigning a dedicated wavelength to it on its link in its path [6]. OXC are meant to provide a light path that connects two networks nodes [7], [9]. In this application, the switches are used inside optical cross-connects (OXCs) to reconfigure them to support new light paths. OXC's are the basic elements for routing optical signals in an. Optical switch requirements for OXC's includes [6].

   a.) Scalability.
   b.) High port counts switches.
   c.) The ability to switch with high reliability.
   d.) The ability to switch to a specific optical path without disrupting the other optical paths.

b.) **Protection switching:** Optical protection switching is the completion of the transmission in the event of system or network level errors. Optical protection switching requires optical switches with smaller port counts i.e. 1x2 or 2x2. Protection switching requires switches must be extremely reliable, since sometimes these switches are single point of failure in the network [6].

c.) **Optical Add/Drop multiplexing:** Optical add/drop multiplexers (OADMs) residing in network nodes insert /extract (add/drop) optical channels (wavelengths) to or from the optical transmission stream. Using an OADM, channels in a multi wavelength signal can be added or dropped without any electronic processing. Switches that function as OADMs are wavelength-

selective switches, i.e., they can switch the input signals according to their wavelengths.

d.) **Optical signal monitoring:** Optical signal monitoring is an important network management operation. It receives a small optically tapped portion of the aggregated WDM signal, separates the tapped signal into its individual wavelengths, and monitors each channel's optical spectra for wavelength accuracy, optical power levels, and optical cross talk.

e.) **Network provisioning:** Network provisioning occurs when new data routes have to be established or existing routes need to be modified. A network switch should carry out reconfiguration requests over time intervals. High-capacity reconfigurable switches that can respond automatically and quickly to service requests can increase network flexibility, and thus bandwidth.

1.5. Performance parameters

In the effort to extend optics from transmission to switching, all-optical switching fabrics play a central role. These devices allow switching directly in the optical domain, avoiding the need for several Opto-Electric and Electro-Optic conversions. The most important parameter of a switch is the switching time. Different applications have different switching time requirements.

Other important parameters of a switch follow.

a.) **Insertion loss:** This is the fraction of signal power that is lost because of the switch. This loss is usually measured in decibels and must be as small as possible. In addition, the insertion loss of a switch should be about the same for all input–output connections (loss uniformity).

b.) **Cross talk:** This is the ratio of the power at a specific output from the desired input to the power from all other inputs.

c.) **Extinction ratio (ON–OFF switches):** This is the ratio of the output power in the on-state to the output power in the off-state. This ratio should be as large as possible.

d.) **Polarization-dependent loss (PDL):** If the loss of the switch is not equal for both states of polarization of the optical signal, the switch is said to have polarization-dependent loss. It is desirable that optical switches have low PDL.

1.6. Switch fabrication techniques

Switching speeds of electronics switches is much-2 lesser than the transmission capacity offered by optics [6]. Optical system can send 10 to 100 wavelengths per fiber with each wavelength modulated at 10Gbps or more [9]. So when switching is required; there is a need to convert Optical Signal to Electrical and vice of versa, to overcome this mismatch problem, there is a requirement to use the optical switches so that bandwidth can be used efficiently. The optical packet switching provides an almost arbitrary fine granularity but faces significant challenges in the processing and buffering of bits at high speeds [6].

1.6.1. Opto-mechanical switches

Opto-mechanical technology was the first commercially available for optical switching. In opto-mechanical switches, the switching function is performed by some mechanical means. These mechanical mean include prisms, mirrors, and directional couplers. Mechanical switches exhibit low insertion losses, low polarization-dependent loss, low cross talk, and low fabrication cost. Their switching speeds are in the order of a few milliseconds, which may not be acceptable for some types of applications. Another disadvantage is the lack of scalability. As with most mechanical components, long-term reliability is also of some concern. Opto-mechanical switch configurations are limited to 1x2 and 2x2 port sizes. Larger port counts can only be obtained by combining several 1x2 and 2x2 switches, but this increases cost and degrades performance. Major applications of these switches are in fiber protection and very-low-port-count wavelengths add/drop applications.

1.6.2. Micro-electro-mechanical System Devices

Micro-electro-mechanical system (MEMS) devices can be considered as a subcategory of opto-mechanical switches [6], they are presented separately, mainly because of the great interest in telecommunications industry has shown in them, and also because of the differences in performance compared with other Opto-mechanical switches.

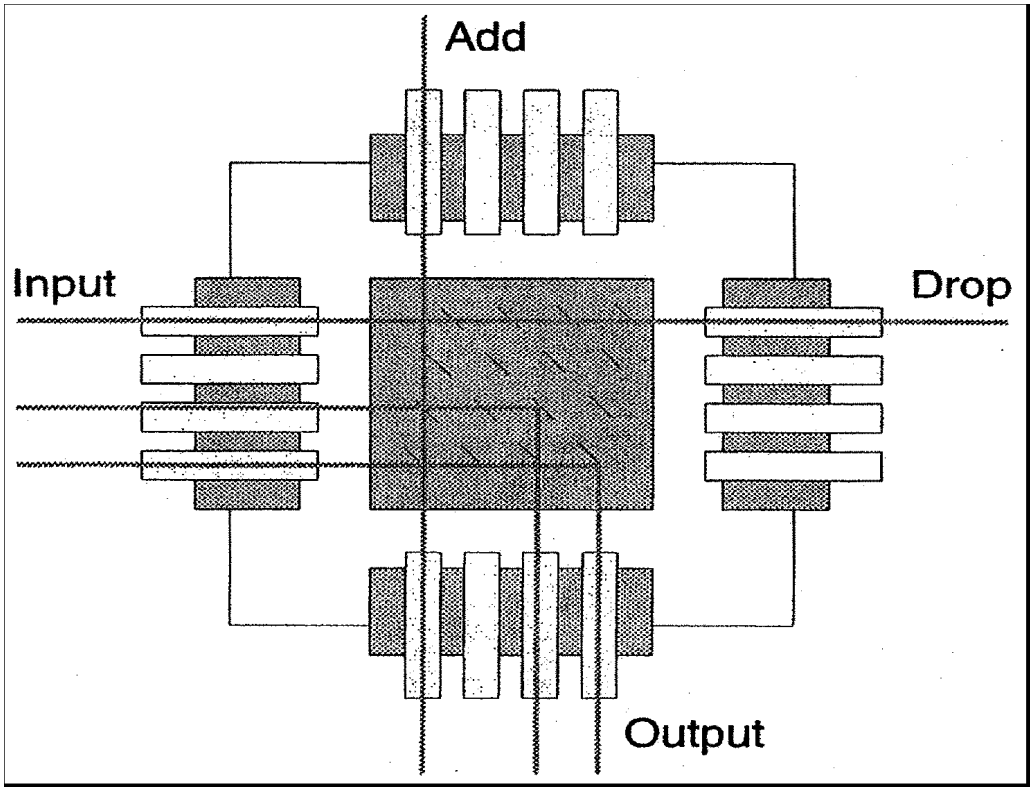


Figure 3: 2D-MEMS Technology [6]

MEMS use tiny reflective surfaces to redirect the light beams to a desired port by either ricocheting the light off of neighboring reflective surfaces to a port or by steering the light beam directly to a port. Figure 3 shows a top view of a 2D MEMS device with the microscopic mirrors arranged in a crossbar configuration to obtain cross-connect functionality. Collimated light beams propagate parallel to the substrate plane. When a mirror is activated, it moves into the path of the beam and directs the light to one of the outputs, since it makes a 45 angle with the beam. This arrangement also allows light to be passed through the matrix without hitting a mirror. This additional functionality can be used for adding or dropping optical channels (wavelengths). The tradeoff for the

simplicity of the mirror control in a 2D MEMS switch is optical loss, while the path length grows linearly with the technology. Number of ports, the optical loss grows rapidly. Commercially available products feature a maximum insertion loss of 3.7 dB for an 8x8 switch, 5.5 dB for 16x16, and 7.0 for 32x32. Therefore, 2D architectures are found to be impractical beyond 32-input and 32-output ports. While multiple stages of 32x32 switches can theoretically form a 1000-port switch, high optical losses also make such an implementation impractical. High optical losses could be compensated by optical amplification, but this will increase the overall system cost. Apart from cost considerations, optical amplifiers are by no means ideal devices. First, optical amplifiers introduce noise, in addition to providing gain. Second, the gain of the amplifier depends on the total input power. For high-input powers, the amplifier tends to saturate and the gain drops. This can cause undesirable power transients in networks. Finally, although optical amplifiers are capable of amplifying many wavelength channels simultaneously, they do not amplify all channels equally, i.e., their gain is not flat over the entire pass-band.

1.6.3. Electro-optic Switches

A 2x2 electro-optic switch uses a directional coupler whose coupling ratio is changed by varying the refractive index of the material in the coupling region. One commonly used material is Lithium-Niobate (LiNbO3). A switch constructed on lithium-niobate waveguide is shown in Figure 4. An electrical voltage applied to the electrodes changes the substrate's index of refraction. The change in the index of refraction manipulates the light through the appropriate wave guide path to the desired port. An electro-optic switch is capable of changing its state extremely rapidly, typically in less than a nanosecond. This switching time limit is determined by the capacitance of the electrode configuration. Electro-optic switches are also reliable, but they pay the price of high insertion loss and possible polarization dependence. Polarization independence is possible but at the cost of a higher driving voltage, which in turn limits the switching speed.

1.6.4. Liquid-Crystal Switches

The liquid-crystal state is a phase that is exhibited by a large number of organic materials over certain temperature ranges. In the liquid-crystal phase, molecules can take up a

certain mean relative orientation, due to their permanent electrical dipole moment. It is possible because when a suitable voltage is applied across a cell filled with liquid-crystal material change the orientation of the molecules. Hence, optical properties of the material can be altered. Liquid-crystal optical switches are based on the change of polarization state of incident light by a liquid.

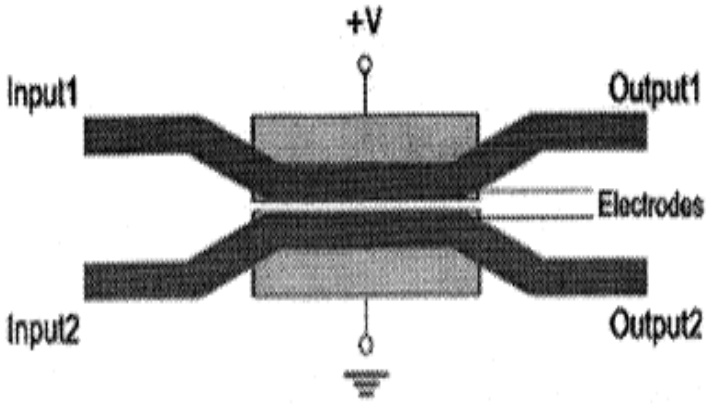


Figure 4: Electro-Optic Switches [6]

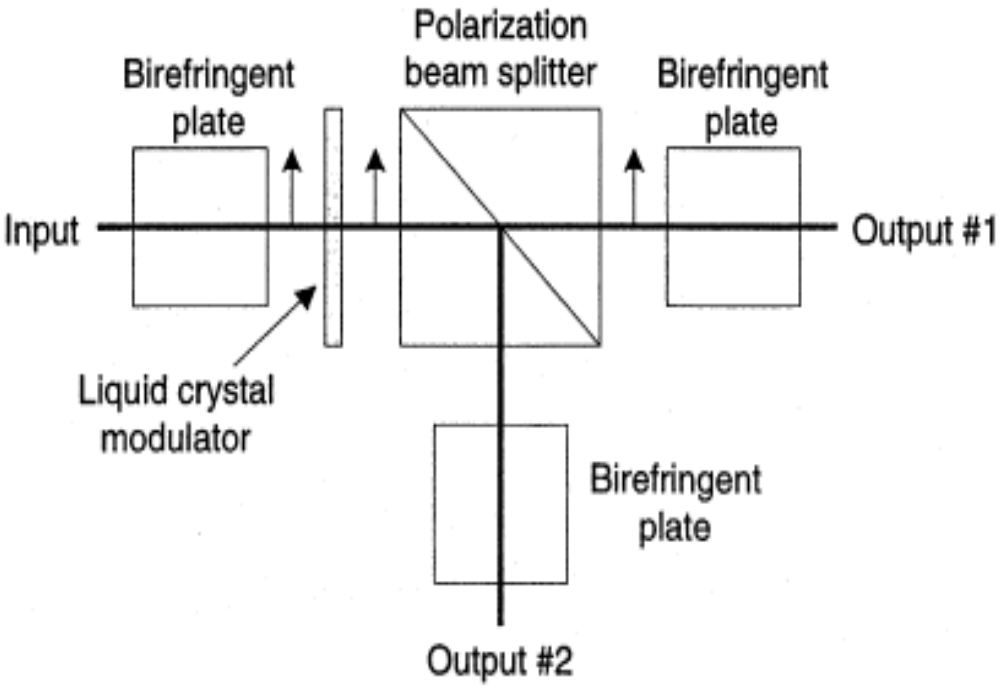


Figure 5: Liquid crystal switch [6]

1.6.5. Semiconductor Optical Amplifier Switches

Semiconductor optical amplifiers (SOAs) [6], [11] are versatile devices that are used for many purposes in optical networks. An SOA can be used as an ON–OFF switch by varying the bias voltage. If the bias voltage is reduced, no population inversion is achieved, and the device absorbs input signals. If the bias voltage is present, it amplifies the input signals. The combination of amplification in the on-state and absorption in the off-state makes this device capable of achieving very high extinction ratios. Larger switches can be fabricated by integrating SOAs with passive couplers. However, this is an expensive component, and it is difficult to make it polarization independent.

Purely optical switches needed to be design to fully exploit the capacity of medium and to cope up the future data transmission demand, major technical difficulties need to be overcome [12],[14]. A comparison between the various types of switches is summarized in the Table 10. In the diverse type of applications we can select the type of switch from the comparison table, best suited to the applications. All Data in tables have been taken from commercially available datasheets [16-24].

1.7. Augmented Data Vortex Switch

Modern high-performance computing systems require networks with high capacity, extremely high throughput and low latency in order to pass messages between thousands of processors and memory elements [24]. Optical Interconnection Networks (OIN) offers a potentially viable solution to this requirement. Since the interconnection networks are at the heart of the system in high performance computing applications, their fault tolerance and reliability are unavoidable issues.

1.7.1. Naming Conventions

The switch fabric size is characterized by two parameters ‘A’ and ‘H’, representing the number of nodes along the ‘angle’ and ‘height’ dimensions, respectively. The available number of input–output (I/O) ports is given by A × H. The number of cylinder levels C is, $C = \log_2 H + 1$ numbered from $0-\log_2 H$, where 0 is the input level and $\log_2 H$ is the output level. Each routing node is labeled uniquely by the coordinates (a, c, h), where, $0 \leq a < A$, $0 \leq c < C$, and $0 \leq h < H$. Data packets enter from the input nodes at the

outermost cylinder and exit from the output nodes at the innermost cylinder. To identify each node by its single decimal address a scheme has been proposed. In this scheme each node label (a, c, h) is converted to its equivalent decimal representation 'nd', where, nd = H × a + (A × H) × c + h. A three dimensional view of the switch is sown in Figure 6.

Table 10: Comparison of optical switching technologies

Parameter	**Opto-Mechanical Switches**	**MEMS Switches (8x8)**	**Electro-optic Switches**	**Semiconductor Optic Amplifier Switches(PSW-4401)**
Wavelength (nm)	1310,1550 both(1310,1550)	1280~1340 1520~1625	1280~1340 1520~1625	1530
Insertion loss (DB)	Max 0.60	Max 3.5	Max 2.0	Typ. 0
Return loss (DB)	Min 55	Min 50	Max 0.45	
PDL (DB)	Max 0.05	Max 0.1	Max 0.1	Max 2.5
Cross talk (DB)	Min 60	Max -50	Max 0.80	Max 0.80
Switching time	4ms	12ms	5 ns	3ns
Power handling	1000(mW)	Max 20 (dBm)	320+10mA 320-10 mA	Max 120mA
Operating Temperature	0~70 C	-5~70 C	0.5~70 C	0~40 C
Available Configurations	1x2 2x2	8x8 16x16 32x32	1x8 1x16	1x4 2x1 4x4
Applications	Very low port count applications, Wave length selective application	Large Port Count	Switching time constraint application	Multicasting and broad casting, Strictly non blocking operations
Effect on scalability	Increase in Fabrication cost, Reduction in performance	Relatively low increase in cost, Practically size greater than 32x32 is not possible	High losses and PDL.	Increase in fabrication cost, Difficult to make PDL independent.

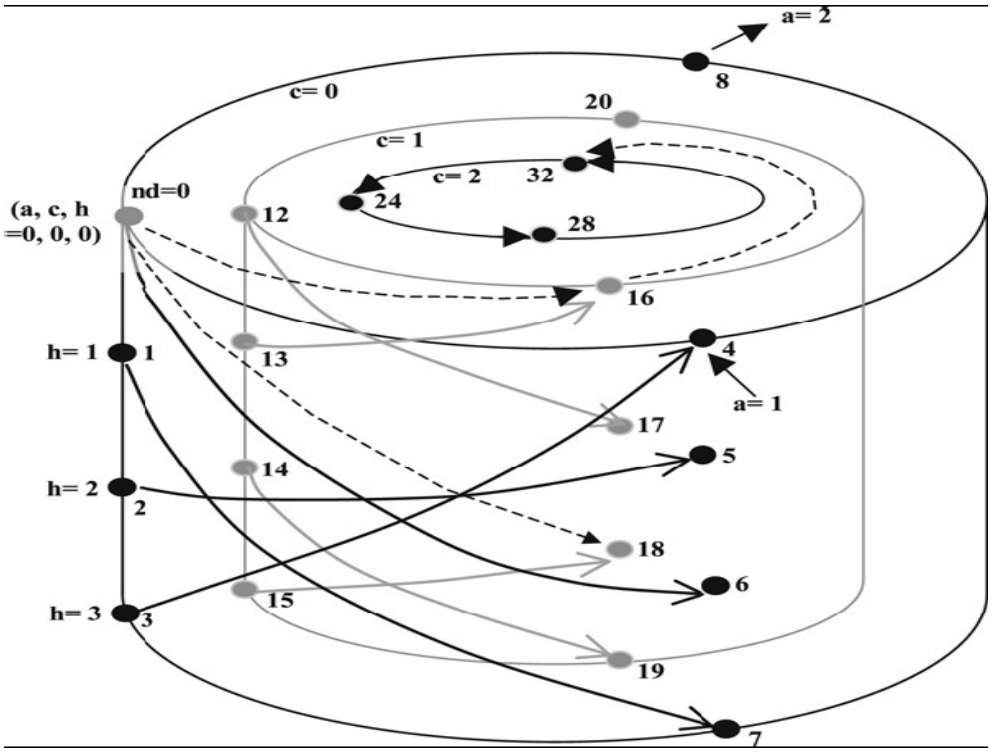


Figure 6: 3-Dimensional view of ADV [25]

1.7.2. Nodes Interconnections

In Figure 6, a switch fabric of size (A=3, H=4) is shown with interconnection patterns, on the same cylinder level as shown in Figure 7 and among different cylinder levels as shown in Figure 8. In the ADV switch, any node N (a, c, h), as shown in Figure 7, at angle a, height h, and cylinder level c, has three data inputs and three data outputs. The node N(a, c, h) on level c has one output terminal connected to node X(a + 1, c, h1), as shown in Figure 7, on the same level c, and two output terminals connected to two nodes Y (a + 1, c + 1, h2) and Z(a+1, c+1, h3) as shown in Figure 8, on level c+1. The height transformations h1 and h2 are similar, while h3 is the augmented connection. For h1 we divide the total number of nodes along the height H into 2c subgroups, where c is the index of cylinders. The first subgroup is then mapped as follows the top half of the remaining nodes at angle a are connected to the bottom half of the remaining nodes at angle a + 1. This step is repeated until all nodes of first subgroup are mapped from angle a to angle a+1. If multiple subgroups exist, the rest of them copy the mapping pattern of the first subgroup. The height h2 is equal to h and h3 is equal to h1. The same connection

patterns are repeated from angle to angle. Similarly, the three inputs to a node N(a, c, h), ($0 < c \leq \log_2 H$) consist of a data input connection from the same level at different height node, a second data input connection from the outer level at same height called node B, and a third data input connection from the outer level at different height, node D as shown in Figure 7.

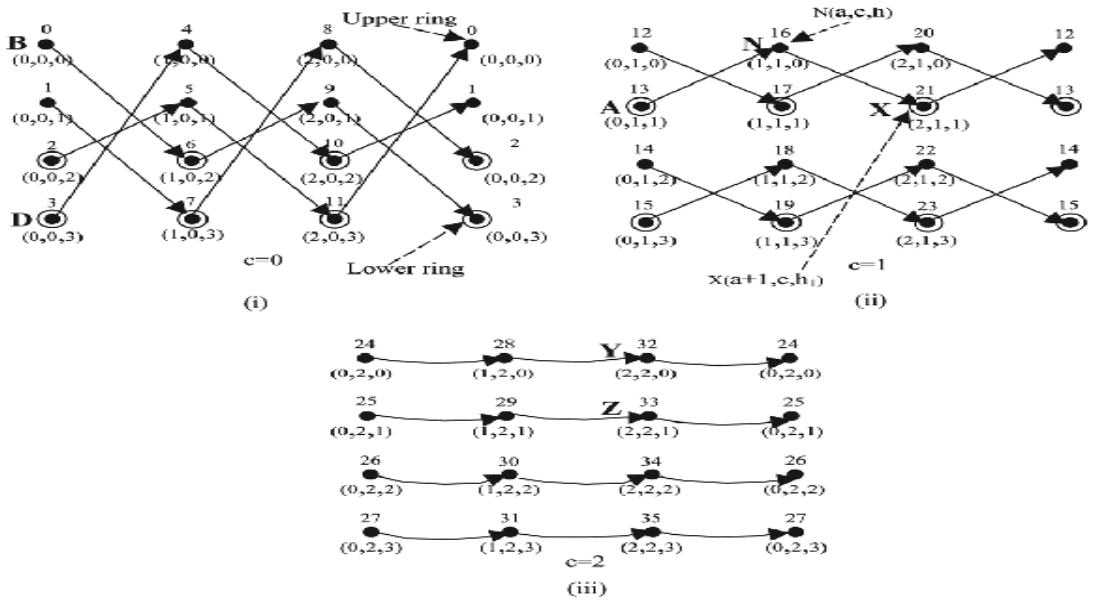


Figure 7: Interconnections on same cylinder levels. [25]

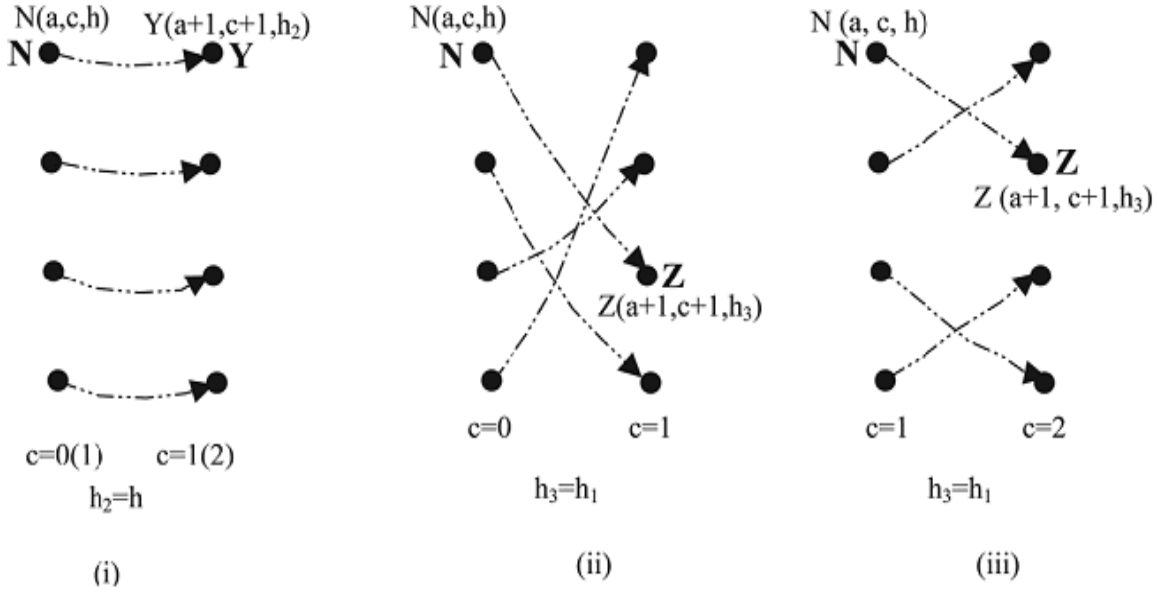


Figure 8: Interconnections among different cylinder levels [25]

1.7.3. Distributed control signaling in the ADV switch

To achieve buffer-less operation and simple routing logic within a node, only one input packet is permitted to enter a node in a given clock cycle [24]. Hence, to avoid conflict between the three inputs for node access, a modified priority scheme is applied to messages with the help of separate control bits, other than the header and the payload bits. Control bits are allowed to pass between nodes, through separate control inputs and outputs at each node, before the packet arrives at a given node. For example, a message at node A, moving on the same level is given first priority, a message at node B, progressing one level at the same height is given second priority, and message at node D, progressing one level at a different height is given third priority. Suppose A has a packet to pass to N. Then it sends a control signal to B, which in turn sends a control signal to D, that blocks data at B and D from progressing to N. The blocked packets are deflected to corresponding nodes on their current cylinder level. If no packet is passing from A to N, but B and D both have packets to send to N, then A passes a signal to B to pass the packet to N, and B passes a signal to D, to deflect the packet on the same level. If A and B have no packets to pass to N, then A passes a signal to B, and B to D, to pass the packet to N. The control messages thus permit only one packet to enter a node in any given time period. Thus the situation of two or more packets contending for the same output port never occurs. This eliminates the need for optical buffers at the nodes and simplifies the logic operations at the nodes.

1.7.4. Self-Routing scheme

This section explains a new self routing scheme for the ADV switch with 3 × 3 nodes. As seen in Figure 8, each level c has 2c rings, and each ring has N/2c nodes [24]. For the purpose of self-routing in the ADV switch, the nodes in any complete ring are divided into two halves: the ‘upper ring’ and the ‘lower ring’. The upper ring consists of the half of the nodes of the ring which are physically located on the upper half portion of the ring and the lower ring consists of the rest half of the nodes which are located on the lower half. In Figure 7, the nodes in upper ring are represented by simple black dots; whereas the nodes in lower ring are represented by encircled black dots. The routing takes place as follows: The packets are injected at the outermost cylinder (c = 0) and emerge at the

innermost cylinder ($c = \log_2 H$). The total number of header bits in the packet header is ($\log_2 H$), which represents the height of the destination ring in binary. Each header bit is encoded by a unique wavelength. As the packet moves from the outer cylinder to the inner cylinder, every cylindrical progress extracts (decodes) a specific bit within the binary header address, using wavelength filtering. At any node on cylinder level i, bit di of the header is decoded using wavelength filtering. For the node on the 'upper ring', if di = 0, the packet is routed to the upper output link of the 3 × 3 node to the next level, If di = 1, it is switched to the lower output link. For the node on the 'lower ring', the routing logic is just reversed, i.e. if di = 0, the packet is routed to the lower output link, and if di = 1, it is switched to the upper output link. If the destined output link is 'blocked' due to conflict, or a link failure, or a failure in the node at the next level to which the output link is connected, the packet is routed (deflected) through the chain-out-link to another node within the ring. An identical routing algorithm in the new node will utilize the same di bit. If the destined output link is again 'blocked', this packet will be routed to yet another new node in the ring. A packet can be routed through as many nodes within a ring as needed. When a good output link is found the packet proceeds to the next level, and in this way, finally reaches the destination ring on cylinder level $\log_2 H$. The last cylinder level allows the packet to circulate, thus providing temporary output buffering.

Chapter 2 Literature Survey

2.1."Permutation Capability of Optical Multistage Interconnection Networks", Yuanyuan Yang, Jianchao Wang:

In this paper, optical multistage interconnection networks (OMINs) are studied. This paper answers that why and how the optical communication can meet the emerging demand of the bandwidth in future. This paper state that in optical transmission, performance is high and have low communication latency. Although optical MINSs hold great promise and have demonstrated advantages over their electronic counterpart, they also hold their own challenges. Due to the unique properties of optics, crosstalk in optical switches should be avoided to make them working properly. Most of the research work described in the literature is for electronic MINSs, and hence, crosstalk is not considered. In this paper, we introduce a new concept semi permutation to analyze the permutation capability of optical MINSs under the constraint of avoiding crosstalk [30].

2.2."Optical Multistage Interconnection Networks: New Challenges and Approaches", Yi Pan, C. Qiao and Y. Yang:

This paper is focusing on an introduction about Optical interconnections for communication networks and multiprocessor systems. It is also discussing the basic element of optical switching networks is a directional coupler with two inputs and two outputs or switching elements. Depending on the control voltage applied to it, an input optical signal is coupled to either of the two outputs, setting the SE to either the straight or cross state. A class of topologies that can be used to construct optical networks is multistage interconnection networks, which interconnect their inputs and outputs via several stages of SEs (Switching elements), are also mentioned in this paper. In this paper various ways to deal with the unique problem of avoiding crosstalk in the SEs (Switching elements) are given [31].

2.3.“WDM Packet Routing for High Capacity Data networks”, Qimin Yang, Student Member, IEEE, Keren Bergman:

In this paper experimental and numerical analysis of novel packet-switch architecture, the data vortex, designed for large-scale photonic interconnections have been studied. The self routing multi hop packet switch efficiently scales to large port counts (>10k) while maintaining low latencies, a narrow latency distribution, and high throughput. To facilitate optical implementation, the data-vortex architecture employs a novel hierarchical topology, traffic control, and synchronous timing that act to reduce the necessary routing logic operations and buffering. As a result of this architecture, all routing decisions for the data packets are based on a single logic operation at each node. The routing is further simplified by the employment of wavelength division multiplexing (WDM) encoded header bits, which enable packet-header processing by simple wavelength filtering. The packet payload remains in the optical domain as it propagates through the data-vortex switch fabric, exploiting the transparency and high bandwidths achievable in fiber optic transmission. In this paper, we discuss numerical simulations of the data-vortex performance and report results from an experimental investigation of multi hop WDM packet routing in a re-circulating test bed.

2.4.“Message Routing and Scheduling in Optical Multistage Networks Using Simulated Annealing”, Ajay K Katangur, Yi Pan, Martin D Fraser:

In the optical transmission, major issue is the Cross Talk. In the paper, focus is on the ways to avoid the cross talk. In the paper routing algorithms for routing the traffic through N*N have been discussed. Many researchers have proposed the many algorithms like, Sequential algorithm, degree descending algorithm, etc. In this research paper simulated annealing algorithm is used to improve the performance and optimizing the result [29].

2.5.“Gemini: An Optical Interconnection Network for Parallel Processing”, Roger D. Chamberlain:

In this paper Gemini network has been studied. This paper state that the Gemini interconnects is a dual technology (optical and electrical) interconnection network designed for use in tightly coupled multicomputer systems. It consists of a circuit-switched optical data path in parallel with a packet-switched electrical control/data path [27]. The optical path is used for transmission of long data messages and the electrical path is used for switch control and transmission of short data messages. This paper enclosed the details of architecture of the interconnection network and related communications protocols.

2.6.“Efficient parallel routing algorithms in Optical Multistage Interconnection Network”, M. Abdullah, M. Othman, R. Johari:

In This paper the focus is on the parallel routing. The paper is written by keeping the OMEGA having shuffle exchange network in mind. In this paper Sequential Window Method (SWM), UPWM and BPWM have been discussed. The research shows that the Balanced Parallel Window Method (BPWM) is 85% faster than Sequential Window Method SWM [3].

2.7.“New algorithm to avoid crosstalk in optical Multistage Interconnection Networks”, M. A. Shabi, M Othman, R. Johari, S. Subramanian:

In this paper major focus is on the cross talk. The paper has been written by focusing on the OMEGA shuffle exchange networks. In this paper Window Method is discussed and a new Method is proposed called ZeroX Method. The research state that in the average number of passes, the result of the ZeroX algorithm is much better than the SA [5].

2.8.“A Comparison Study of Optical MIN Networks with Parallel Planes”, Qimin Yang:

The major challenges in designing optical switched interconnection networks include the lack of optical buffering and the crosstalk originated from the optical switching elements. Vertical Stacking Banyan networks and Data Vortex networks provide two different approaches to solve the problem. Both networks utilize several parallel planes in conjunction with traffic scheduling to eliminate the contention and the crosstalk within the routing nodes. While the two share similar compromise between system cost and switch performance, their traffic control mechanism has fundamental difference due to specific configurations of the parallel routing planes. This paper provides a comparison study of two network approaches, focusing on the overall system complexity and the resulting switch performance in throughput and latency [34].

2.9.“Efficient Window Method in Optical Multistage Interconnection Network”, Farzaneh Abed, Mohamed Othman:

In this paper an efficient method called “Window Method” has been discussed. Focus of this paper is on how to switch the packets to avoid the cross talk. This method is focusing on N*N network and discussing a way to group the packets and transmit them in such a way so that there is no cross talk. In this paper, the main issue is the total number of windows for N*N network is given be $M=\log_2 N$. In this paper proposed BWM method is compared with WM and IWM. The research shows that BWM is much faster than WM and faster than IWM [4].

2.10.The Augmented Data Vortex switch fabric: An all-optical packet switched interconnection network with enhanced fault tolerance”, Neha Sharma, D. Chadha, Vinod Chandra:

This paper is focusing on the fault tolerance of the Data Vortex (DV). This paper state that in future high-performance computing systems require networks with high capacity, extremely high throughput and low latency in order to communicate thousands of processors and memory elements. Optical Interconnection Networks (OIN) offers a potentially viable solution to this requirement. An all-optical packet switched interconnection network called a Data Vortex (DV) designed by Yang et al.

In the Optical network, fault tolerance and reliability are crucial issues, evaluation of which lacked attention for the case of the DV switch [25]. (1) So a new architecture has been proposed called Augmented Data Vortex (ADV) to improve the fault tolerance. (2) The labeling and a numbering scheme. (3) A new self-routing procedure and a priority scheme for distributed control signaling (4) For the first time, on version of the 3-dimensional switch to an equivalent chained-MIN model, has been given, which is more suitable for later analysis of fault tolerance. (5) A multiplexing scheme at input ports and output ports which further enhances the fault tolerance of the ADV switch has been given. (6) Computation has been done of the reliability and fault tolerance of the new architecture via an analytical model. (7) Finally, comparison of the ADV switches architecture with the primary architecture (DV) in view of fault [25].

2.11."Performance evaluation of the Augmented Data Vortex switch fabric: An all-optical packet switched interconnection network", Neha Sharma, D. Chadha, Vinod Chandra:

This paper is focusing on the fault tolerance of the Augmented Data Vortex (ADV). In the optical transmission network, fault tolerance and reliability are crucial issues. The performance as regards fault tolerance of the ADV switch was computed and detailed results were obtained. This paper is focusing on performance of ADV, investigated with reference to parameters such as latency and injection ratio (throughput) by means of numerical simulations. A uniform random traffic model has been used for the performance evaluation. The results obtained are compared with the results reported for the DV switches. The research paper results show that the ADV switch with enhanced fault tolerance also improves the performance regarding latency [26].

2.12."Fast method to find conflicts in optical multistage interconnection networks", F. Abed, M. Othman:

This paper presents, one undesirable problem introduced by the optical multistage interconnection network is cross talk. It is caused by coupling two signals within a switching element is considered. To avoid a crosstalk, many approaches have been proposed such as time domain and space domain approaches. Because the messages should be partitioned into several groups to send to the network, window method is used to find out which messages should not be in the same group. In this paper fast window method based on bitwise operations (BWM) is represented. This algorithm reduces the execution time approximately more than ten times compared with previous algorithms Window Method (WM) [35].

2.13."A new Algorithm for Routing And Scheduling In Optical Omega Network", M. A. Al-Shabi, M. Othman:

In this paper the major focus is on the multistage interconnection networks (MINs) as a switching application and related problem occurring in it. The major problem in Using the MIN as the switching application is the crosstalk. In this paper a new algorithm ZeroY is proposed to avoid the cross talk and compared with the Other SA algorithms and heuristics algorithms. The result of the research shows that the ZeroY algorithm is more efficient and can reduce the number of passes used to transmit the message without cross talk [32].

Chapter 3 Problem Statement

The problem undertaken for dissertation is **"A new buffered fabrication technique for augmented data vortex switch".** In the Optical Transmission we generally use electro-optic switch. From the literature survey, switch architecture was proposed called Augmented Data Vortex (ADV). In the ADV control bit is converted into electric signal by opto-electric convertor and the control path is decided. But ADV uses a hierarchical scheme of the connection. Each node of the switch has three inputs and outputs. One of the outputs is connected to the same level and other two are on the inner level. When the network is extremely congested (peak load, heavy broadcast domain), an infinite loop can be created or considerable delay can be introduced in the network performance.

To avoid the problem of time delay and to resolve the problem of infinite loop creation in the network, a new switch architecture named BADV (Buffered augmented data vortex) has been proposed. The proposed BADV architecture uses buffer for temporary storage, SRQ for maintaining the services by the node. A simulator has been designed in VB 6.0 with the following features.

a.) To show the working and problems of existing ADV architecture.

b.) To show the working of newly proposed architecture BADV.

Chapter 4 Buffered Augmented Vortex

4.1. Introduction

Future is the time of high band width communication. Even today, the present technology is not sufficient to cope up with tremendous demand of bandwidth. Photonic transfer is one of the technologies, which can satisfy the bandwidth requirement as per the current technologies. Electro-optic switches are used in conjunction with the optical media for switching the data packets. In this regard, a switch architecture named as augmented data vortex (ADV) is available [24],[25]. While designing the switch architecture, attention must be paid towards the time delay in delivering the packets, in the peak load hours. But in the switch architecture less attention has been paid to the peak loads hours time delay. In this thesis time delay is considered in peak load hours to improve the performance. New switch architecture is proposed that improve performance in the peak load hour. ADV is compared with the new switch Buffered augmented data vortex (BADV).

4.2. Buffered augmented data vortex Switch

In the Buffered augmented data vortex switch each node contains three buffers, input buffer, service request queue and waiting buffer. Buffered augmented data vortex Switch (BADV) follows the same connection schema as in Augmented Data Vortex. Each node has three inputs and outputs. Packets enter from the outermost layer and emits from innermost layer. In no load condition the BADV behaves like ADV, but when traffic congestion occurs, its working changes, the role of the buffer come into existence and help to eliminate/avoid the infinite loop that can cause the network down or slow response.

The block diagram of the Buffered augmented data vortex is shown in Figure 9. Packet comes to the outermost level at any node at first enter into Input Buffer. If the node at inner level is busy then packet is placed into the waiting buffer. Service Request Queue is mainly responsible for handling the services.

4.3. Naming Convention and Node Interconnection

Buffered augmented data vortex follows the same naming convention and Node Inter connection schema but the internal architecture of the node is changed. BADV Node consists of various buffers to perform its operation.

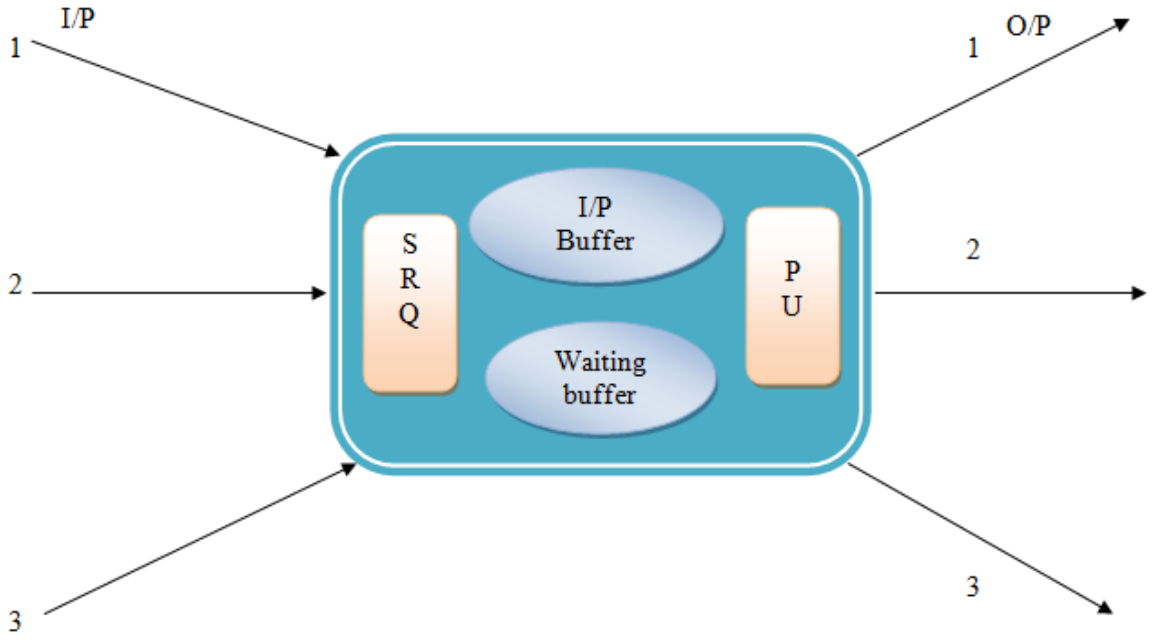


Figure 9: Buffered augmented data vortex

4.4. Self Routing Scheme in the Buffered augmented data vortex

In this section routing scheme of BADV switch with 3 × 3 nodes is explained. Since the node contains same number of inputs and outputs and control pins, so it will follow the same scheme while the switch doesn't have the peak load or inner level node is not busy. The major role of buffer will be visible when inner node is busy. All BADV nodes are divided into 2 levels. Each level c has 2c rings, and each ring has N/2c nodes and contains three buffers named as input buffer, waiting buffer, service request queue. For the purpose of self-routing in the ADV switch, the nodes in any complete ring are divided into two halves: the 'upper ring' and the 'lower ring'. The upper ring consists of the half of the nodes of the ring which is physically located on the upper half portion of the ring and the lower ring consists of the rest half of the nodes which is located on the lower half [25]. The routing takes place as follows: The packets are injected at the outermost

cylinder (c = 0) and emerges at the innermost cylinder (c = $\log_2$ H). The total number of header bits in the packet header is ($\log_2$ H), which represents the height of the destination ring in binary. Each header bit is encoded by a unique wavelength. As the packet moves from the outer cylinder to the inner cylinder, every cylindrical progress extracts (decodes) a specific bit within the binary header address, using wavelength filtering. At any node on cylinder level i, bit di of the header is decoded using wavelength filtering. For the node on the 'upper ring', if di = 0, the packet is routed to the upper output link of the 3 × 3 node to the next level, If di = 1, it is switched to the lower output link. For the node on the 'lower ring', the routing logic is just reversed, i.e. if di = 0, the packet is routed to the lower output link, and if di = 1, it is switched to the upper output link. If the destined output link is 'blocked'. The complete packet is stored into waiting buffer, input buffer is made clear, and a deflection timer is set. A request is made to node intending to send the data, the address of the node is placed in the 'Service Request-Response Queue' and tending node also stores the address of the requesting node. When the tending node is free or it has completed its assigned packet transfer, it will give preferred service to the node had sent the service request (whose address is in the SRQ). It will send an acknowledgement to the tending node and will request for data transmission. The tending node will transmit the data immediately and free SRQ buffer on both sides, waiting buffer after transmission. In the mean time if another packet comes to input buffer and waiting buffer is occupied then packet is placed in the input buffer and wait for the waiting buffer to be free. If within the time the tending node doesn't respond then packet is analyzed for the deflection bit, if not set then packet is transmitted to the same level as in ADV and deflection bit is set. If deflection bit is set then packet is discarded. In this way; the packet only travels one hop in case of peak load hours and eliminates the infinite loop. When there are peak load hours, packet has to wait only for the 4-clock cycle to get disappear from the existing level.

4.5. Algorithm: BADV simulator in VB 6.0

This section explains the algorithm of simulator; BADV. The simulator has been coded in VB 6.0 to show the working of the BADV.

For transferring the data packets from outer layer to inner layer node do.

1. Initiate the node 2 and 18.
2. Initiate the node 3 and 16.
3. Make node 2 and 18 color green and link between them and start the data transfer
4. Make node 3 and 16 color green and link between them and start the data transfer
5. Initiate node 0.
6. If node (16/18) busy do following
7. Show the link between 0 to 16 and 0 to 18 red as buy.
8. Initiate the waiting buffer of node 0.
9. Place the data packet at node zero to waiting buffer and assign high priority to it
10. Find the lower Node number connected to node 0.
11. Finish the data transfer from 0 to 16.
12. Send the service acknowledgement from node 16 to 0.
13. Transmit the data packet from node 0 to 16.
14. Update the SRRQ of node 16 and 0.
15. Clear the waiting buffer of node 0.
16. Else if Node(16) free transmit the data packet to node 16
17. Else transmit data packet node 18.

4.6. Algorithm: ADV Simulator in VB 6.0

In this algorithm the data packet will start to move from node 0 and return to zero. The packet will hang-up and create infinite loop. Algorithm which used to create the simulator is given below.

Step 1: For transferring the data packets from node 0 to 6 do.

1. Initialize the Node 3 and 16.
2. Initialize the Node 2 and 18.
3. Start data transfer between 3 and 16.
4. Start data transfer between 2 and 18.
5. Initialize Node 0 to transmit data packet.
6. If busy node(16/18)

Deflected data packet from 0-> 6 on same level.

Step 2: For transferring the data packets from node 6 to 9 do.

1. Initialize the Node 4 and 22.
2. Initialize the Node 5 and 21.
3. Start data transfer between 4 and 22.
4. Start data transfer between 5 and 21.
5. Initialize Node 6 to transmit data packet.
6. If busy node(21/22)

Deflected data packet from 6-> 9 on same level.

Step 3: For transferring the data packets from node 9 to 3 do.

1. Make free to Node 3.
2. Initialize the Node 10 and 13.
3. Initialize the Node 11 and 15.
4. Start data transfer between 10 and 13.
5. Start data transfer between 11 and 15.
6. Initialize Node 9 to transmit data packet.
7. If busy node(13/15)

Deflected data packet from 9->3 on same level.

Step 4: For transferring the data packets from node 3 to 4 do.

1. Make free to Node 4.
2. Initialize the Node 0 and 16.
3. Initialize the Node 1 and 19.
4. Start data transfer between 0 and 16.
5. Start data transfer between 1 and 19.
6. Initialize Node 3 to transmit data packet.
7. If busy node(16/19)

Deflected data packet from 3->4 on same level.

Step 5: For transferring the data packets from node 4 to 10 do.

1. Make free to Node 10.
2. Initialize the Node 7 and 20.
3. Initialize the Node 6 and 22.

4. Start data transfer between 7 and 20.
5. Start data transfer between 6 and 22.
6. Initialize Node 4 to transmit data packet.
7. If busy node(20/22)

    Deflected data packet from 4->10 on same level.

Step 6: For transferring the data packets from node 10 to 1 do.

1. Make free to Node 1.
2. Initialize the Node 9 and 13.
3. Initialize the Node 8 and 14.
4. Start data transfer between 9 and 13.
5. Start data transfer between 8 and 14.
6. Initialize Node 10 to transmit data packet.
7. If busy node(13/14)

    Deflected data packet from 10->1 on same level.

Step 7: For transferring the data packets from node 1 to 7 do.

1. Make free to Node 7.
2. Initialize the Node 2 and 17.
3. Initialize the Node 3 and 19.
4. Start data transfer between 2 and 17.
5. Start data transfer between 3 and 19.
6. Initialize Node 1 to transmit data packet.
7. If busy node(17/19)

    Deflected data packet from node 1->7 on same level.

Step 8: For transferring the data packets from node 1 to 7 do.

1. Make free to Node 8.
2. Initialize the Node 4 and 20.
3. Initialize the Node 5 and 23.
4. Start data transfer between 4 and 20.
5. Start data transfer between 5 and 23.
6. Initialize Node 7 to transmit data packet.
7. If busy node(20/23)

Deflected data packet from 7->8 on same level.

Step 9: For transferring the data packets from node 8 to 2 do.

1. Make free to Node 2.
2. Initialize the Node 10 and 14.
3. Initialize the Node 11 and 12.
4. Start data transfer between 10 and 14.
5. Start data transfer between 11 and 12.
6. Initialize Node 8 to transmit data packet.
7. If busy node(12/14)

Deflected data packet from 8->2 on same level.

Step 10: For transferring the data packets from node 2 to 5 do.

1. Make free to Node 5.
2. Initialize the Node 1 and 17.
3. Initialize the Node 0 and 18.
4. Start data transfer between 1 and 17.
5. Start data transfer between 0 and 18.
6. Initialize Node 2 to transmit data packet.
7. If busy node(17/18)

Deflected data packet from 2->5 on same level.

Step 11: For transferring the data packets from node 5 to 11 do.

1. Make free to Node 11.
2. Initialize the Node 6 and 21.
3. Initialize the Node 7 and 23.
4. Start data transfer between 6 and 21.
5. Start data transfer between 7 and 23.
6. Initialize Node 5 to transmit data packet.
7. If busy node(21/23)
8. Deflected data packet from 5->11 on same level.

Step 12: For transferring the data packets from node 11 to 0 do.

1. Make free to Node 0.
2. Initialize the Node 8 and 12.

3. Initialize the Node 9 and 15.
4. Start data transfer between 8 and 12.
5. Start data transfer between 9 and 15.
6. Initialize Node 11 to transmit data packet.
7. If busy node(12/15)

    Deflected data packet from 11->0 on same level.

Chapter 5 Experimental Results

5.1. Introduction

This section summarizes the results of the experiment, conducted on the existing and proposed methodologies. In the existing methodology, time in terms of clock cycles increases linearly as the number of hops increases. When a network storm comes into existence the time graph grows linearly, causing delay in the delivering of packets.

The above problem is solved in the proposed Buffered augmented data vortex architecture. Where the packet will be sent to the right direction or discarded using maximum four clock cycles.

5.2. Experimental Results of ADV

In Augmented Data Vortex (ADV), time to mature the packet grows linearly with increase in number of hops. When huge load is on the network, the demand of the requirement is more. The bandwidth can be provided by means of optical fiber, but the use of optical fiber is not sufficient. High speed switching devices are also required to accommodate that bandwidth. Augmented data Vortex seems to be fail when the traffic starts to come from each and every side of the switch on the outer layer. The major problem arises because of the temporary buffering; nodes will be busy in keeping the data packets and will not accept the new packets coming on it. Secondly the other problem arises when every node on outer layer have data packets for transmission. The situation is shown through the Snap shots taken from the simulation. The simulator shows how the packets are delivered in a specific situation of infinite loop, when the network storm occurs.

In Figure 10 node ‘2’ is transmitting data packet to node 18 and node 3 is transmitting data packet to node 16. In the mean while a request arises on node 0 to transmit the data packet. Node 0 finds that its inner node 16 and 18 are busy. Therefore it deflects the data packet to the node 6 on the outer layer.

Step 1:

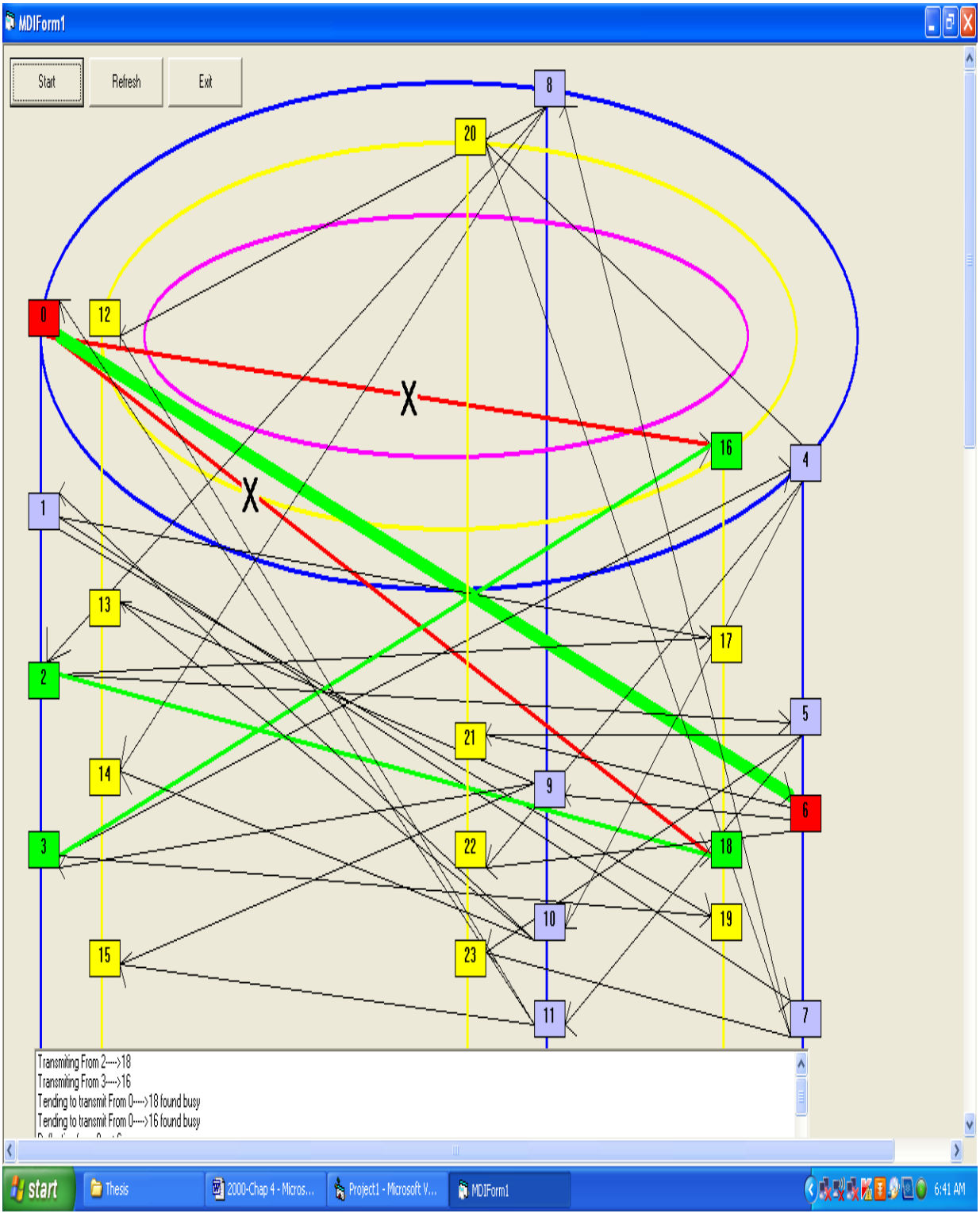


Figure 10: Deflecting data packet from node 0 to 6

Step 2

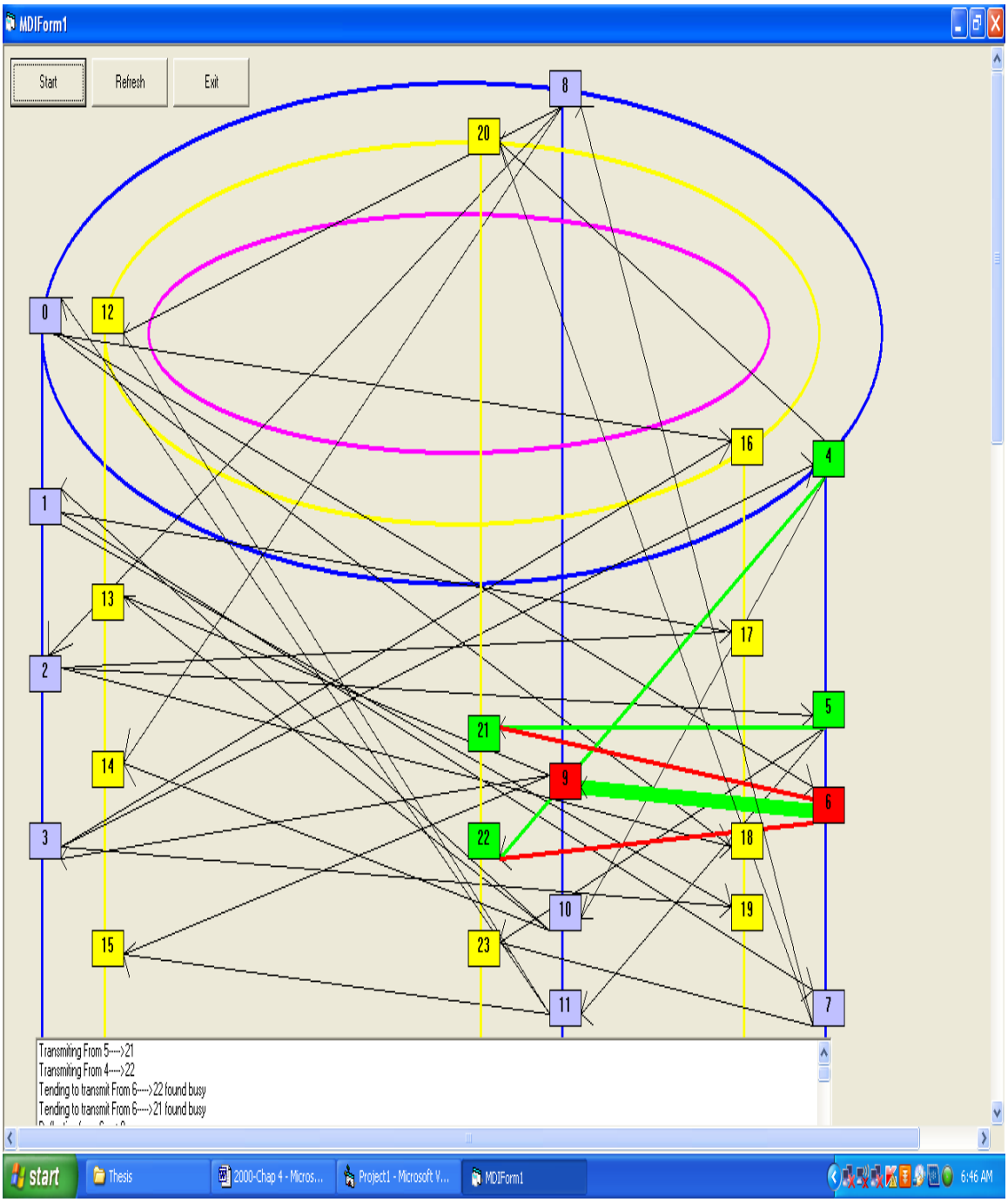


Figure 11: Deflecting data packet from node 6 to 9

In Figure 11 node 4 is transmitting data packet to node 22 and node 5 is transmitting data packet to node 21. In the mean while a request arises on node 6 to transmit the data packet. Node 4 finds that its inner layer nodes 21 and 22 are busy. So it deflects the data packet to the outer layer node 9.

Step 3

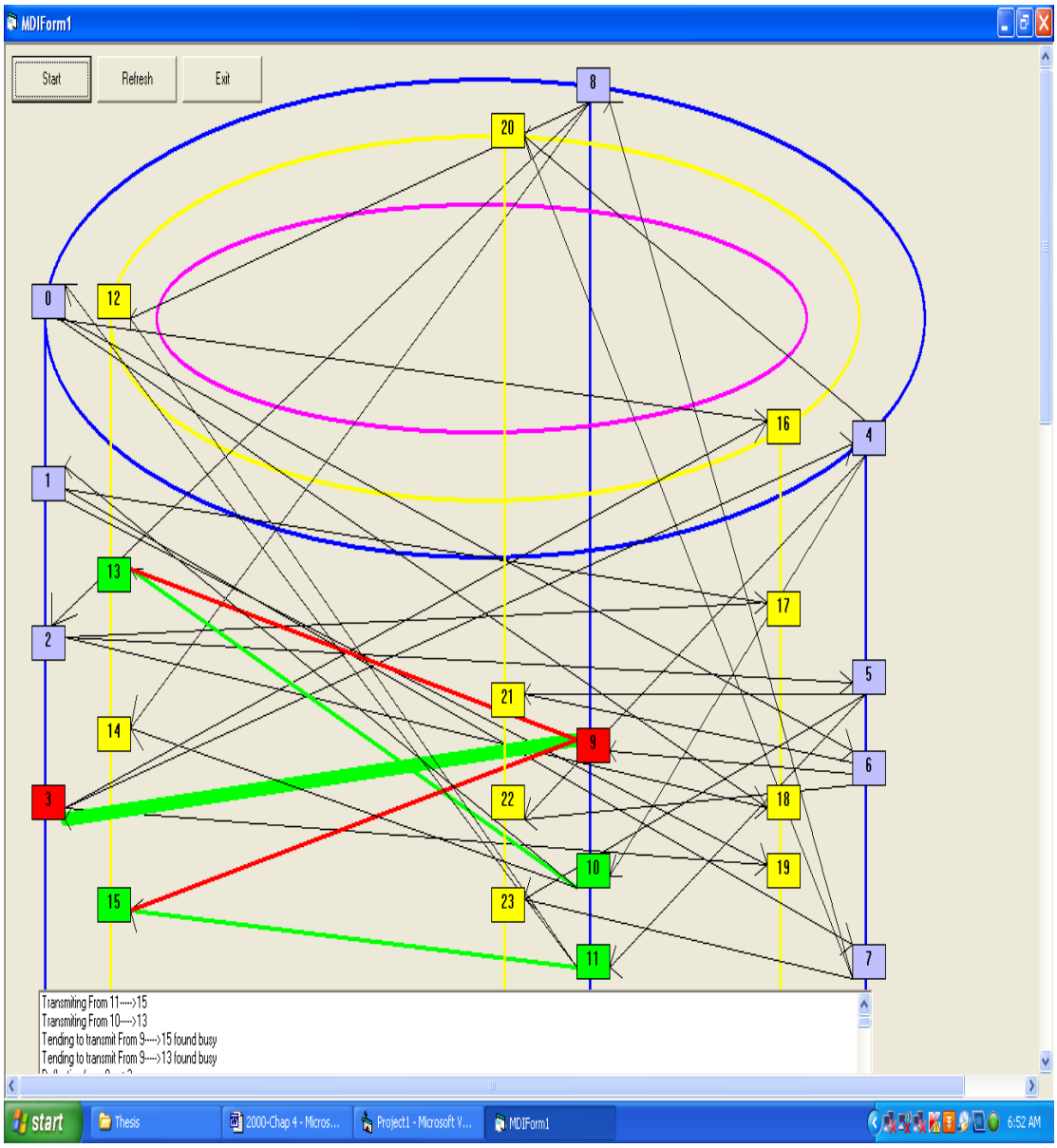


Figure 12: Deflecting data packet from node 9 to 3

In Figure 12 node 11 is transmitting data packet to node 15 and node 10 is transmitting data packet to node 13. In the mean while a request arises on node 9 to transmit the data packet. Node 9 finds that its inner layer nodes 13 and 15 are busy. So it deflects the data packet to the outer layer node 3.

In the same way simulation continues and the data packet remains on outer layer. The data packet doesn't reach to the intermediate layer. It leads to the creation of infinite loop. The final step is shown in Figure 13.

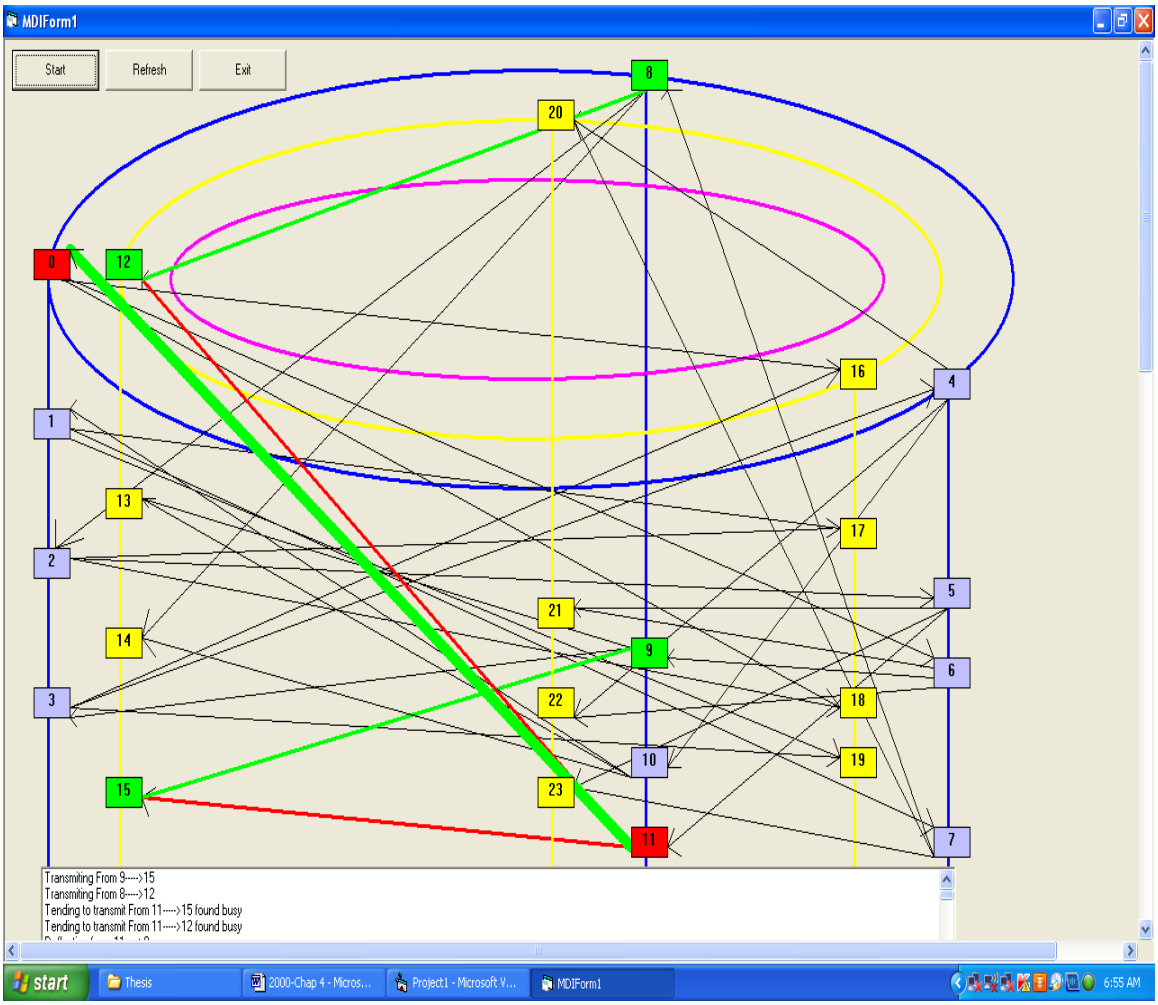


Figure 13: Deflecting data packet from node 11 to 0

In Figure 13 final step is shown, where data packet again reaches to node 0. In Figure 13 node 9 is transmitting data packet to node 15 and node 8 is transmitting data packet to node 12. In the mean while a request arises on node 11 to transmit the data packet. Node 11 finds that its inner nodes 12 and 15 are busy. So it deflects the data packet to the outer layer node 0.

So in this fashion an infinite loop can be created. The problem caused by ADV can be handled by proposed architecture BADV as explained in the next section.

5.3. Experimental Results of BADV

The proposed architecture BADV handle the problem caused by ADV efficiently. The working of BADV is as follows.

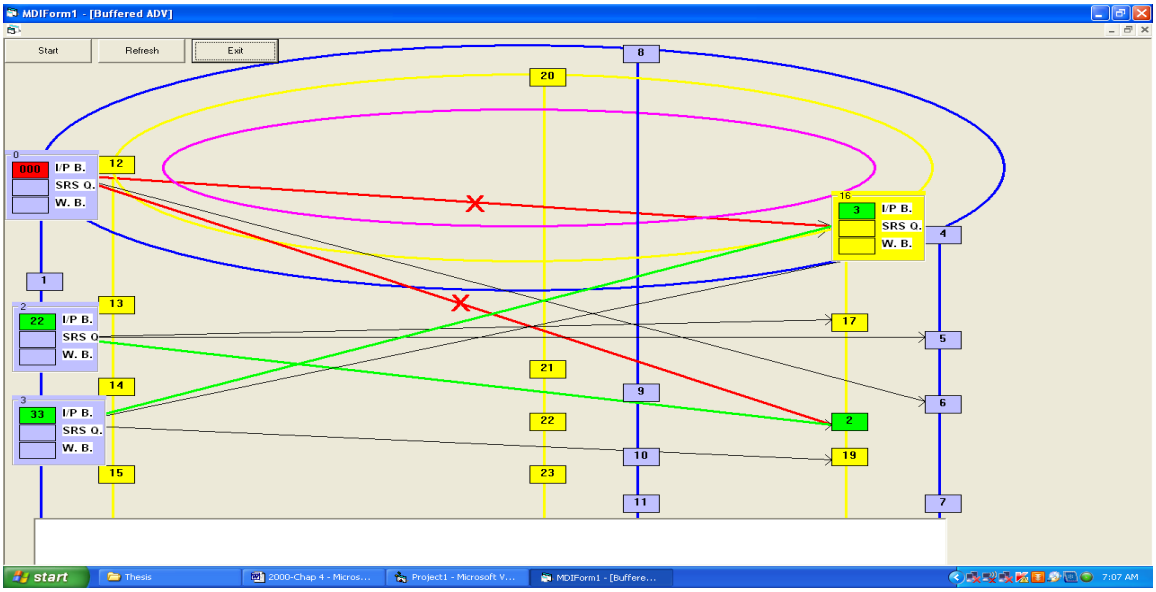


Figure 14: Nodes 16 and 18 are busy for node 0

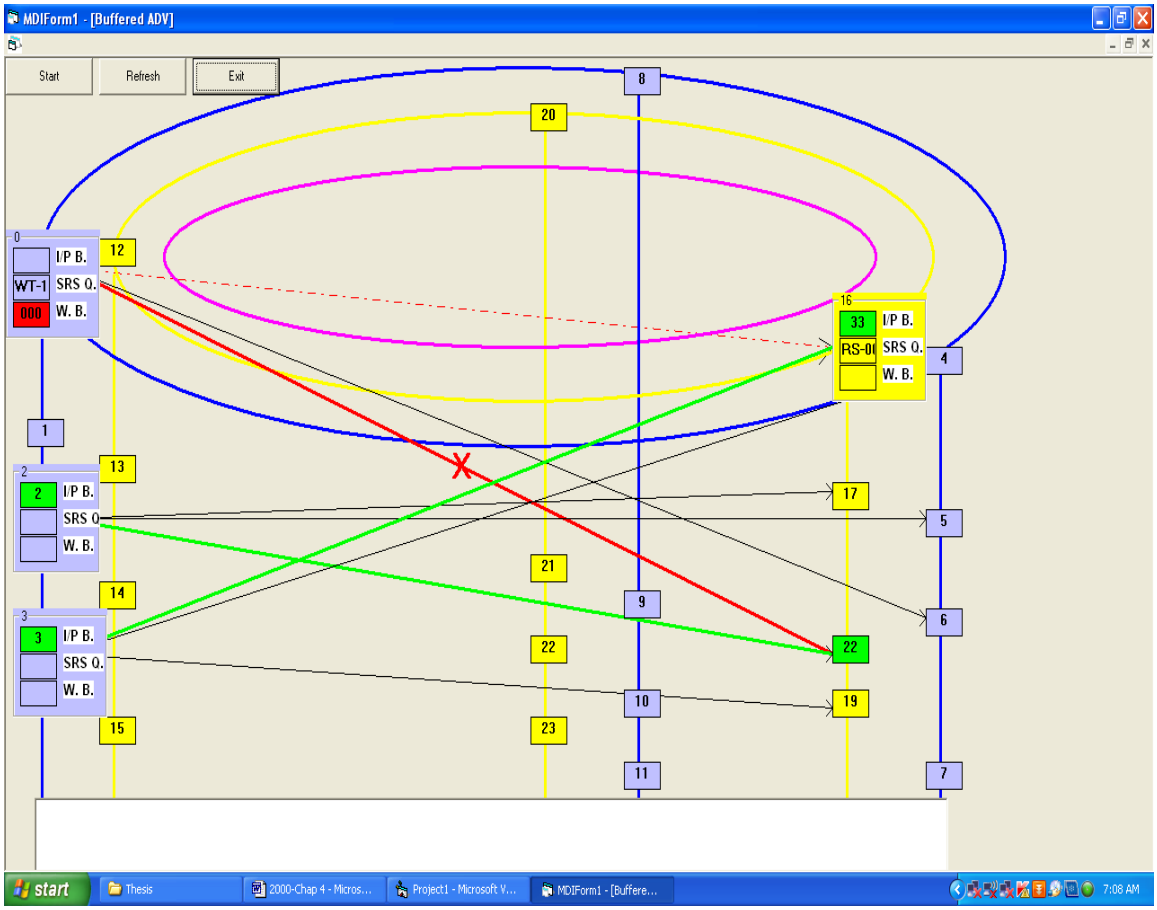


Figure 15: A service request on SRQ of node 16

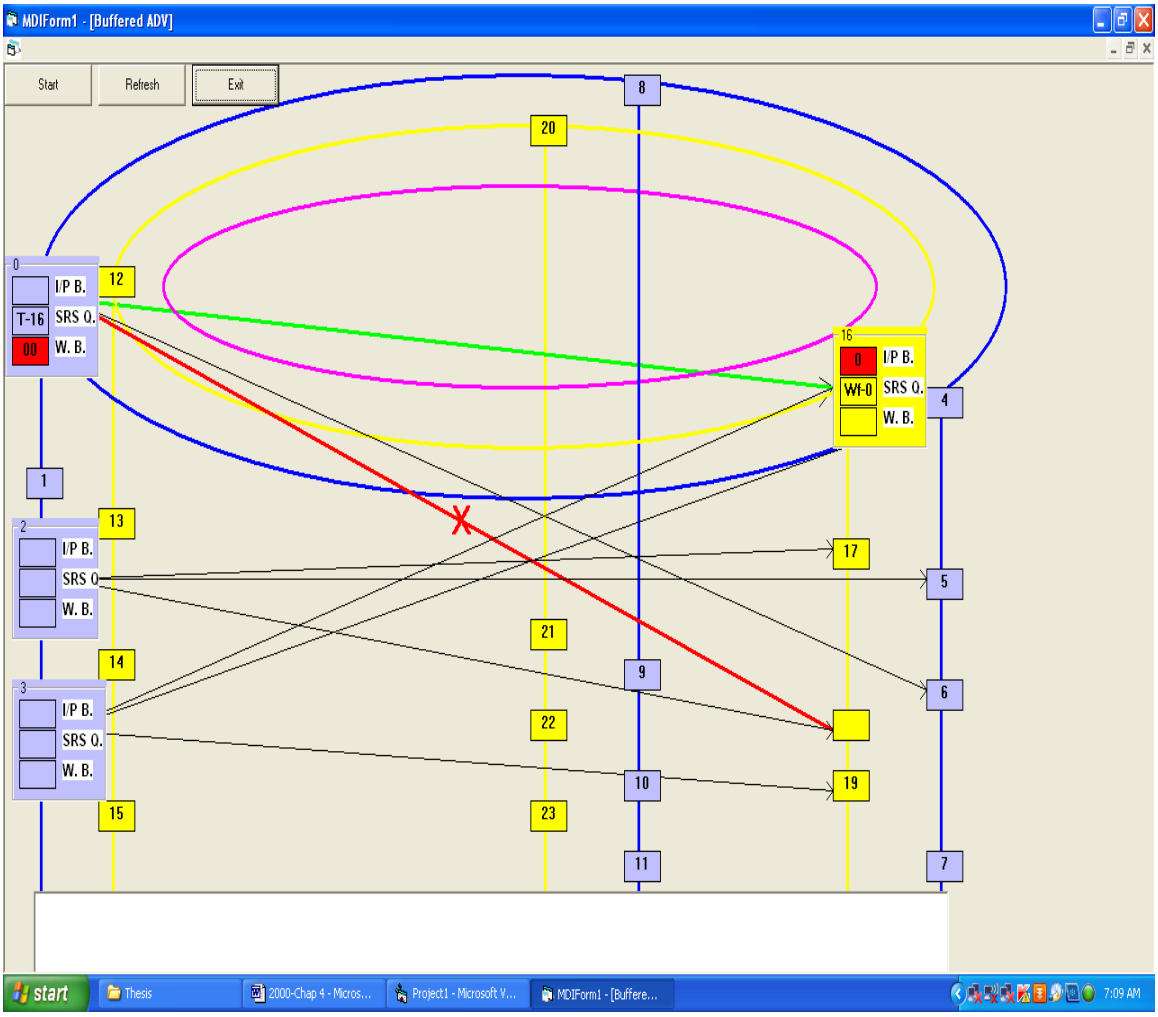


Figure 16: Transmission of data packet from Node 0 to node 16

In Figure 14, Node 2 is transmitting data packet to node 18 and node 3 is transmitting to node 16 in the mean while a request arises on node 0. Node 0 finds that both its inner node 18 and 16 are busy. In Figure 15 it places a request in SRQ buffer of node 16 and wait for it to response. In Figure 16, node 0 is transmitting the packet to 16. In this way infinite loop can be avoided in the BADV.

5.4. Comparison of ADV and BADV

a.) Time analysis of ADV

A graph is plotted in between number of hops Deflected on x-axis and time taken on y-axis.

Table 11: Hops Vs Time

Number of Hops Deflected	Time(Clock Cycle)
1	2
2	3
3	4
4	5
5	6
6	7
7	8

The following graph comes out from the Table 11

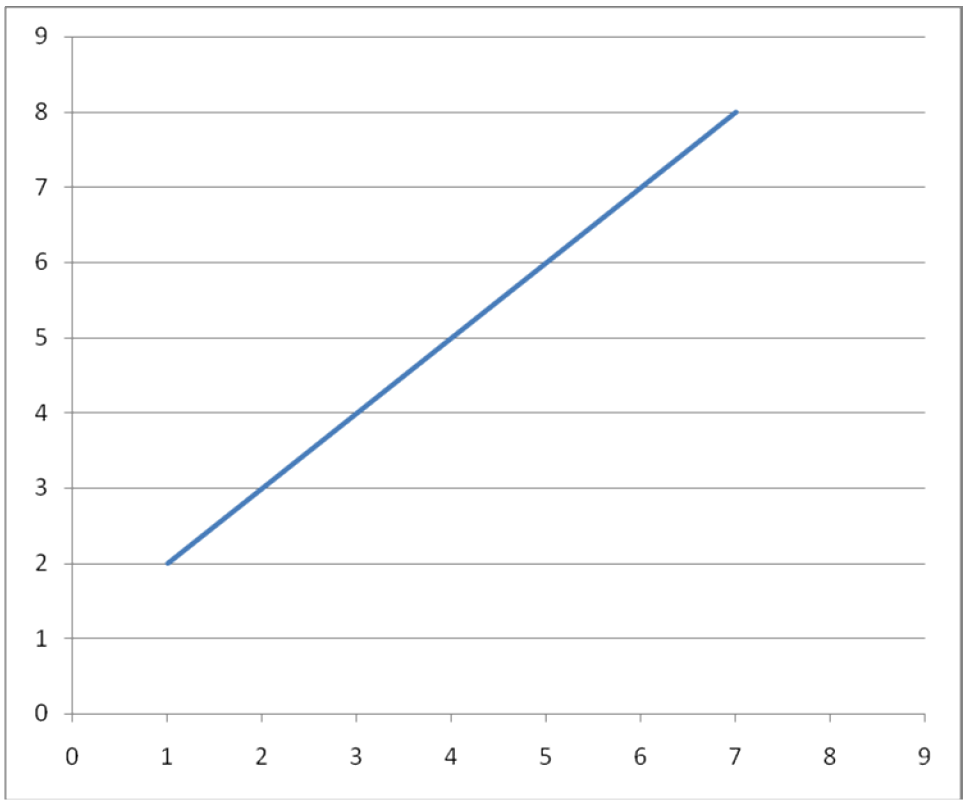


Figure 17: Number of Hops on x-axis VS Time on y-axis in ADV

From the above table it is clear that as the number of hops increases the time to deliver the packet increase constantly. A situation will come and the packet will be deflected continuously on the same level and the will not be delivered to the internal level, at that time infinite loop comes in the existence. The infinite loop leads to the whole network break down. The situation of infinite loop is explained in the section of algorithm for ADV simulator.

b.) Time analysis of BADV

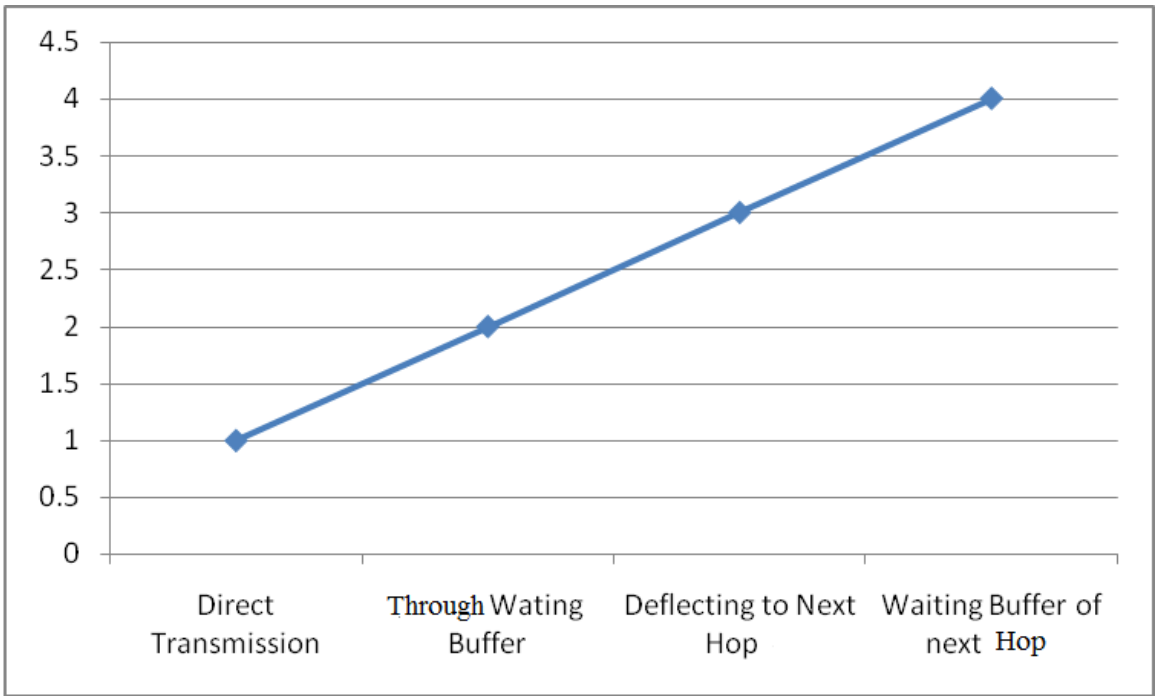


Figure 18: Number of Hops on x-axis VS Time on y-axis in BADV

From the Figure 18 following four situations are clear.

1.) **Direct transmission:** Data packet is transmitted directly to its inner layer node. In this situation inner layer nodes are not found to be busy.

2.) **Through Waiting Buffer:** Data packet is routed through waiting buffer. In this situation inner layer nodes are busy. The data packet is placed in the waiting buffer and a request is raised for service in the SRQ of inner layer node.

3.) **Deflecting to Next Hop:** Data packet is deflected to outer layer and directly transmitted to inner layer. In this situation inner layer nodes are not found to be busy for the next node. In the particular situation data packet takes only four clock cycle to reach to inner layer nodes.

4.) **Through Waiting Buffer of Next Node:** Data packet is routed through waiting buffer of next node. In this situation inner layer nodes are busy for next node. The data packet is placed in the waiting buffer and a request is raised for service in the SRQ of inner layer node. In this situation either the

data packet will be on its right path or it will be rejected. If within the time inner nodes do not response then packet will be discarded. Four clock cycles is the maximum duration for a packet to be mature.

From the above results it is clear that BADV can handle the problem caused by ADV. The infinite loop cannot be created in the BADV. Maximum mature time of a packet in BADV is four clock cycles. The architecture does not lead to the unusual traffic load in the switch and hence improves the performance.

Chapter 6. Conclusion

Major problem in existing architecture Augmented Data Vortex was the considerable time delay and existence of infinite loop at the time of huge network congestion. If the switch is too busy then the time delay will be induced and the situation of infinite loop occurs.

The proposed architecture of BADV (Buffered augmented data vortex) resolves the problem of infinite looping and improves the performance in terms of time delay. In the proposed architecture within 4 clock cycles, the packet will be deflected towards its right path or it will be rejected. Working of ADV and BADV is shown graphically through a simulator developed in VB 6.0.

Chapter 7 Future Work

1. Irregular MIN can be used for interconnection and performance can be evaluated.
2. The switch size can be increased to provide the fault tolerance.
3. Performance can be calculated with the deflection up to two or more hops.
4. Simulator can be coded in other languages to match and show the results.

References

[1] G.B. Adams, D.P. Aggrawal, H.J. Siegel, “A survey and comparison of fault tolerant in multistage interconnection networks”, IEEE Computers, vol. 20, no. 6, pp. 14-27, 1987.

[2] T. Y. Feng, “A Survey of Interconnection Networks”, IEEE Computer, vol. 14, pp. 743-758, 1981.

[3] M. Abdullah, M. Othman, R. Johari, "Efficient Parallel Routing algorithms in Optical Multistage Interconnection Networks”, 13th IEEE International Conference on Networks, vol. 1, pp. 505-509, 2005.

[4] F. Abed., M. Othman, "Efficient Window Method in Optical Multistage Interconnection Networks”, IEEE International Conference on Networks, pp. 181-185, 2007.

[5] M.A. Al-Shabi, M. Othman, R. Johari, S. Subramanian, "New algorithm to avoid crosstalk in Optical Multistage Interconnection Networks”, 13th IEEE International Conference on Networks, vol. 1, pp. 501-503, 2005, doi: 10.1109/ICON.2005.163 5534.

[6] I. G. Papadimitriou, Chrisoula Papazoglou, A. S. Pomportsis, “Optical Switching: Switch Fabrics, Techniques and Architectures”, IEEE Communication Magazine, pp. 384-405, 2003.

[7] R. Ramaswami, K. N. Sivarajan, “Optical Networks: A Practical Perspective”, vol. 2, Morgan Kaufmann Publishing, 1998.

[8] J.I. Dadap, P.B. Chu, I. Brener, C. Pu, C.D. Lee, K. Bergman, N. Bonadeo, T. Chau, M. Chou, R. Doran, R. Gibson, R. Harel, J.J. Johnson, S.S. Lee, S. Park, D.R. Peale, R. Rodriguez, D. Tong, M. Tsai, C. Wu, W. Zhong, E.L. Goldstein, L.Y. Lin, J.A. Walker, "Modular MEMS-based Optical Cross-Connect with Large Port-Count”, IEEE Photonics Technology Letters, vol. 15, no. 12, pp. 1773-1775, 2003.

[9] D. J. Bishop, C. R. Giles, G. P. Austin, "The Lucent LambdaRouter: MEMS technology of the Future here Today”, IEEE Communications Magazine, vol. 40, no. 3, pp. 75-79, 2002.

[10] P. D. Dobbelaere, K. Falta, L. Fan, S. Gloeckner, S. Patra, "Digital MEMS for Optical Switching", IEEE Communications Magazine, vol. 40, no. 3, pp. 88-95, 2002.

[11] A. Dugan, L. Lightworks, J. C. Chiao, "The Optical Switching Spectrum: A primer on Wavelength Switching Technologies", IEEE Telecommunication Magazine, 2001.

[12] S. Bregni, G. Guerra, A. Pattavina, "State of the art of Optical switching technology for All-Optical Networks", Communications World, Rethymo, Greece: WSES Press, 2001.

[13] K. Sakuma, H. Ogawa, D. Fujita, H. Hosoya, "Polymer Y-branching thermo-optic Switch for Optical Fiber Communication Systems", 8th Microoptics Conference Osaka, 2001.

[14] G. I. Papadimitriou, D. G. Maritsas, "Learning automata-based receiver conflict avoidance algorithms for WDM Broadcast-And-Select Star Networks", IEEE Transactions on Networking, vol. 4, pp. 407–412, 1996.

[15] G. I. Papadimitriou, A. S. Pomportsis, "Self-Adaptive TDMA protocols for WDM star Networks: A learning-automata-based approach", IEEE Photonics Technological Letter, vol. 11, pp. 1322–1324, 1999.

[16] Light Bend Mini 1x1, 1x2, 2x2 (Bypass) Opto-Mechanical Fiber optic Switch, URL: http://www.lasercomponents.com-/fileadmin/user_upload/home/Datasheets /agiltron/ sm_schalter/lb_mini_1x1_1x2_2x2.pdf.

[17] What is MEMS Technology?, URL: https://www.mems-net.org/mems/what-is.html.

[18] Marko Lackovic, Robert Inkret, Cristian Bungarzeanu, "Sensitivity Analysis of Transmission Network with Optical Packet Switching", URL: http://www. markolackovic.Com/2006Cost291Workshop.pdf.

[19] PSW 4401 Optical Switch Module, URL: http://www.fionix-.com/img/SOA%20 Optical %20Switch.pdf.

[20] 1x2 Opto-Mechanical Switch, URL: http://www.oemarke-t.com/product_info.php? products_id=44.

[21] Miniature Opto-Mechanical Switch, URL: http://www.jdsu-.com/product-iterature/momswitch_ds_cc_ae_042707.pdf.

[22] 1x1x2 Electo Optic Pager, URL: http://imghost.indiamart.com-/data1/C/B/MY460/LIGHTWAVELINK_Catalog.pdf.

[23] 1x1x2 Optical Switch, URL: http://lightwavelink. com.tw-/1x1x2-2f.htm.

[24] Q. Yang, K. Bergman, G.D. Hughes, F.G. Johnson, "WDM Packet Routing For High-Capacity Data Networks", Journal of Lightwave Technology, vol. 9, no. 10, pp. 1420-1426, 2001.

[25] Neha Sharma, D. Chadha, Vinod Chandra, "The Augmented Data Vortex switch fabric: An all-Optical packet switched Interconnection Networks with enhanced fault tolerance", Optical Switching and Networking, vol. 4, pp. 92-105, 2007.

[26] Neha Sharma, D. Chadha, Vinod Chandra, "Performance evaluation of the Augmented Data Vortex switch fabric: An all-Optical packet switched Interconnection Networks", Optical Switching and Networking, vol. 4, pp. 213-224, 2007.

[27] Roger D. Chamberlain, Mark A. Franklin, Ch'ng Shi Baw, "Gemini: An Optical Interconnection Networks for Parallel Processing", IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 10, pp. 1038-1055, 2002.

[28] Qimin Yang,, Keren Bergman, G. D. Hughes, F. G. Johnson "WDM Packet Routing for High Capacity Data Networks", Journal of Lightwave Technology, vol. 19, no. 10, pp. 1420-1426, 2001.

[29] A. K. Katangur, Yi Pan, M. D. Fraser, "Message routing and scheduling in Optical Multistage Networks using Simulated Annealing", International Symposium on Parallel and Distributed Processing, pp. 201-208, 2002.

[30] Yuanyuan Yang, Jianchao Wang, Yi Pan, "Permutation capability of Optical Multistage Interconnection Networks", Symposium on Parallel and Distributed Processing, pp. 125-133, 1998.

[31] Yi Pan, Chunming Qiao, Yuanyuan Yang, "Optical Multistage Interconnection Networks: New Challenges and Approaches", IEEE Communications Magazine, vol. 3, no. 2, pp. 50-56, 1999.

[32] M.A. Al-Shabi, and M. Othman, "A new algorithm for Routing and Scheduling in Optical Omega Networks", International Journal of the Computer, vol. 16, no. 1, pp. 26-31, 2008.

[33] Ravinder Yadav, Rinkle Rani Aggarwal, "Survey and Comparison of Optical Switch Fabrication Techniques and Architectures", Journal of Computing, vol. 2, no. 4, pp. 133-137, 2010.

[34] Qimin Yang, **"**A comparison study of Optical MIN Networks with parallel planes", Anaheim: ACTA Press, pp. 123-127, 2006.

[35] F. Abed, M. Othman, "Fast method to find conflicts in Optical Multistage Interconnection Networks", International Journal of the Computer, vol. 16, no. 1, pp. 18-25, 2008.

Paper Published

- Ravinder Yadav, Rinkle Rani Aggarwal, "Survey and Comparison of Optical Switch Fabrication Techniques and Architectures", Journal of Computing, vol. 2, no. 4, pp. 133-137, 2010.