DESIGN OF OPTICAL BURST SWITCHES BASED ON DUAL SHUFFLE-EXCHANGE NETWORK AND DEFLECTION ROUTING

Choy Man Ting

A thesis submitted in Partial

Fulfillment of the Requirements for

the Degree of Master of Philosophy

in

Information Engineering

July 2003

Acknowledgments

Many people support my efforts during my M.Phil study in CUHK. I would like to thank them for making this two-year study period a fruitful and rewarding experience.

First of all, I would like to express my deepest gratitude towards my supervisor Prof. Tony T. Lee for his continuous support, for his invaluable advice, and most importantly, for being a true teacher to me. I am grateful to him. Many thanks to Dr. Soung Y. Liew for his sincere encouragement, for his constructive criticism, and for the overall directions he provided.

I would also like to thank all the members in Broadband Lab in Department of Information Engineering: Mr. Deng Yun and Mr. Mui Sze Wai for their comments and continuous discussion on my research work. Mr. Zhang Liang, Mr. Man Wai Hung, Mr. Lam Lui Fuk, Mr. Lee Chi Ming and Mr. Wong Tze Chun for their various kinds of help during these two years without them these two years life will lack of many beautiful memory.

Finally, I am grateful to my family for morally supporting during all these years.

摘要

光突發交換因其易實現、高帶寬利用率和適用性而被看作是未來可提供兆兆比特交換的可行解決方案之一。在光突發交換中，多個數據包在源處積聚成一個較大的數據突發，然後再發送到目的端。通過在發送數據突發前送出它的控制信息并不等待其回應的方式，在中間節點處預留帶寬。因爲控制信號和數據是分開發送，在處理控制信息時，中間節點就不需要緩存來臨時儲存數據。這在光存儲技術還非常薄弱的時期是一個很大的優點。電路交換中的由于建立連接和在包交換中的頭信息的處理引起的過耗，在突發交換中分別用單程預留和突發積聚的方式解決。因此，光突發交換可以在需要的時候提供電路或者包交換，而只需調整預留方式和突發數據長度。

我們提出一種新的方案用雙重洗牌交換網絡來實現光突發交換機結構。這種網絡原本設計爲包交換機。在我們的設計中，此網絡的書入口和網絡內都不需要光緩存來存儲數據。數據可以通過一個路由標志通過該網絡。這說明該網絡不需要中央控制器因此有很好的可擴展性。可以證明改雙重洗牌-交換網絡的複雜度是 $N\log N$, 很接近香農極限。我們也相信改網絡的無緩存和异步的特性很適合光突發交

換。它的自路由的性質亦可在低丟包率和高流量的情況下降低系統的複雜度。

Abstract

The optical burst switch (OBS) has been highly regarded as a viable solution on providing terabit switching in the near future because of its easy implementation, high bandwidth utilization and flexibility. In optical burst switching, multiple packets are aggregated into a larger burst at the source before sending to the destination. Bandwidth is reserved in each intermediate node by one-way protocols in which data burst are sent after its control packet without waiting for the acknowledgment. As the control and data are sent separately, no buffering in the intermediate nodes are needed to store data temporarily while the control packet is being processed. This is highly preferable as optical RAM development is still in its early stage. In fact, the OBS compromises the circuit and the packet switching schemes. The overhead caused by connection setup in circuit switching and the headers in packet switching are remedied by means of the one-way reservation and the burst aggregation, respectively. Therefore, the OBS scheme can also provide circuit or packet switching when necessary, simply by adjusting the reservation scenario and the burst length of the scheme.

We propose a novel approach to implement the OBS switch fabric by the dual shuffle-exchange network (DSN). The DSN was originally developed to support packet switching. It uses the idea of deflection routing to solve the problem of packet contention. As a result, no buffer is needed to queue the packets at the input or inside the network. Packets can be properly routed through the DSN simply by a routing tag. This implies that the network does not need a central controller and thus is highly scalable. It can be shown that the complexity of DSN is of the order $N\log N$, which is above the Shannon's lower bound on switch complexity. It is believed that DSN's buffer-less and asynchronous natures are highly preferable in the OBS environment. Its self-routing property can substantially reduce the system complexity while still achieving low blocking probability and high throughput.

TABLE OF CONTENTS

LIST OF FIGURES

Number *Page*

Chapter 1

Introduction to Optical Burst Switching

With the rapid development on the dense wavelength division multiplexing (DWDM) technology, an optical fiber link can provide a dramatic amount of bandwidth available. On the other hand, optical switching technology is facing a lot of challenges. Challenges like how to avoid the O/E/O conversions needed, how to provide asynchronous transmission for variable sized packets and also the requirement of optical buffering in order to handle bursty traffic. Optical Burst Switching (OBS) is one such method for transporting traffic directly over a bufferless optical WDM network [1-12].

Circuit and packet switching have been very commonly used in our daily data communications. Burst switching, on the other hand, is less common. In circuit switching, a path between two stations has to be setup first in order to allow data to be transfer. Resource reservation is started from time you setup the connection to the time the connection is disconnected. In packet switching, data are broken down into small packets for transmission. Each packet is switched individually. The resources can be shared by different sources.

Circuit switching is advantageous when constant data rate is given or high bandwidth guarantee is needed. However, it is not suitable under bursty traffic conditions and the bandwidth utilization is low. Packet switching works well with variable rate traffic like data traffic, and can achieve higher utilization.

Circuit switching uses two-way reservation schemes that would result in high latency. While packet switching has a large buffer requirement and complicated control and strict synchronization issues. Optical burst switching (OBS) has been proposed to achieve the balance between the circuit switching and the packet switching. It is based on one way reservation protocols which a data burst (a number of packets) follows a corresponding control packet without waiting for an acknowledgment. Table 1-1 compares the three switching paradigms qualitatively.

Table 1-1 Comparison among the three switching paradigms.

Optical Switching Paradigms	Bandwidth Utilization	Latency (set-up)	Optical Buffer	Overhead	Adaptivity (traffic & fault)
Circuit	Low	High	Not required	Low	Low
Packet	High	Low	Required	High	High
OBS	High	Low	Not required	Low	High

1.1 OBS Network Architecture

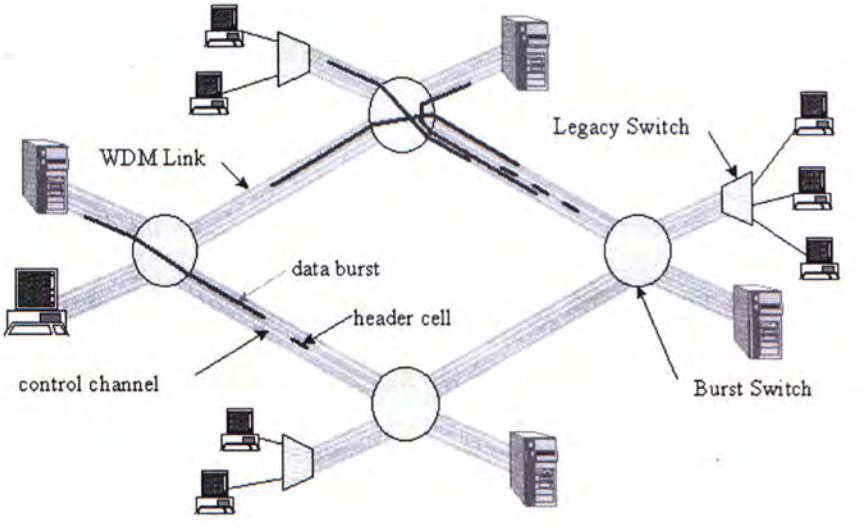


Figure 1-1 OBS Network Architecture

In optical burst switching networks, the source would combine small sized packets, which destined to the same destination, to form a large burst. This burst would be switched through the network all-optically. This burst would have a common header, which would be sent before the data burst. The header would contain information like, the destination address, burst length and the time when the data burst start transmission. The header is sent before and separately with data burst is because in this way, the data burst can be buffered at the source while switches have time to configure themselves according to the header information. The data burst would start transfer without waiting for acknowledgements from the switches. Figure 1-1 shows an OBS network. It consists of edge nodes (pc and server) and core nodes (the green circles). An OBS network consists of optical burst switches interconnected with WDM links. Each WDM fiber links can carry a large

number of wavelengths, which can also be seen as a channel. The control packet can be transmitted in the switch in band with the data burst, or be transmitted separately in a different control channel.

Figure 1-2 illustrates a possible structure of a core node. There are incoming and outgoing fiber links, each of which has a number of wavelengths for carrying data bursts (solid lines) and one additional wavelength for carrying control packets (dotted lines). Every control packet is processed by the electronic control module inside an OSN, which generates appropriate control signals to set up the wavelength converters, FDL buffers, and switching fabric. The optical switching fabric switches each burst on an incoming wavelength as it arrives (i.e., without having to synchronize it with other incoming bursts).

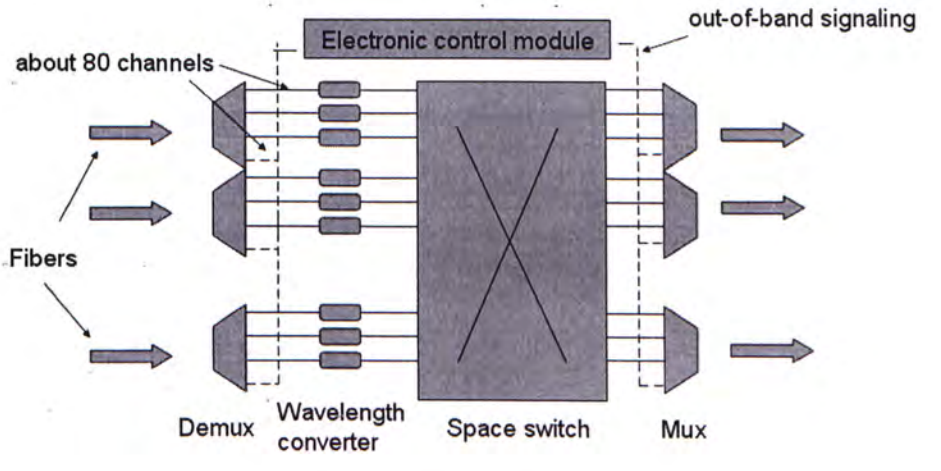


Figure 1-2 A possible structure of a core node

1.2 Offset Time and Reservation Schemes

Burst switching techniques are well developed in electronic area, a number of signaling schemes have already been proposed to reserve network resources in different ways. One of them is that when a source wants to send data in the network, it sends a request signal first. This request signal is processed by all the switches along its path. This request would be accepted when all the switches agree to carry this data transmission. A confirmation would be sent back to the source and the source would start transmitting the data. Another approach is that the source can send a request to the network and does not wait for the confirmation. After sending the request, the source would start transmitting the data immediately. In this way, as some of the switches along its path may not be able to carry the data traffic and the data would be dropped there. These two kinds of signaling scheme are adopted in optical network.

Each of two signaling schemes mentioned above have their own disadvantages when applied to on optical network. The first scheme is similar to a tell-and-wait scheme. Tell-and-wait (TAW) systems would have the problem of high latency and low bandwidth utilization. The second scheme is similar to a tell-and-go (TAG) system. In TAG system, as the data burst is sent immediately after the control header is sent, while the header is being processed at the switch, the data burst would have to wait at the intermediate switching nodes. This requires optical buffering in each switching node. It is not appealing as optical buffering is still

immature. Therefore, an intermediate scheme known as Just Enough Time (JET) was proposed in [1].

In JET, in order to avoid buffering at the intermediate switches inside the network, control header and the data burst are sent separately by an offset time, as shown in Figure 1-3. The control header is processed by each switch node inside the network while the data can be buffered at the source. This implies that the offset time should be equal or larger than the total processing time by all the intermediate nodes. In this way, buffering at intermediate switch node can be avoided. The control signal can also carry the information on the duration of the burst so that the switch node can know when it would become available again and can accept another burst. This technique is known as Delayed Reservation (DR) [1]. A further improvement of the JET scheme can be obtained by reserving resources at the optical burst switch from the time the burst arrives at the switch, rather than from the time its control packet is processed at the switch.

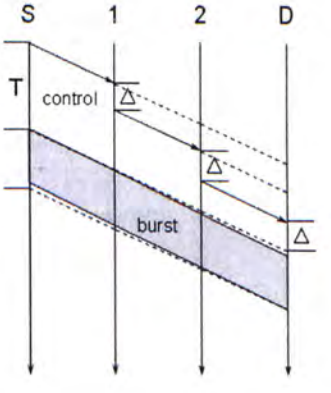


Figure 1-3 The use of offset time in OBS.

In [1] a variation of JET was proposed which supports quality of service. Data belonging to high-priority class would have a larger offset time. A burst with larger offset time implies that it has the right to reserve resources at the first place as other bursts cannot reserve them now as they do not have such a large offset time.

1.3 Research Objectives

In this thesis, we investigate the OBS network architecture, its advantages, issues, core switching network designs, reservation policies and related concepts. Our research is concentrated on Core Switching Network Designs in OBS networks.

We propose a novel approach to implement the OBS switch fabric by the Dual Shuffle-exchange Network (DSN). The DSN was originally developed to support packet switching. It uses the idea of deflection routing to solve the problem of packet contention. As a result, no buffer is needed to queue the packets at the input or inside the network. Packets can be properly routed through the DSN simply by a routing tag. This implies that the network does not need a central controller and thus is highly scalable. It can be shown that the complexity of DSN is of the order $N\log N$, which is above the Shannon's lower bound on switch complexity. We discovered that DSN's buffer-less and asynchronous natures are highly preferable in the OBS environment. Its self-routing property can

substantially reduce the system complexity while still achieving low blocking probability and high throughput.

1.4 Overview

This report consists of six chapters. This chapter has outlined a brief introduction to optical burst switching as well as the research objectives. Chapter 2 covers prior art on core switching fabric designs in optical networks. Chapter 3 proposes the concept of implementing Optical Burst Switching routers using Dual Shuffle-exchange Network. Chapter 4 introduces some schemes to handle output multiplexing and to improve the performance of the proposed architecture. In Chapter 5, another approach, named vertical expansion, is also proposed to improve the performance. As 8×8 MEMS switches are available, we have some modifications on our existing switch in order to make use of these switches. Analysis on these modifications is also discussed in Chapter 5. Chapter 6 concludes the thesis.

Chapter 2

Prior art on Optical Burst Switching

In this chapter, we describe an optical burst switch architecture proposed by Jonathan Turner. This switch architecture gives us appealing performance results. However, it requires a lot of expensive optical components and complicating routing algorithms.

2.1 WDM crossbar architectures

The scalable burst switch architecture proposed is shown in Figure 2-1. It is based on a three-stage Clos (or multistage Benes) interconnection network. External links are connected to a set of I/O modules (IOM). The burst switch elements (BSE) are $d \times d$ switching elements and this switch architecture can support up to d^2 external links (each carrying h WDM channels). This implies each stage can have d BSEs.

There are two wavelength converting switch designs, both use optical modulators and tunable lasers to transfer a signal from an input wavelength to a tunable output wavelength. The first design also use optical crossbars to provide space division switching, while the second substitutes wavelength grating routers (WGR) for the crossbars.

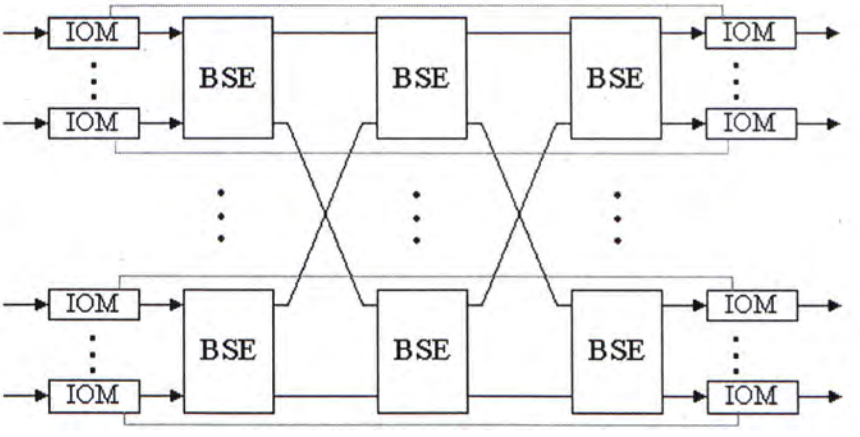


Figure 2-1 A Scalable Burst Switch
Architecture proposed by J.Turner [3]

2.2 Switch Based on Optical Crossbars

Figure 2-2 shows the first wavelength converting switch design. Each fiber is connected to a de-multiplexer which separates the different wavelength channels. The separated wavelength channels will then propagate through some Tunable Wavelength Converters, which would quickly change the input burst to any available wavelengths channels out of its destined output port. The converted burst would then propagate through an $h \times d$ crossbar switch. The $h \times d$ crossbar switch can be composed by a number of d x d crossbar switches. In this way, the expensive large-sized crossbar switch can be avoided. As there can be more than two bursts destined to the same output port, each output of the crossbar is followed by a passive wavelength multiplexer. As long as these bursts are with different wavelengths, they can share the same multiplexer and crossbar output.

To route an incoming burst to its destined output, the wavelength of the incoming burst is converted to any available wavelengths available at the destination. The crossbar is configured to allow burst signal to reach its destined output port. The crossbar is a non-blocking switch which can ensure no internal blocking if there are enough available wavelengths at the output destination to convert to. As there may be no internal buffering at the switch, bursts that can't find free wavelength at its destined output would be dropped. But since the number of wavelength channels in each fiber is high, the probability of dropping is quite low.

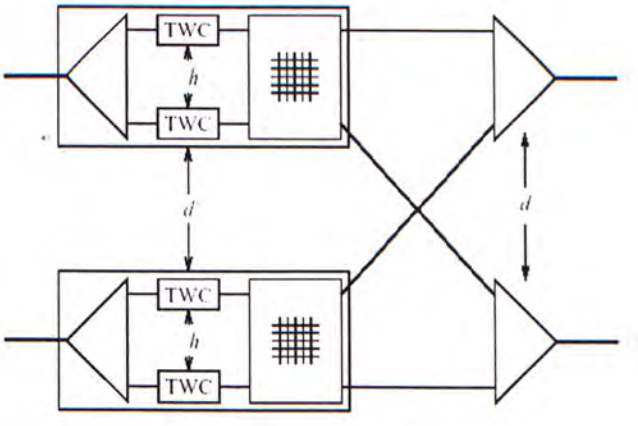


Figure 2-2 Wavelength converting switch using Tunable Wavelength Converters (TWC), Optical Crossbars and Passive Multiplexors and Demultipleors [3]

2.3 Switch Based on Wavelength Grating Routers

An alternative design for a wavelength converting switch is shown in Figure 2-3. This design uses a passive wavelength grating router (WGR) to replace the optical crossbars used in the previous design. Thus, the tunable

wavelength converters are the only active components. Since the wavelength routers have h inputs and h outputs, there are h/d fibers connecting each input section with each output section. For $h = 256$ and $d = 8$, there will be 32 fibers connecting each input section with each output section. In this design, the tunable wavelength converters serve two purposes.

First they are used to switch signals to different wavelengths so that there will be no wavelength contention at the output fiber links. Secondly, they switch the signals to different wavelengths so that the Wavelength Grating Router (WGRs) can provide space switching according to this given information. By switching the wavelength to the set of h/d wavelengths that destined to the desired output, the signal can be forwarded to its desired output port. The drawback of this design is the number of choice of wavelengths is reduced as in the previous design; we have h wavelengths to choose from while we only have h/d to choose from in this design. This may lead to extra blocking. That is the destined output address still have free wavelength channels available, not at the input section, all the available wavelength channels are used. Another drawback of this design is that it requires complex algorithm to control the tunable wavelength converters to serve the two purposes mentioned above.

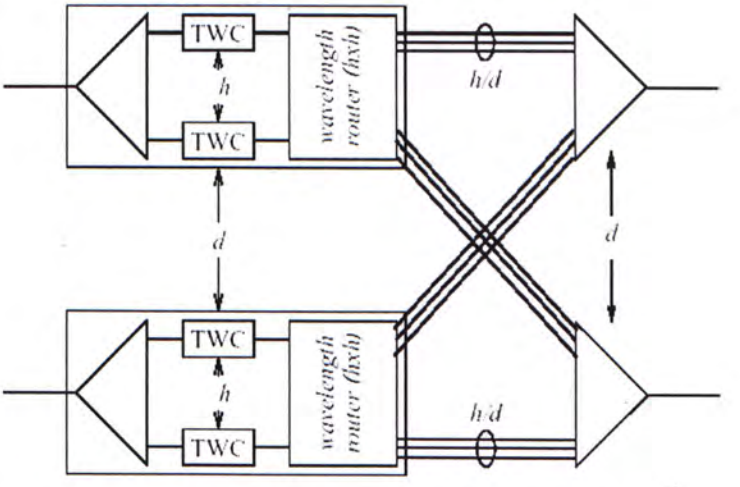


Figure 2-3 Wavelength switch Tunable Wavelength Converters (TWC) and Passive Wavelength Grating Routers (WGR).[3]

Chapter 3

Proposed Architecture

In this project, we propose a novel approach to implement the OBS switch fabric by the Dual shuffle-exchange network (DSN) [13]. The DSN was originally developed to support packet switching. It uses the idea of deflection routing to solve the problem of packet contention. As a result, no buffer is needed to queue the packets at the input or inside the network. Packets can be properly routed through the DSN simply by a routing tag. This implies that the network does not need a central controller and thus is highly scalable. It can be shown that the complexity of DSN is of the order $N\log N$, which is above the Shannon's lower bound on switch complexity. We discovered that DSN's buffer-less and asynchronous natures are highly preferable in the OBS environment. Its self-routing property can substantially reduce the system complexity while still achieving low blocking probability and high throughput.

3.1 Basics of Dual Shuffle Exchange Network

Figure 3-1 shows an 8 × 8 Dual Shuffle-exchange Network with 5 stages. We define N as the number of input ports of the switch and L be the number of stages, $N = 8$ and $L = 5$ in this case. The 4 × 4 switching

modules inside the network are interconnected in this way such that packets from any input ports can self-route to its destination port in $n = \log_2 N$ stage if there is no packet contention. This connection pattern also permits an error-correcting routing algorithm. When contention occurs inside a 4 × 4 switching module, the loser packet will be deflected to one of the idle output ports available. A one-stage routing instruction will be added to this packet based on which output port this packet is deflected. By successfully following this routing instruction in the next stage, the deflected packet can return to the state where it was deflected and resume its routing. Successive deflections can also be corrected by this algorithm. Figure 3-2 shows the state-transition diagram of this error-correcting algorithm. Each state represents the number of remaining stages the packet still have to go through until it reaches its destination. Theoretically, all loss probability requirements (P_{loss}) can be achieved by means of increasing L and it has been shown [13] that L is analytically bounded by

$$L \leqq 2.793n - 3.554 \ln(n+1) + 3.554 \ln P_{loss}^{-1} + 1.162 \qquad (3.1)$$

Since each stage consists of $N/2$ switch modules, the complexity of the DSN for a given P_{loss} is therefore $N\log N$.

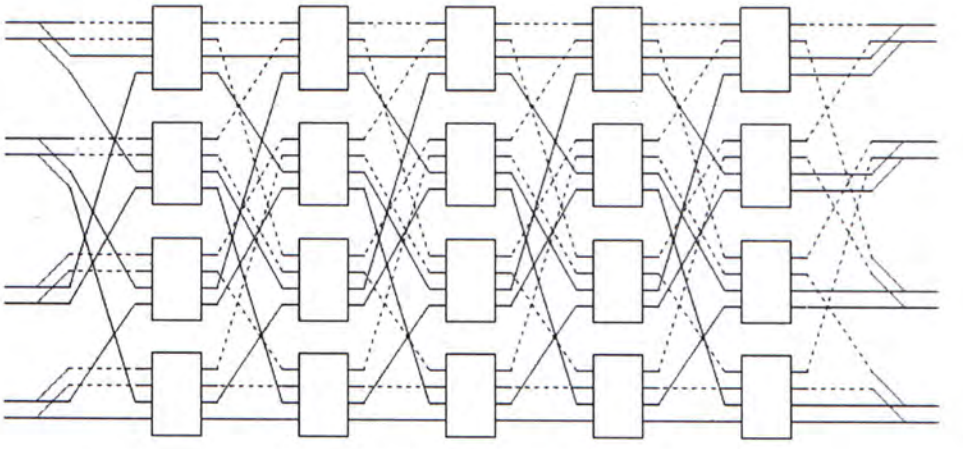


Figure 3-1 An 8×8 Dual Shuffle-exchange Network

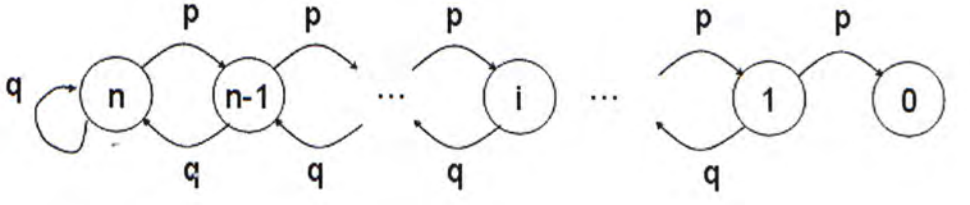
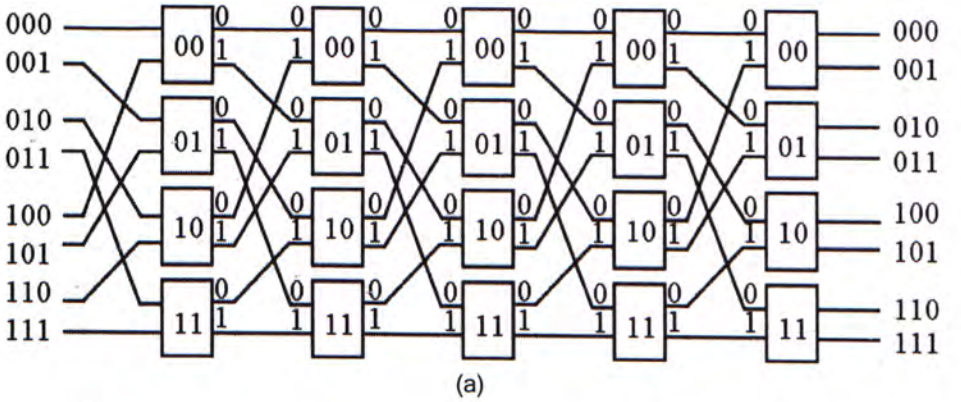


Figure 3-2 Markov Chain for bounding L [13]

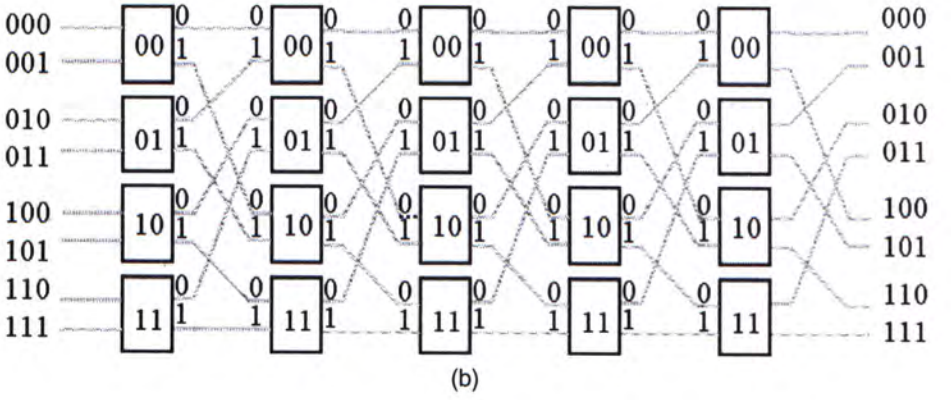
3.2 Dual Shuffle-exchange Network

One way to look at the dual shuffle-exchange network is to consider it as being constructed of two sub-networks, a shuffle network (SN) and an unshuffle network (USN) -- the mirror image of the shuffle network, as illustrated in Figure 3-3. Both SN and USN networks have the self-routing property. A routing tag would be given to each arriving packet based on the packet's destination address. Normally, the routing is simply the binary representation of packet's destination address port. One bit of the routing tag is examined in each stage. For example, if the destination address is

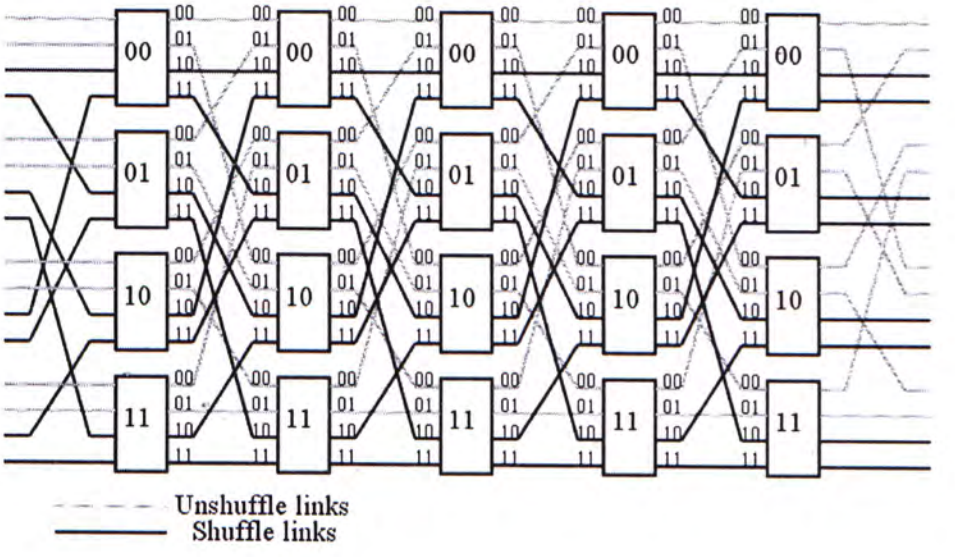
010, then the first bit is examined in the first stage. The packet would be forwarded to the upper link when bit '0' is read and would be forwarded to the lower link when bit '1' is read. In the second stage, the second bit is examined and the packet is forwarded to the lower link. Then '0' is read at the third stage and the packet is forwarded to the upper link and reaches its output destination address 010. However, contention might occur at every stage. Typical solution is to deflect one of the incoming packets to another output port and that packet will have to be routed again starting from the first bit. Consider a packet needs one more stage to reach its destination and is unluckily deflected; the penalty of this collision could be very large. Therefore, Dual Shuffle-exchange Network is designed so that when a packet is deflected from node *i* to node *j*, we must have a link in the reverse direction connecting node *j* to node *i*, the packet can travel back to node *j* from node *i*, correcting the deflection in one step, as shown in Figure 3-4.



(a)



(b)



(c)

Figure 3-3 Construction of a dual shuffle network using a shuffle-exchange and a unshuffle exchange networks: (a) Shuffle-exchange network. (b) Unshuffle-exchange network. (c) Dual shuffle-exchange network.

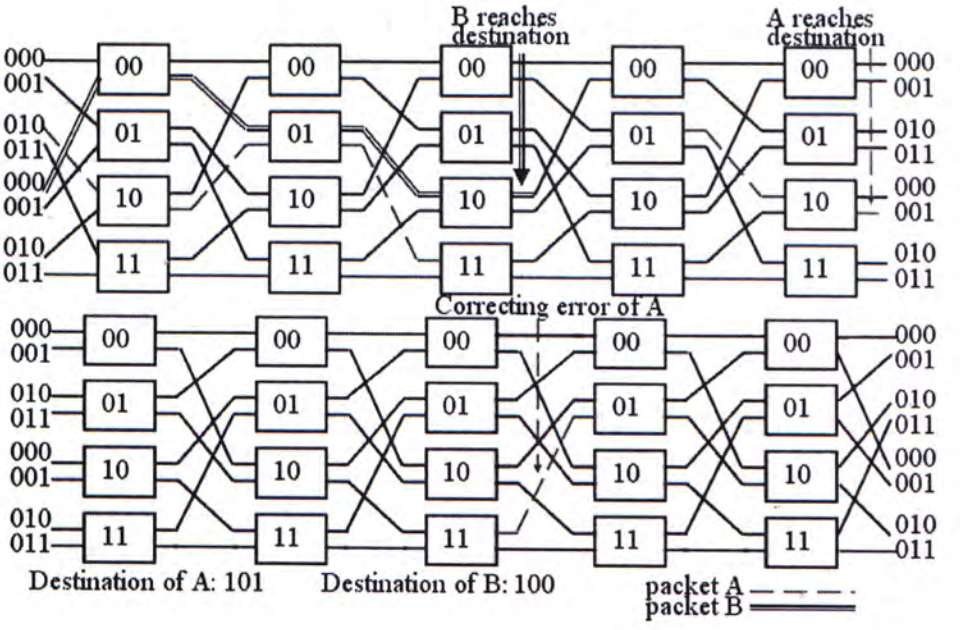


Figure 3-4 Correcting deflection in one step

3.3 Proposed Architecture based on DSN

Figure 3-5 shows our proposed switching node architecture. There are d incoming and d outgoing optical fibers; each contains h+1 wavelength channels. One of the channels is reserved for control signals. Each of the d input fibers is connected to an optical demultiplexer that separates the different wavelength channels before propagating through the central switching fabric. The central switching fabric is a $dh \times dh$ DSN switch. Outputs of the DSN switch are then connected to distinct multiplexers and wavelength converters to ensure no wavelength contention in each output fiber. This switch can be operated to support a number of optical burst switching protocol, like Just-Enough-Time (JET) signaling protocol. However in this thesis, simulation are based on the Just-In-Time (JIT) [14]

signaling protocol. In JIT, the basic switching modules inside the switch are directly reserved for the incoming burst immediately after the arrival of the request message, and remain until the arrival of a release message.

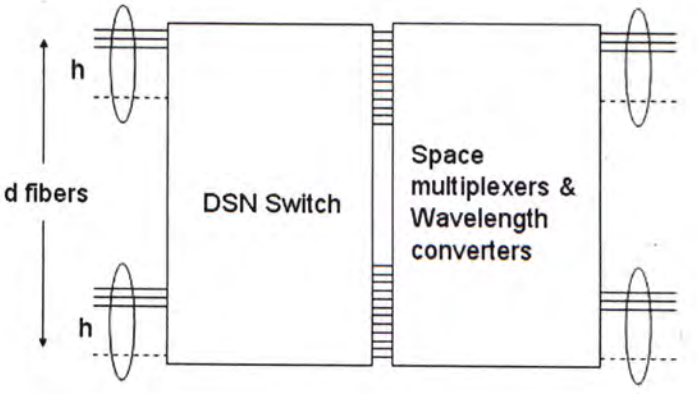


Figure 3-5 Block diagram of the proposed architecture.

3.4 Analysis on blocking due to output contention

There are two ways to have blocking (or loss) in this DSN switch. One of them is due to the insufficient number of stages available. In this case, some bursts are deflected too many times in the switch and cannot fully-routed at the end of the switch. Another one happens when there are already too much bursts occupying a specific output fiber. In this case, even if the burst can fully routed in the switch, it does not have a free wavelength channel to leave the switch. This is called output contention. In this section, we will discuss the blocking due to output contention. Blocking due to insufficient number of stages will be discussed in Chapter 4.5.

Assume the traffic on each input port is independent and have the following arriving pattern.

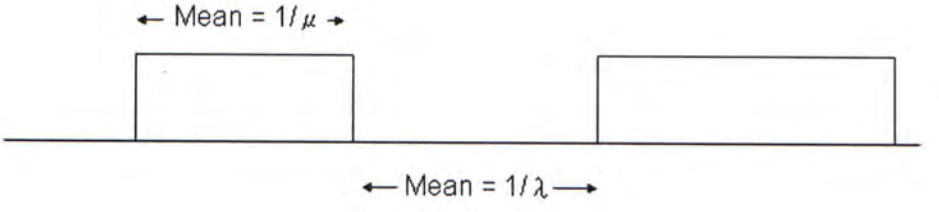


Figure 3-6 On-Off burst arrival

Durations times of burst and idle periods are exponentially distributed with mean $1/\mu$ and $1/\lambda$ respectively. Simple models like Poisson process can't capture the important characteristics of the sources, we model it using the On-Off model as shown in the following figure,

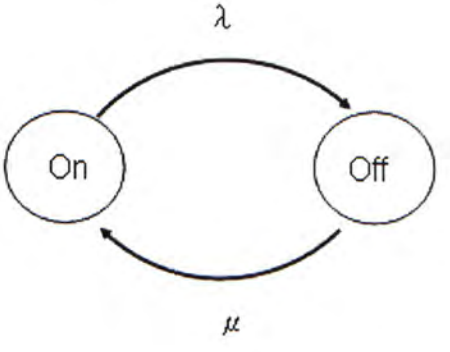


Figure 3-7 On-Off model

$$\text{Input loading} = \rho = \frac{\frac{1}{\mu}}{\frac{1}{\mu}+\frac{1}{\lambda}} = \frac{\lambda}{\lambda+\mu} \qquad (3.2)$$

Define d be the number of fibers, h be the number of wavelengths in each fiber, then the number of active input ports as seen by an arriving burst can be calculated by the following transition diagram.

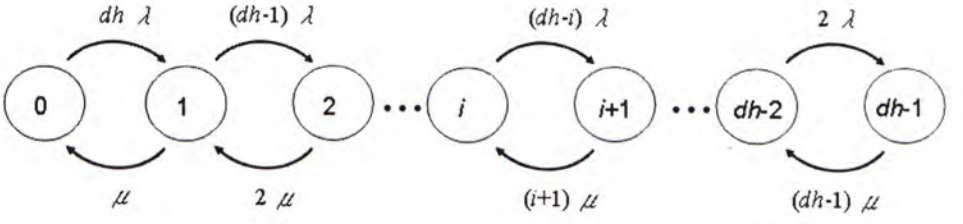


Figure 3-8 Transition diagram for On-Off source

The states of the diagram depict the number of active input ports. Then we have the limiting probability π_i

$$\pi_i = \frac{\binom{dh}{i}\mu^{dh-i}\lambda^i}{(\mu+\lambda)^{dh}-\lambda^{dh}} = \frac{\binom{dh}{i}(1-\rho)^{dh-i}\rho^i}{1-\rho^{dh}} \tag{3.3}$$

Given we have i active input channels out of dh, the probability that a specific output fiber have k active channels follows a binomial distribution ($k > h$ implies that k-h bursts have already been dropped):

$$P(i,k) = \binom{i}{k}\left(\frac{1}{d}\right)^k\left(1-\frac{1}{d}\right)^{i-k} \tag{3.4}$$

Therefore the arriving burst would be dropped if its destination fiber has equal or more than h active channels and the probability equals

$$P_{drop} = \frac{1}{d}\sum_{i=h}^{dh-1}\left[\pi_i \sum_{k=h}^{i} P(i,k)\right]$$
$$= \frac{1}{d\left(1-\rho^{dh}\right)}\sum_{i=h}^{dh-1}\left[\binom{dh}{i}(1-\rho)^{dh-i}\rho^{i}\sum_{k=h}^{dh-1}\binom{i}{k}\left(1-\frac{1}{d}\right)^{i-j}\left(\frac{1}{d}\right)^{j}\right] \quad (3.5)$$

The analysis and simulation results are presented in Figure 3-9.

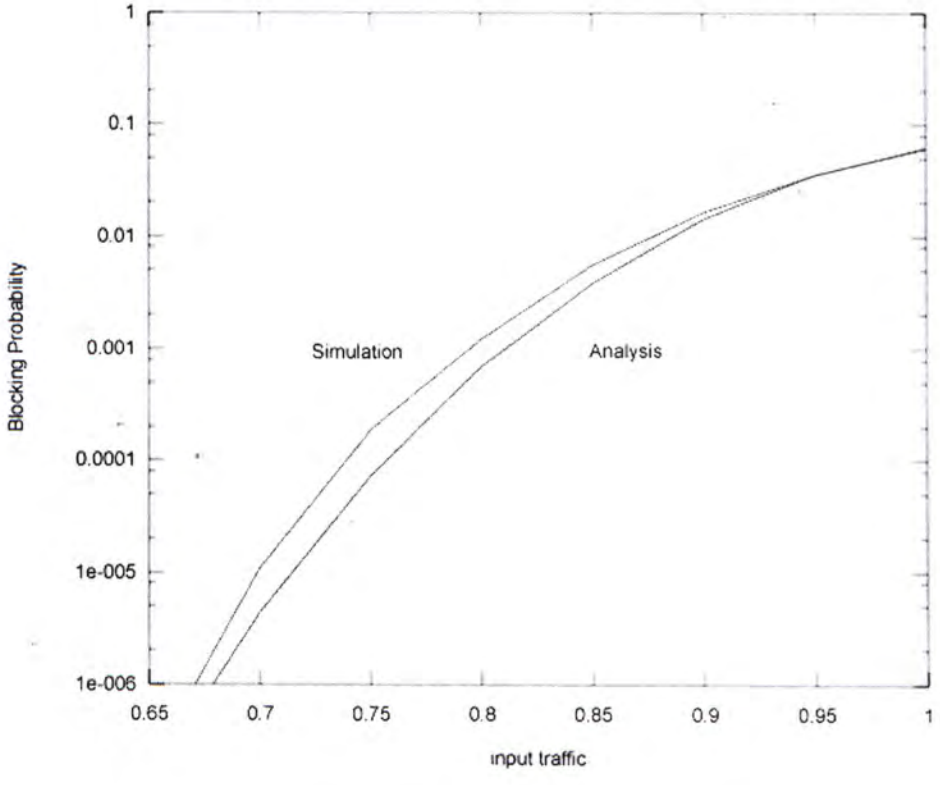


Figure 3-9 Analysis and Simulation Results on blocking due to output contention

3.5 Implementation issues on the 4 × 4 switching module.

For each 4 × 4 switching module, four output links are connected to the next stage while another four output links are connected to the space multiplexers. Therefore, 4 additional 1 × 2 switches are needed as illustrated in Figure 3-10.

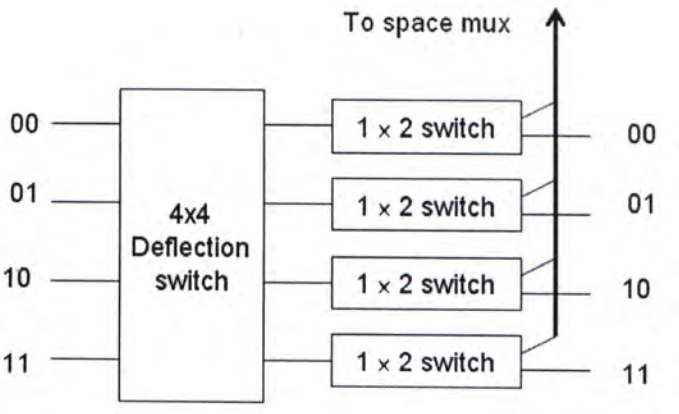


Figure 3-10 Block diagram of a 4 × 4 switch module. [13]

If we implement the 4 × 4 switching module using a non-blocking crossbar switch, a total of 16 cross-points are needed. It was showed that [13] in the packet switching environment, a two-stage banyan switch as shown in Figure 3-11(a), is a possible alternative. Granted that a burst may be deflected because of "internal conflict" even when there is no contending burst for the same "external output" (see Figure 3-11(b)), the increase in the overall deflection probability is actually quite small. The non-blocking and banyan design alternatives will be compared in more detail in the next subsection.

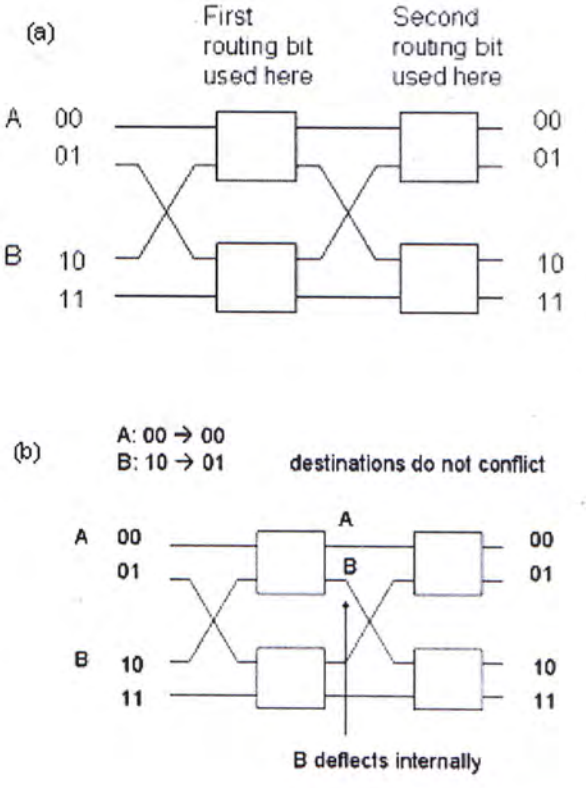


Figure 3-11 (a) A 4 × 4 banyan deflection switch; (b) An example of internal conflict when there is no output conflict. [13]

3.6 Analysis: Non-blocking versus banyan.

As bursts are separated to the shuffle and unshuffle plane, the internal traffic model in each plane is changed as follows,

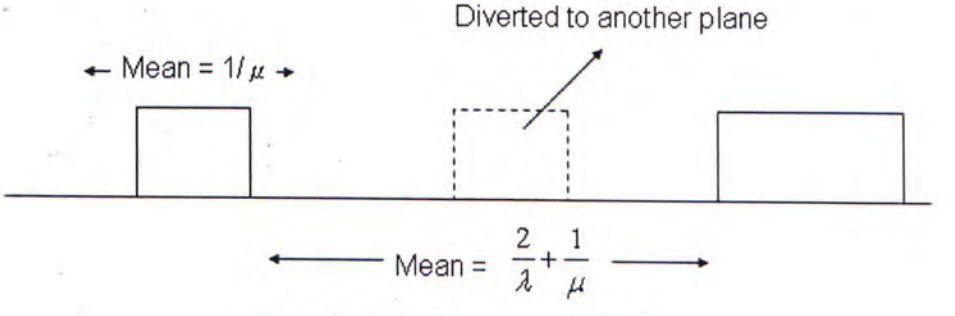


Figure 3-12 On-Off burst arrival for the two internal planes

Now the new idle time is formed by adding two idle time and one burst length. The new idle time is not exponential distributed but a hypoexponential distribution. However, for simplicity reason, we assume the newly formed idle time is still exponential distributed. Let $\frac{1}{\alpha}$ be the mean idle time for internal traffic

$$\frac{1}{\alpha} = \frac{2}{\lambda} + \frac{1}{\mu} \tag{3.6}$$

$$\alpha = \frac{\mu\lambda}{\lambda + 2\mu} \tag{3.7}$$

The on-off model is modified as follows for internal planes,

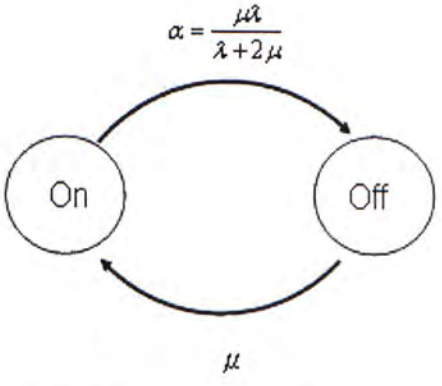


Figure 3-133 On-Off model for internal planes

Let us first derive the deflection probability of an internally non-blocking switching module. When a new burst arrives at this non-blocking switching module, it will see at least one output port is idle, or equivalently 0 to 3 output ports are being occupied. Assuming each burst is equally likely to destine to one of the four output ports, the burst length is exponential with mean $1/\mu$ and idle time is exponential with mean $1/\alpha$, Figure 3-14 depicts a transition diagram for this 4 × 4 non-blocking

switching module. The states of the transition diagram are the number of busy output ports seen by a new arriving burst.

$$\pi_i = \frac{\binom{4}{i}\mu^{4-i}\alpha^i}{(\mu+\alpha)^4 - \alpha^4} \tag{3.8}$$

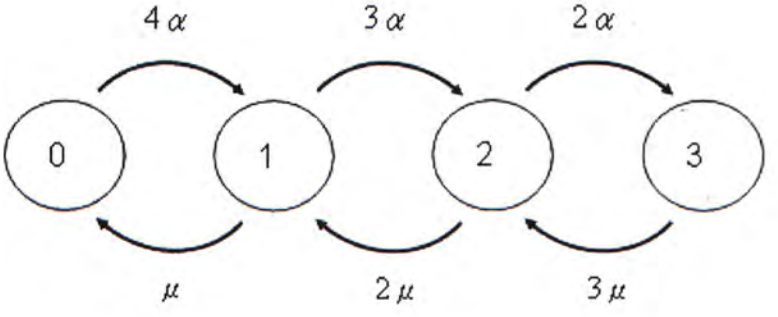


Figure 3-14 Transition diagram for Non-blocking switch.

After obtaining the limiting probabilities π_i, the deflection probability for non-blocking switching module can be calculated as follows

$$\begin{aligned} q_n &= 1 - \left(\pi_0 + \frac{3}{4}\pi_1 + \frac{2}{4}\pi_2 + \frac{1}{4}\pi_3\right) \\ &= 1 - \frac{\mu(\mu+\alpha)^3}{(\mu+\alpha)^4 - \alpha^4} \\ &= 1 - \frac{\mu\left(\mu + \dfrac{\mu\lambda}{\lambda+2\mu}\right)^3}{\left(\mu + \dfrac{\mu\lambda}{\lambda+2\mu}\right)^4 - \left(\dfrac{\mu\lambda}{\lambda+2\mu}\right)^4} \\ &= 1 - \frac{16-8\rho}{16-\rho^4} \end{aligned} \tag{3.9}$$

Similarly, the transition diagram in Figure 3-15 depicts one of the cross-points in Figure 3-11. With the same set of parameters as above, we obtain the deflection probability for the whole banyan switch as follows

$$\begin{aligned} q_b &= 1-\left(\pi_0+\frac{1}{2}\pi_1\right)^2 \\ &= 1-\left[\frac{\mu(\mu+\alpha)}{(\mu+\alpha)^2-\alpha^2}\right]^2 \\ &= 1-\left(\frac{4-2\rho}{4-\rho^2}\right)^2 \end{aligned} \tag{3.10}$$

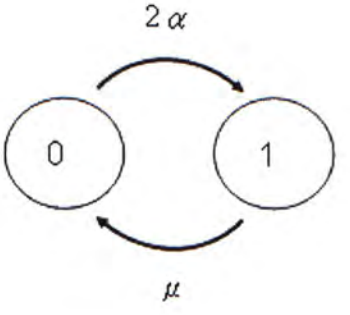


Figure 3-15 Transition diagram for a 2 × 2 cross-point.

Figure 3-16 and 3-17 plots the deflection probability of the two switching node designs. And also by differentiating the difference between q_n and q_b, we obtain the maximum absolute difference is no bigger than 0.1142 at ρ = 0.571.

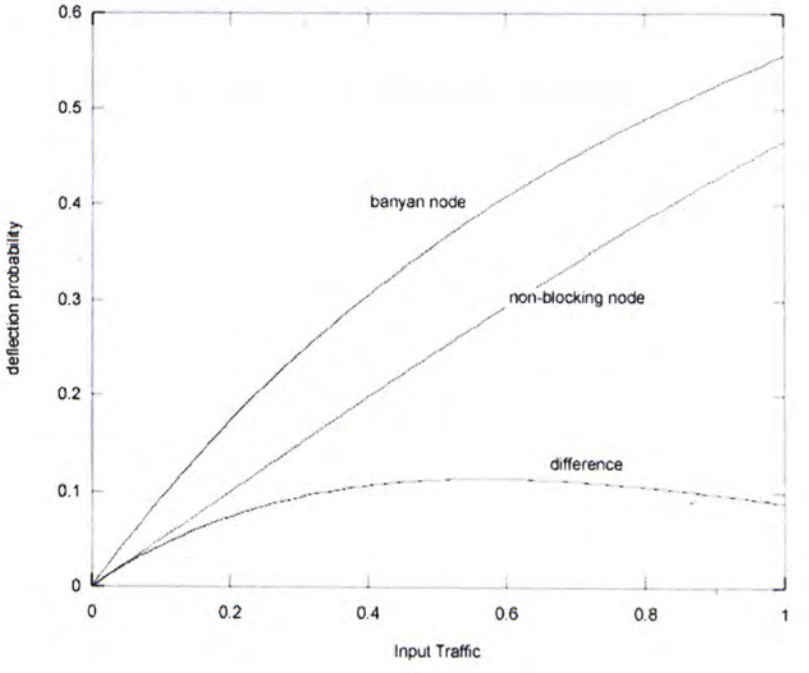


Figure 3-16 Deflection probability on banyan node and non-blocking node

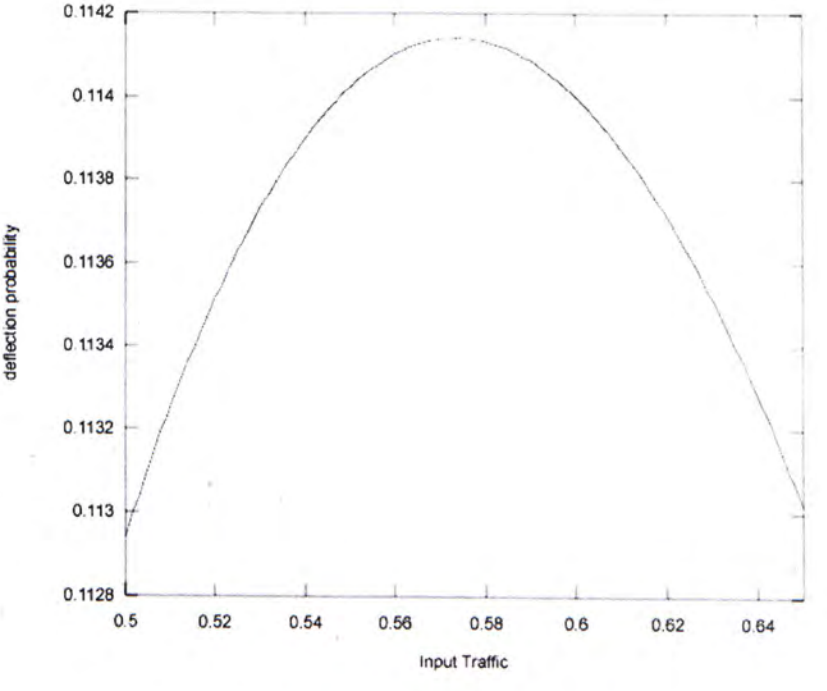


Figure 3-17 Difference in deflection probability between the two node designs

Chapter 4

Output multiplexing

With the deflection routing property of the DSN, when a burst has finished its routing at any stage, it can be outputted from the network. Therefore for each fiber, we have a multiple of h internal output ports but only h output wavelength channels available. These internal output ports have to be multiplexed before it can be forwarded to a free wavelength channels. In this section, we propose three implementation schemes to handle the output multiplexing. The analytic results on performance between banyan and non-blocking nodes are validated by simulation here also.

4.1 First Scheme.

Consider a system with 8 input/output fibers and number of wavelength channels contained in each fiber is 32. The first fiber would be connected to the ports of DSN having port numbers starting from 0 to 31, that is 0000 0000 to 0001 1111. As we can see, the last 5 bits are don't care bits. One way to assign routing tag to control packet is to have three actual routing bits and 5 don't care bits (in reality, each routing bit here should be leaded by a bit carrying information on which plane this control packet should go,

but for simplicity reason, we just omit these bits here). When a don't care bit is read by a switching node, the node can forward the control packet to one of the two output ports of the same plane. However, it is not necessary for a fiber having continuous port numbers. We can assign ports with port number 0000 0000, 0000 1000, 0000 2000 and up to 1111 1000 to the first fiber. In this way, the don't care bits would becomes the first 5 routing bits. As control packet can reach the destination starting from any input port of the DSN switch, the first five don't care routing bits can be omitted, resulting in routing tag with only three bits, which carry fiber information only. This assignment scheme is very appealing as shorter routing tag should require smaller number of stages in general.

To handle output multiplexing for this assignment scheme, internal output ports with the same port number address would share the same output wavelength channel. These output ports would be connected to a multiplexer and then follow by a passive wavelength converter which performs fixed wavelength conversion. In this way, we have h multiplexers for each fiber and each multiplexer would be serving $(L\text{-}D+1)$ internal output ports, as shown in Figure 4-1. Whenever a fully-routed control packet finds its corresponding multiplexer being occupied, this control packet would have to be deflected as if it is deflected due to contention inside the network. When this control packet is fully-routed again, it can try another multiplexer. The multiplexer has to give feedback control

signal to each switching node so that the node can know whether the multiplexer is being occupied.

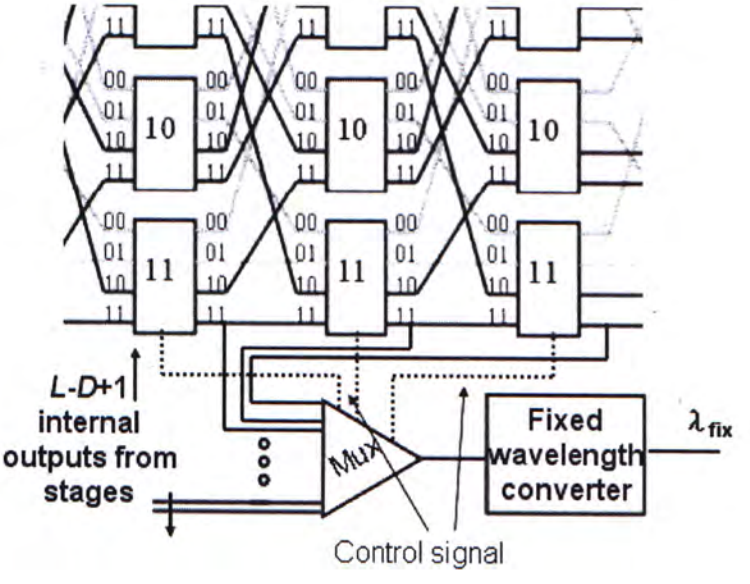


Figure 4-1 Scheme 1 hardware implementation

As deflected packet would return to the original state where it was being deflected, in actual implementation, it is not wise to have internal output ports with the same port number address sharing the same output wavelength channel. Consider Figure 4-2, a control packet has finished it routing and it wants to exit from port 1111 (dashed line). However, the multiplexer is occupied. The packet would be deflected and then come back to the same state. It is highly likely that the multiplexer is still being occupied. Therefore, it is better to have internal output ports with *different* port number addresses sharing the same output wavelength channel.

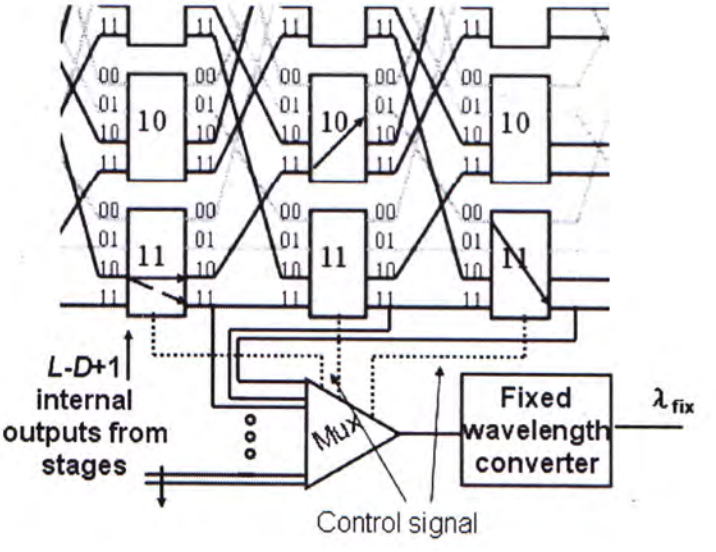


Figure 4-2 deflected control packet
would revisit the same state

4.2 Simulation on the first scheme.

Figure 4-3 plots the simulated loss probability as a function of the number of available stages of the switch. The input traffic loadings in the figure are 0.5 (d = 8, h = 128) and 0.571 (d = 32, h = 32), which brings us the maximum difference in loss probability between the non-blocking nodes and the banyan nodes as described in the previous section. From the graph, we see that the performances of the two node designs are almost identical for traffic load of 0.5. The number of stages needed for a fixed loss probability in the banyan design is slightly higher than that in the non-blocking design for input load of 0.571. This favorable observation was already suggested in Figure 4-4. In packet-based Dual Shuffle-exchange Network, the performance difference is negligible for small n.

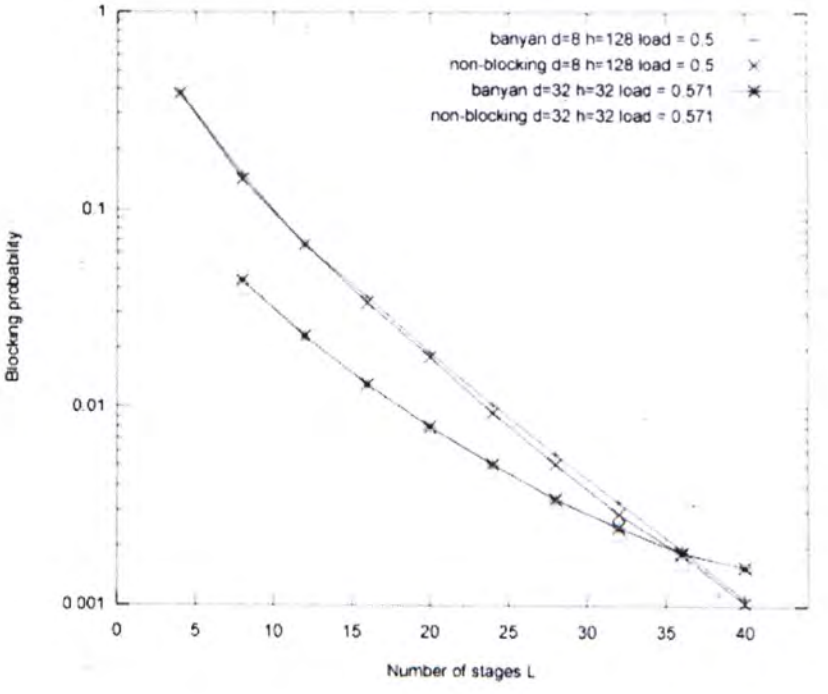


Figure 4-3 (scheme 1) Loss probability versus number of stages for banyan and non-blocking node.

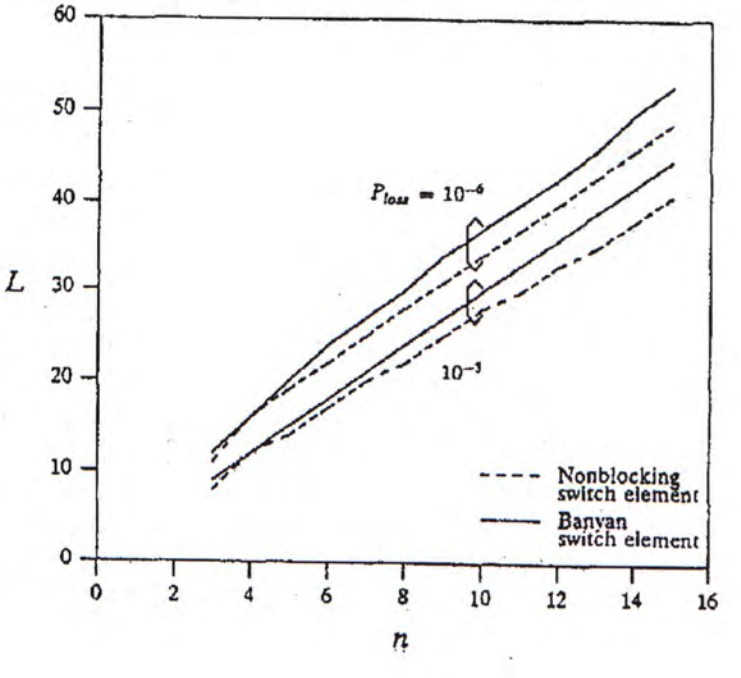


Figure 4-4 (packet DSN) L versus n, fixing P_{loss} at 10^{-3} and 10^{-6} [13]

On the other hand, this design does not inherit the characteristic of rapid drop in loss probability with little increase in stages. In Figure 4-5, P_{loss} drop significantly with the increase in L. However, we cannot see this favorable feature in our simulation. The loss probability in our simulation is simply decreasing linearly. It is suspected that this drawback is due to the difficulty in finding a free multiplexer although there are free wavelengths available in that fiber.

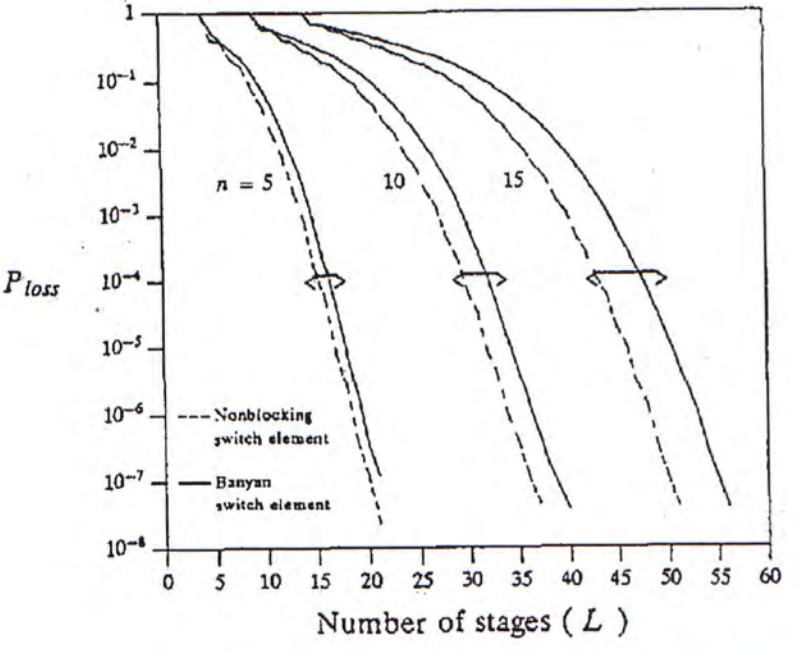


Figure 4-5 P_{loss} versus L for various n; packet DSN [13]

We have simulation on the loss probability versus the input load with various setups. Figure 4-6 shows simulation results of (d, h) = (8, 128), (8, 64) and (16, 64). The number of stages tested here is 20. It is observed that, for various numbers of stages, the differences among those fiber-

wavelength combinations are not remarkable. This observation is not encouraging as the loss probability is expected to be reduced notably when the wavelengths are doubled, as suggested by the Erlang B formula. This further suggests that the problem may aroused from the fact that the control packet could not find free multiplexers to reach the output fibers

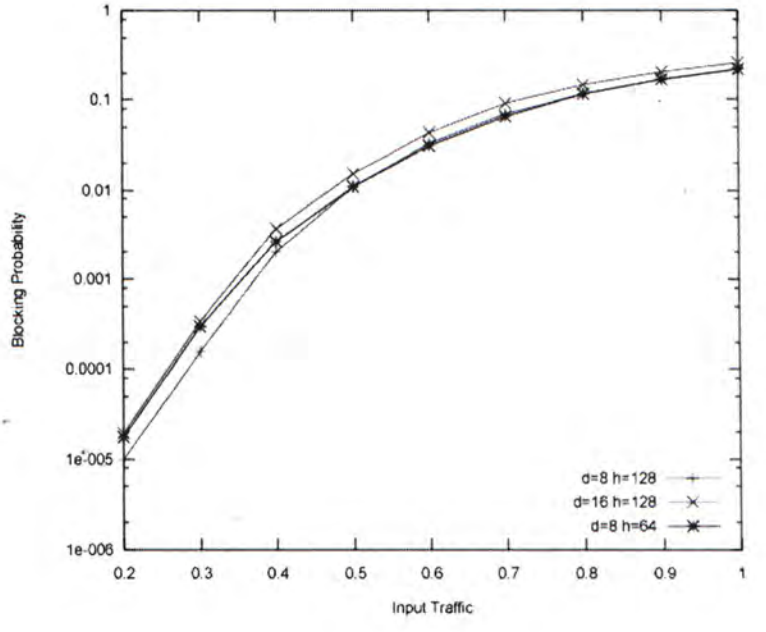


Figure 4-6 (scheme 1) Loss probability versus input traffic load for various combinations of *d* and *h*.

Figure 4-7 shows the loss probability versus loads for $d = 8$, 16 and 32. The number of wavelengths tested here is 64 and the input load is 0.5 as usual. It is showed that although the number of fibers is doubled, which implies the number of input ports is also doubled; the extra number of stages needed is small. When we increase the number of fibers from 8 to 32, the extra number of stages needed is at most 8 as observed from the figure.

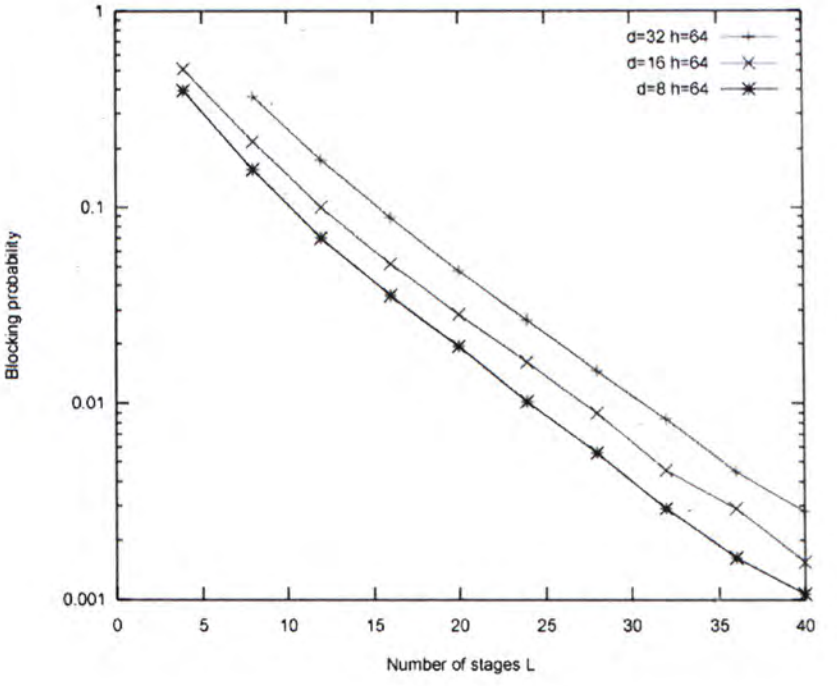


Figure 4-7 (scheme 1) Loss probability versus L for various combinations of *d* and *h*.

4.3 Second Scheme: Tunable wavelength converter.

In the previous sub-section, we learnt that the performance of the proposed DSN switch may be heavily affected by output multiplexing. Theoretically, the output multiplexing should be implemented in a better way. In the first scheme, when a control packet has been fully routed, it would be converted to a fixed wavelength if the corresponding multiplexer is not already being occupied by another burst come from another internal switching node. If it is the case, it has to be deflected to another "set" of internal nodes which share the same specific output wavelength channel. This limitation is not essential; a fully routed burst should have h wavelength channels to choose from. However, consider we would be having $h(L\text{-}D+1)$ internal output

ports connected to one specific output fiber, a $h(L\text{-}D+1) \times h$ switch is necessary here for each fiber. On the other hand, we might consider using tunable wavelength converter to implement output multiplexing, as shown in Figure 4-8. Every outgoing link from the internal nodes is followed by a tunable wavelength converter which its output is connected to the external output fiber. This requires some sort of output controlling unit to keep track of which wavelength channels are being occupied and which are free on the other hand. When a control packet has reached its final destination, the controller would tell the corresponding tunable wavelength converter which wavelength it should change the burst to. The drawback of this design is that tunable wavelength converter is still immature and very expensive.

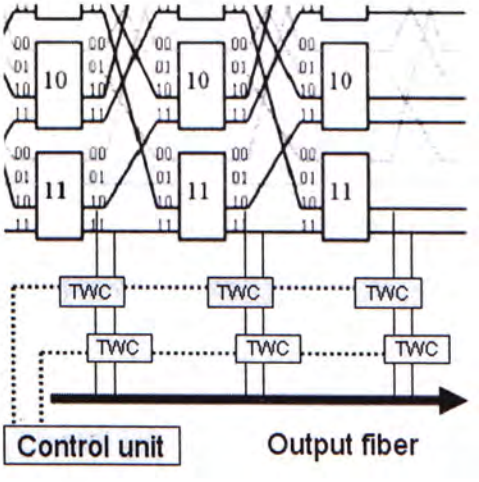


Figure 4-8 Second scheme - hardware implementation

It can be showed that the DSN switch architecture is very attractive in building optical burst switches when tunable wavelength converter can be employed. Figure 4-9 shows the performance difference between the two multiplexing schemes. With input loading of 0.5, $d = 8$ and $h = 128$, the second scheme has comparable performance with the packet-based DSN switch. We also notice that the difference between the banyan switching node and the non-blocking switching node is again very small for the second scheme. Simulation results with various numbers of fibers are given in Figure 4-10. Number of wavelengths tested here is 64 and numbers of fibers are 8, 16 and 32. It shows that the performance difference between the banyan design and the non-blocking design grows with the increase of d.

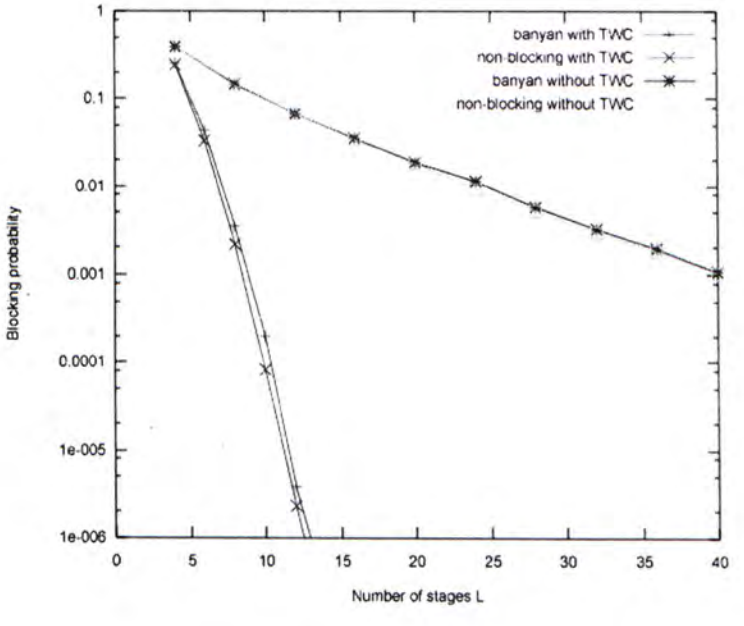


Figure 4-9 Performance difference between first and second schemes

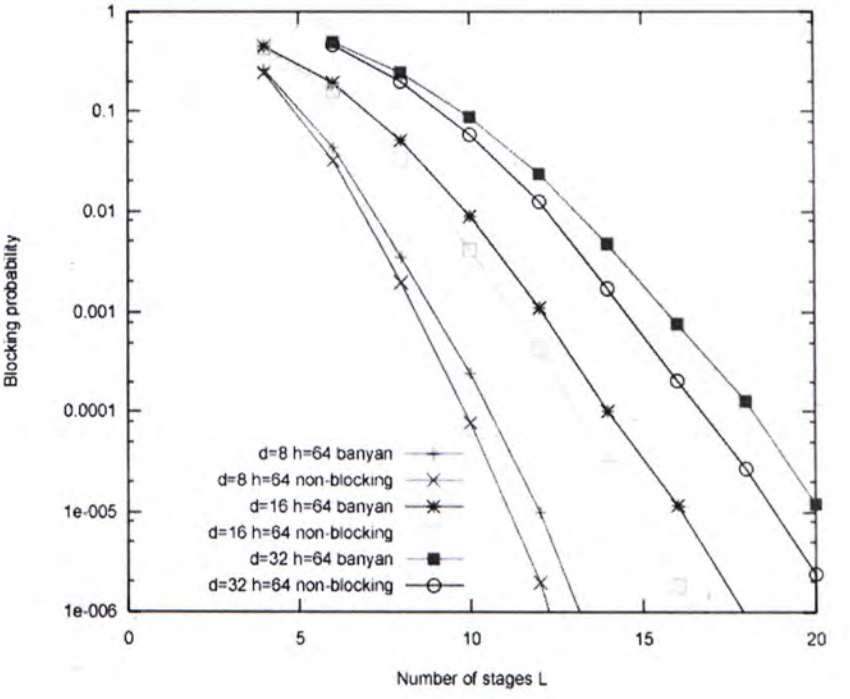


Figure 4-10 Blocking versus L with various d and h for second scheme.

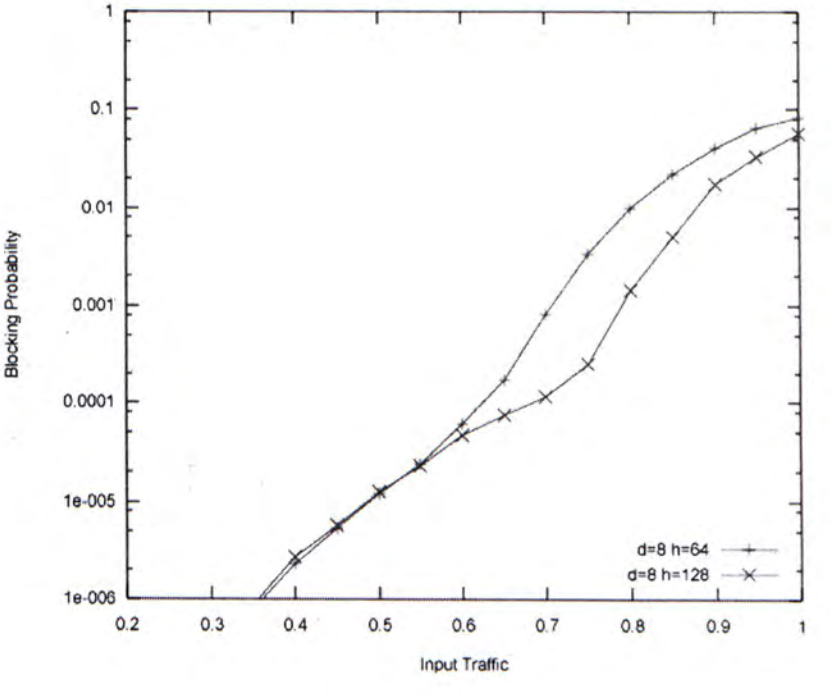


Figure 4-11 Performance of second design d = 8, L=12.

Next, we test our second design to see how the blocking performance varies with different numbers of wavelength channels available inside the fibers. We plot the blocking performances of the switch against various loads with $h = 64$ and $h = 128$ both on Figure 4-11 and Figure 4-12. Figure 4-11 shows simulation results on the switch with $d = 8$ and $L = 12$ while figure 4-12 is based on architecture with 16 fibers and 16 stages. Both figures show that the 128-wavelengths configuration performs better than the 64-wavelengths configuration with input loading starting from 0.6. These results are what we have been expecting as it shows the beauty of the Erlang B formula and WDM technology; with higher channel counts, the blocking performance would become lower.

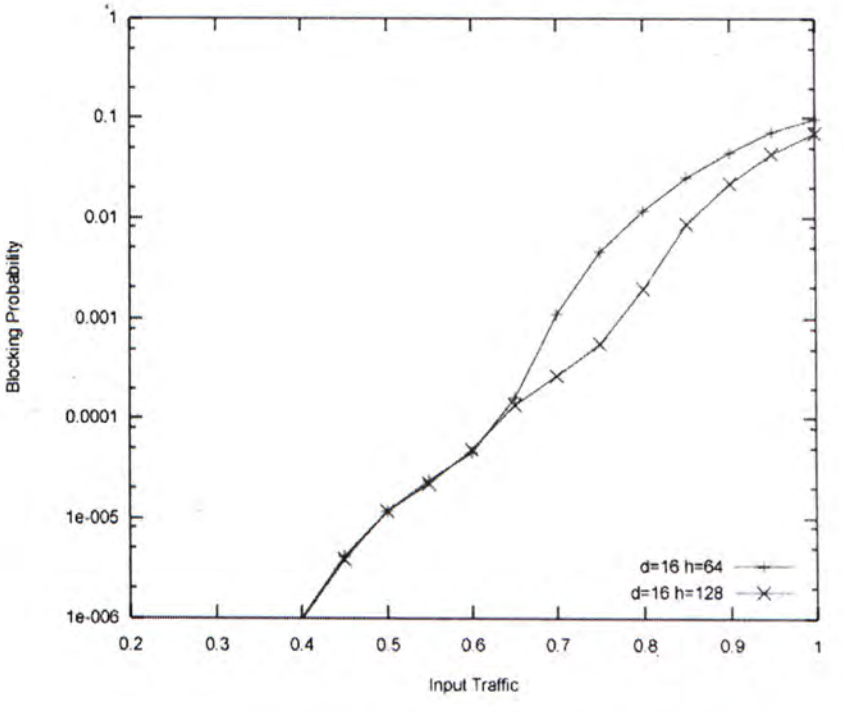


Figure 4-12 Performance of second design $d = 16$, $L = 16$.

4.4 Third Scheme: Route to specific wavelength port.

We have showed that our second scheme would be very promising when tunable wavelength converters become mature in some later time. In the meantime, there is still some room for improvement. As we have proved that the problem of our first scheme lies on the output multiplexing; where control packets have enough instructions to reach its destined output fiber easily, but still have to find a free wavelength channels blindly. In this sub-section, we modify our design by giving the control packets not only the instructions required to reach its output fiber, but also the instructions needed to reach a particular free wavelength channel. In the first scheme, the minimum stage a burst needs to travel is $\log_2 d$ where d is the number of fibers. Now as the burst also needs routing instructions to reach a particular wavelength channel, at least $\log_2 d + \log_2 h$ stages are needed. The way how multiplexers is connected is the same as the first scheme, as shown in Figure 4-13, expect feedback signals from multiplexer is not needed and an input controlling unit is needed to record which wavelength channel is being occupied so that it could easily assign a free wavelength channel to a newly arrived control packet.

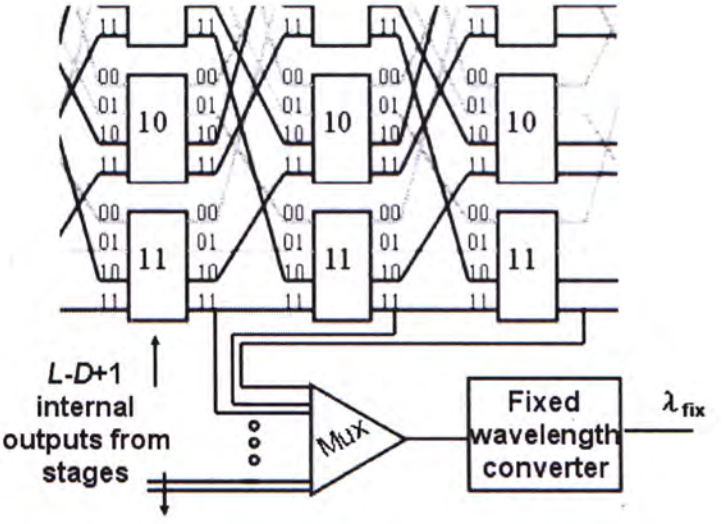


Figure 4-13 scheme 3 implementation

Figure 4-14 shows blocking performance differences between the three schemes. Scheme 2 performs the best but it requires tunable wavelength converter. Scheme 3 is better than the first scheme especially when lower blocking probability is needed. The parameters tested here is $d = 8$, $h = 128$, offered load is 0.5. Figure 4-15 and 4-16 give us information about how much stages are needed while the number of fibers or the number of wavelengths is doubled. Again, the result is encouraging. It seems an increase of 4 stages is enough to tackle a double in the d and h. Notice that in this scheme with $\rho = 0.5$, a system with $h = 128$ performs worse than a system with $h = 64$, as opposed to the inverse observations in Scheme 1 and 2. Figure 4-17 compares Scheme 3 with the two wavelength converting switch designs proposed by J. Turner as described in Chapter 2. We show that with $L = 40$, we can achieve better performance than the one which uses passive wavelength grating router.

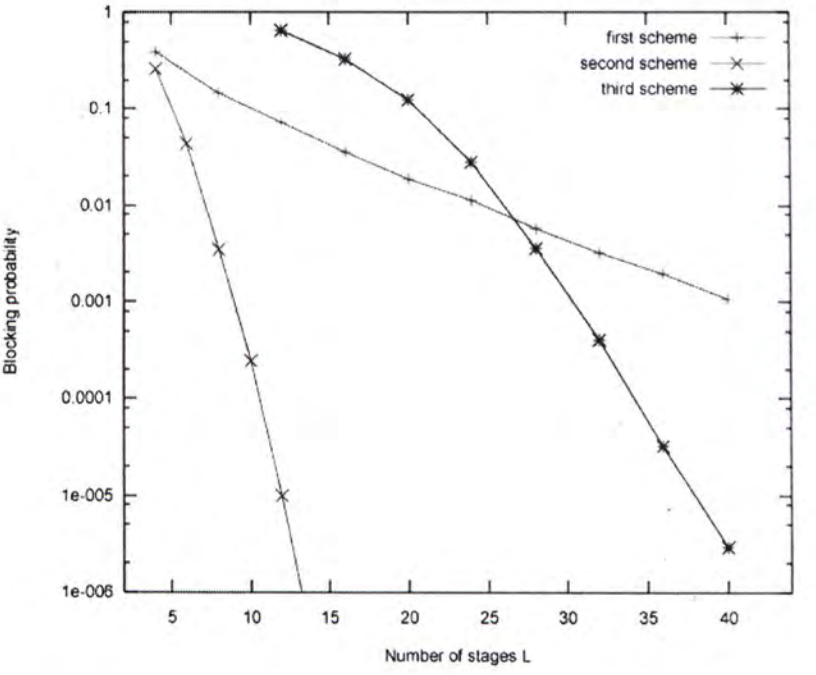


Figure 4-14 Performance differences between the three schemes

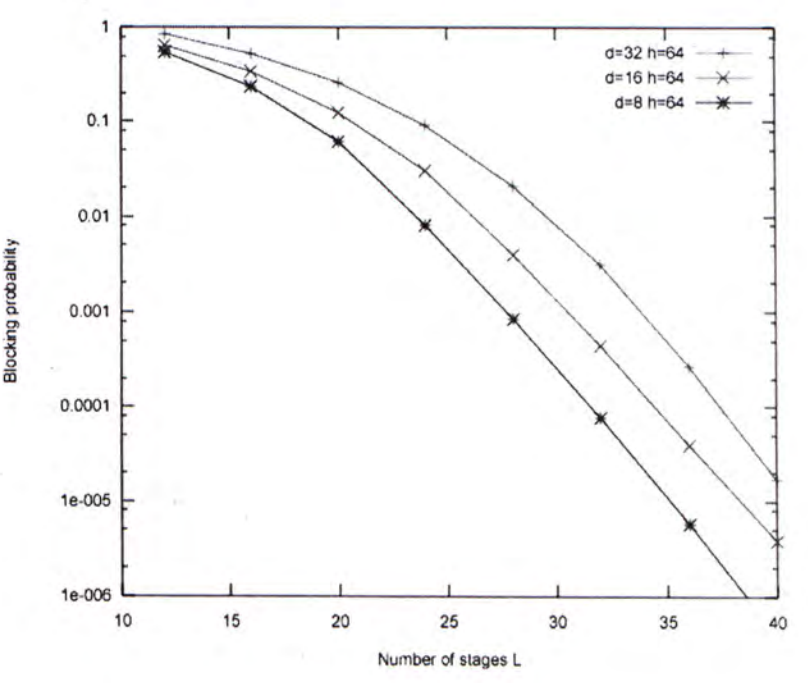


Figure 4-15 Performance of the third scheme with different number of fibers

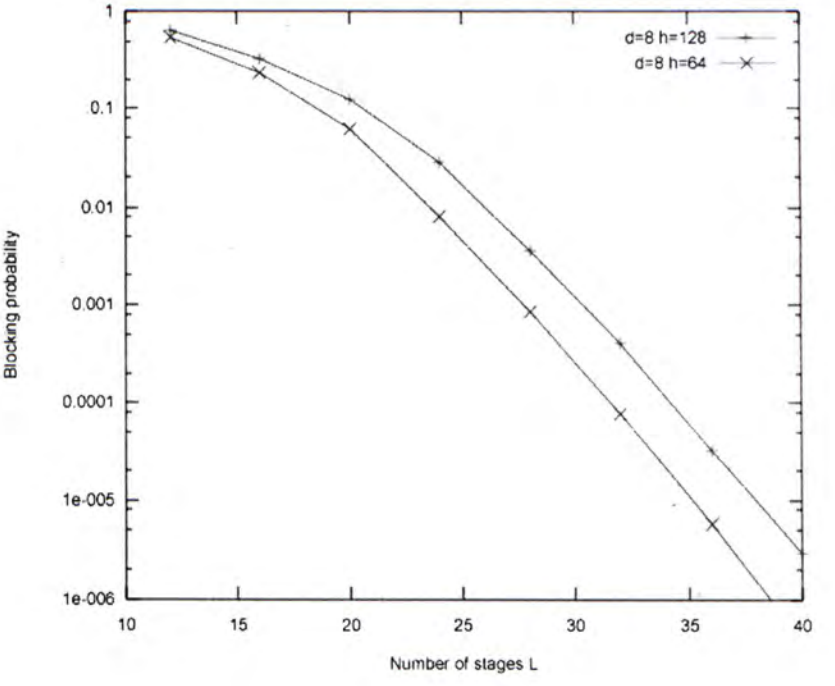


Figure 4-16 Performance of the third scheme with h = 64 and 128

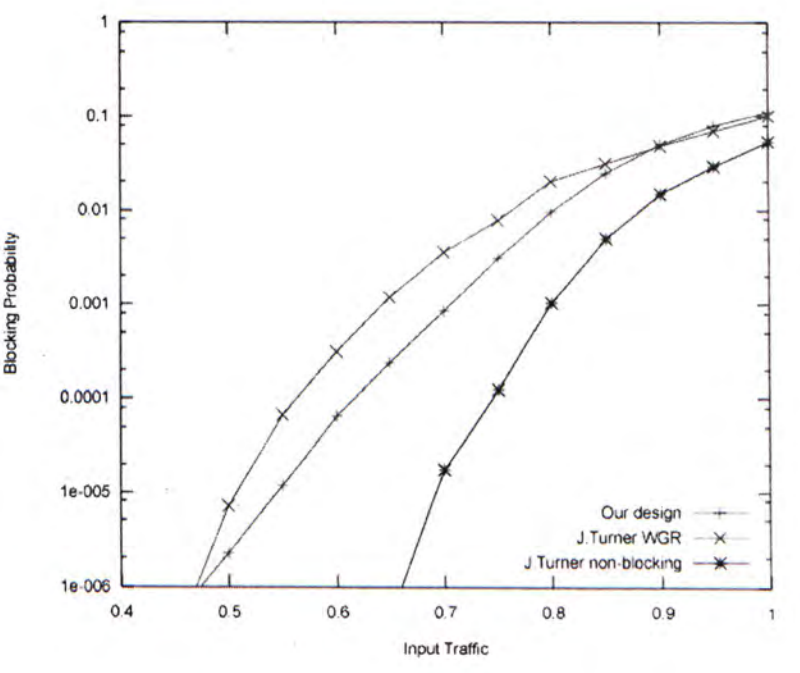


Figure 4-17 Blocking probabilities of system designs described in Chapter 2.2, 2.3 and 4.4

4.5 Analysis on blocking due to insufficient stages

In this section, we calculate blocking due to insufficient number of stages in the switch.

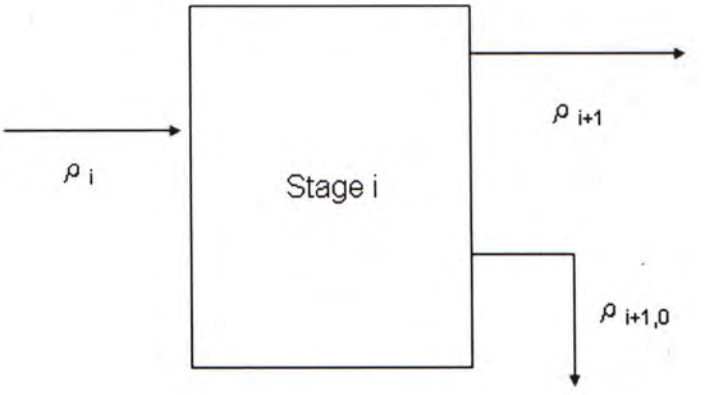


Figure 4-18 Definitions of ρ_i and $\rho_{i,0}$

As shown in Figure 4-18, we define $\rho_{i,j}$ be the amount of input at stage i that is j steps away from its destination. Then we have $\rho_{i,0}$ as the amount of traffic that leave the switch at stage i-1. We then also define ρ_i as the amount of traffic going to stage i. Now we have

$$\rho_i = \sum_{j=1}^{n} \rho_{i,j} \tag{4.1}$$

$$\rho_{i+1} = \rho_i - \rho_{i+1,0} \tag{4.2}$$

As shown in previous section, the undeflected and deflected probabilities in each switching module are

$$p_i = \frac{16 - 8\rho_i}{16 - \rho_i^4} \tag{4.3}$$

$$q_i = 1 - p_i \tag{4.4}$$

By the deflection routing operations of Dual Shuffle-exchange Network, we have the following equations

$$\rho_{i+1,0} = p_i \rho_{i,1} \tag{4.5}$$

$$\rho_{i+1,1} = p_i \rho_{i,2} \tag{4.6}$$

$$\rho_{i+1,j} = p_i \rho_{i,j+1} + q_i \rho_{i,j-1} \tag{4.7}$$

$$\rho_{i+1,n} = q_i \rho_{i,n} + q_i \rho_{i,n-1} \tag{4.8}$$

By recursively applying these equations, we can obtain all the $\rho_{i,0}$. Then we can have the blocking probability due to insufficient number of stages

$$P_{stage} = \frac{\rho_{L+1}}{\rho_0} \tag{4.9}$$

while ρ_0 is equal to the applied load to the switch. Figure 4-19 shows the result between analysis and simulation for d=8 and h=128. The difference might be due to the inaccurate approximation of the idle period. As bursts keep leaving the switch at each stage, this results in the increase of length of idle periods. In this way, the idle periods are no longer exponentially

distributed. Actually, they have a hypoexponential distribution which is resulted from adding exponential distributions. This inaccuracy might lead to incorrect calculation of deflection probability and therefore the overall blocking probability.

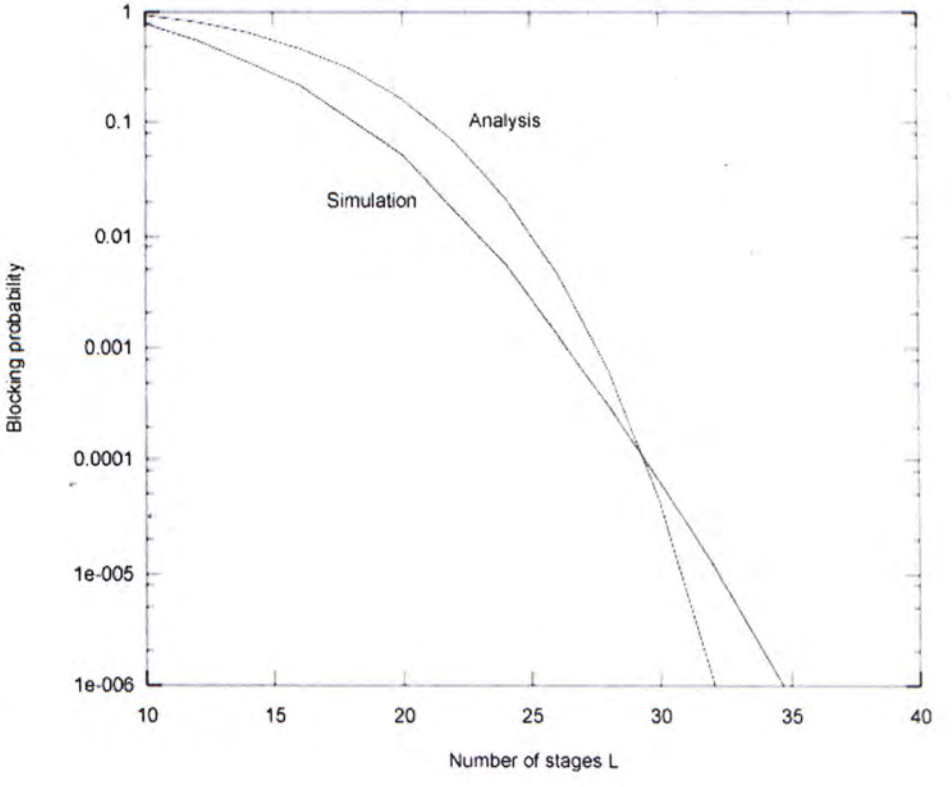


Figure 4-19 Analysis and Simulation results on the third scheme.

Chapter 5

Vertical expansion and 8 × 8 MEMS switches

By analyzing the delay of the switch on various input traffic loads, we find out that the performance of the switch would remarkably drop when the input traffic is high. On the other hand, while the number of ports a DSN switch should have is a power of 2, it is not necessary for the values of d and h to be also powers of 2. Therefore, it is possible for us to decide how many ports from the DSN switch can be left unconnected to the optical fibers. In this way, in the view of the incoming traffic to the switch, the DSN switch is vertically expanded. It can be shown that for high input traffic load, vertical expansion would give better performance than horizontal expansion (by increasing the number of stages) for the same complexity. DSN with 8 × 8 MEMS switching nodes are also discussed here

5.1 Delay analysis of DSN

In our burst traffic model for internal planes, we assume burst length is exponential distribution with mean $1/\mu$, idle time length is exponential distribution with mean $1/\alpha$.

As presented in Section 3.6, the deflection probability of an internally non-blocking switching node is

$$q_n = 1 - \frac{\mu(\alpha+\mu)^3}{(\alpha+\mu)^4 - \alpha^4} = \frac{8\rho - \rho^4}{16 - \rho^4} \tag{5.1}$$

Probability for successful routing

$$p = 1 - q = \frac{16 - 8\rho}{16 - \rho^4} \tag{5.2}$$

Let T_i denote the expected additional number of stages would have taken by a burst in state i before reaching its destination. Then T_n is the expected number of stages to a burst has to visit before leaving the switch, where n is the length of a routing tag. It can be $\log_2 d$ or $\log_2 dh$. As illustrated in Figure 3-2, we have [13]

$$T_n = \frac{n}{p-q} - \left(\frac{1}{p-q} - \frac{1}{p}\right)\left[\frac{1-(q/p)^n}{1-q/p}\right] \tag{5.3}$$

With a large n, the latter part becomes less dominant and can be neglected. Then we have [13]

$$T_n \approx \frac{n}{p-q} = \frac{n(16-\rho^4)}{16-16\rho+\rho^4} \tag{5.4}$$

Figure 5-2 has plotted the equation (5.4), we can see that the delay is large starting from input traffic around 0.8. We might want to reduce the traffic

seen from each node by 20% in order to guarantee small delay. This can be achieved by vertical expansion.

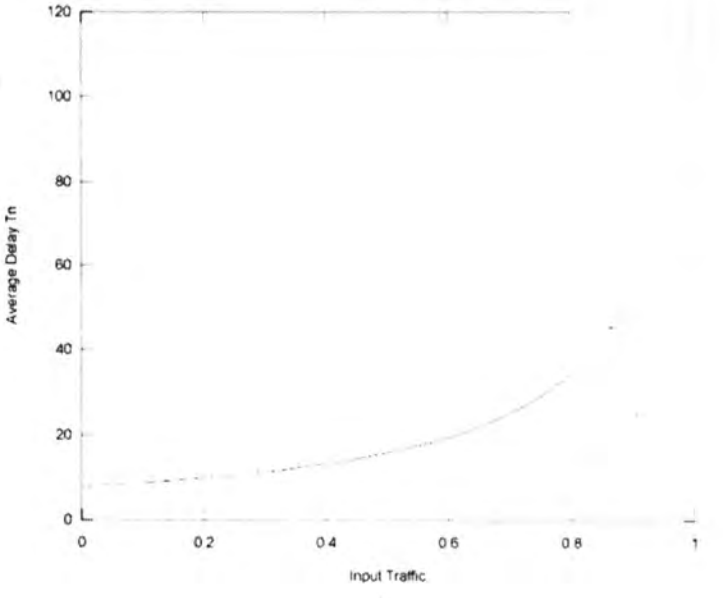


Figure 5-1 The average delay of Dual Shuffle-exchange Network with n = 8

5.2 Vertical Expansion.

Lower blocking probability is achieved normally by increasing the number of stages (horizontal expansion). However, it is also possible to reduce blocking by means of not connecting some of the input/output ports to the optical fibers, as illustrated in Figure 5-3. As the ratio of ports with input traffic to the total number of input ports is decreased, one can see the switch as vertically expanded. These unconnected ports will be distributed as evenly as possible among the input/output ports so that every node in each stage should have almost the same amount of input traffic.

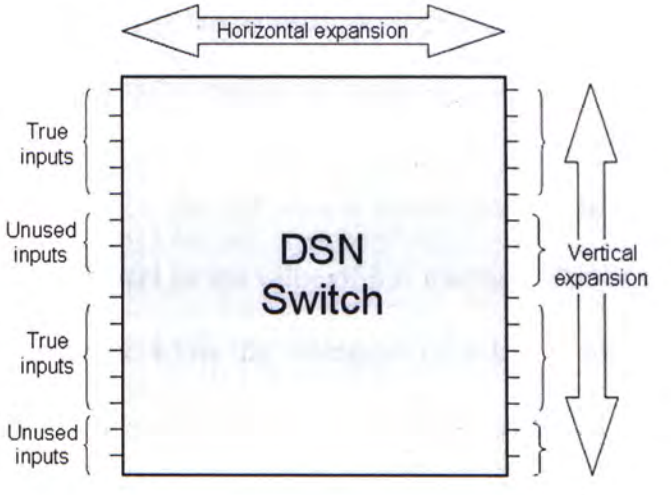


Figure 5-2 The architecture of DSN switch with vertical expansion

5.3 Simulation results on vertical expansion

In this section, we show that vertical expansion can give better result than horizontal expansion in some case. The red line shown in Figure 5-3 is with configuration of $d = 8$, $h = 128$ and $L = 24$. While keeping the same number of nodes in each stage, the switch is vertically expanded by 1.143 when h is reduced to 112. L is decreased to 21 so as to compensate this vertical expansion. (Since the number of nodes in each stage is unchanged, h and L should be changed in proportional). A vertical expansion by 1.33 would make $h = 96$ and $L = 18$. As it is suggested by the analysis from the previous sub-section, our DSN switch would have unsatisfactory performance when working at traffic rate from 0.8 to 1. In order to avoid this traffic rate region, we can have a vertical expansion of 1.25. Therefore expansion values 1.143 and 1.33 are tested here and should give better

performance when traffic rate is high, as shown in Figure 5-3. Simulation is based on the third output multiplexing scheme.

One may argue that the way we proved vertical expansion can give better performance is incorrect as the value of h is changed. But as suggested by the Erlang B formula, while the customers ($d \times h$) to servers (h) ratio is kept constant, the blocking probability would becomes lower when we have more customers or servers. Therefore, as we do vertical expansion by reducing h, the blocking probability should be higher as opposed to what Figure 5-3 shows.

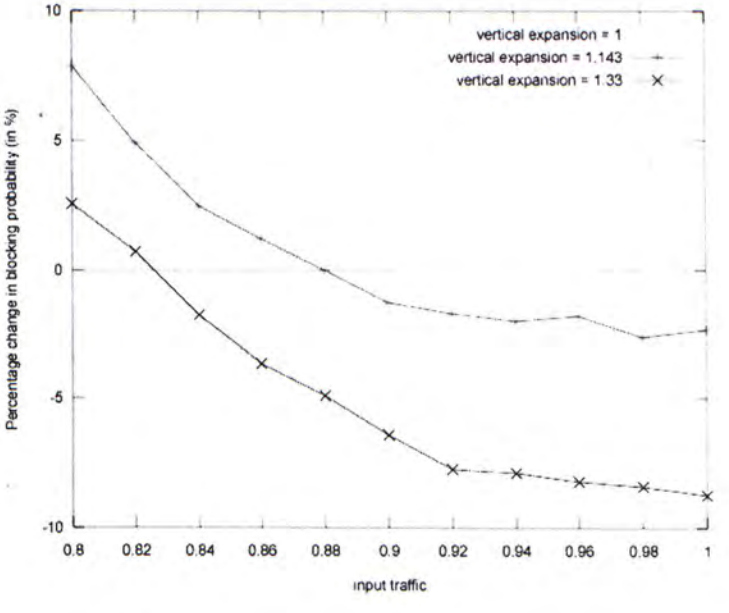


Figure 5-3 Percentage change in blocking probability for vertical expansion = 1.143 and 1.33

5.4 Building DSN with 8 × 8 MEMS switches.

Microelectromechanical systems (MEMS) [15-20] switches are regarded as the most promising technology to achieve functionality. Based on their structure and operation, MEMS switches can be divided [19] into two-dimensional (2D) MEMS switches and three-dimensional (3D) MEMS switches. In a typical 2D switch, the mirrors simply flap up and down in the optical equivalent of a cross-bar switch. When they're down, light beams pass straight over them. When they're up, they deflect the beam to a different output port. The biggest single block of 2D MEMS switches reported so far is a 32 × 32 switch already [20]. As Dual Shuffle-exchange Network uses 4 × 4 switching nodes, we have to make some modification in order to fully utilize the advantages of the bigger-sized MEMS switches.

For simplicity reasons, we first focus on 8 × 8 MEMS switches. There are at least two ways to build a self-routing, one-step error correcting network based on the concept of Dual Shuffle-exchange Network. One of them is to expand the bandwidth of the existing links. In Figure 5-4, a DSN with every link doubled is shown. As every stage has one more path to the next stage, the deflection probability can be made lower in this way. The other way to build the network is to change the structure of the underlying shuffle network. For a typical shuffle network, the switching nodes are divided into two halves vertically. Each node is connected to its previous stage with one link from the upper half and one link from the lower half. Now as we are using 8 × 8 nodes to build a 'Dual' network, the two

underlying networks (one and its mirror image) should be built by 4 × 4 nodes. Therefore, the shuffle network is now divided into four equal parts vertically. Each node is connected to its previous stage with one link from each quarter, as shown in Figure 5-5. In this way, the network is again self-routed and each stage could process two bits from the routing tag.

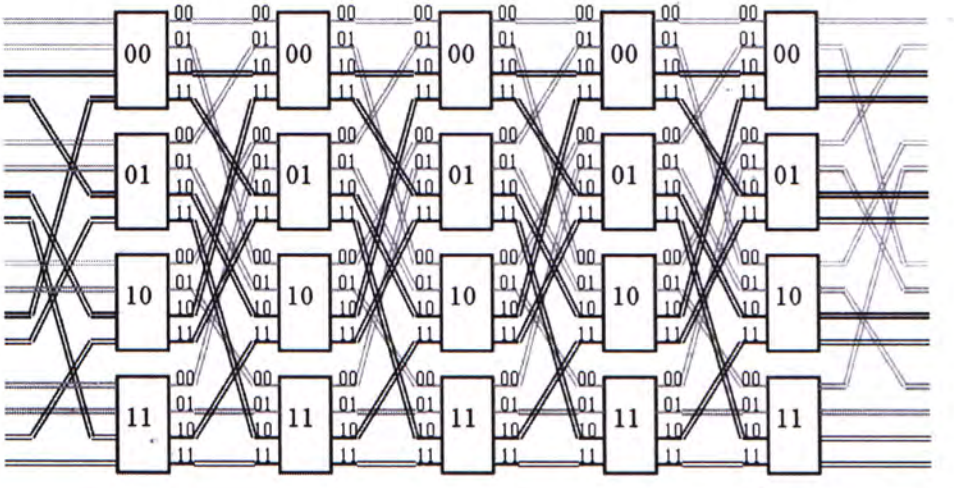


Figure 5-4 Dual Shuffle-exchange
Network with each links doubled

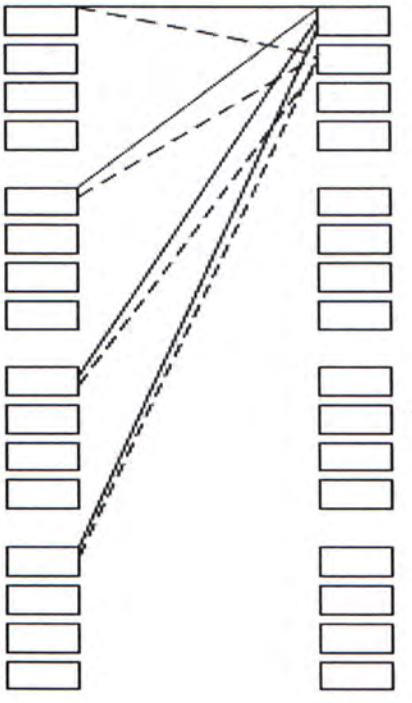


Figure 5-5 Quarter shuffle. Each node is connected to its previous stage with one link from each quarter

5.5 Prove of the proposed Quarter shuffle network

We label the links of the switch in a binary fashion. Now the top link is 00…00 and the bottom is 11…11. Then the modules in the quarter shuffle network is connected in the way that outgoing link $x_n x_{n-1} \ldots x_2 x_1$ of a stage is connected to incoming link $x_{n-2} x_{n-3} \ldots x_n x_{n-1}$ of the next stage as shown in Figure 5-6. That is two cyclic left shifts of the link label.

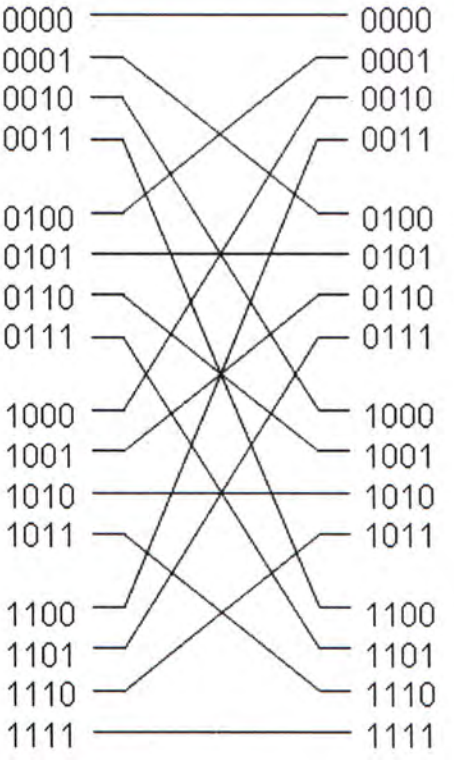


Figure 5-6 Quarter Shuffle with link labels.

Now, there are $2^{(n-2)}$ 4 x 4 switch elements in any stage. Therefore we can label it using (n-2) bits. The four incoming and the four outgoing links connected to the switch element $x_{n-2}\ldots x_1$ are labeled $x_{n-2}\ldots x_1 00$ if the two routing bits are 00, to link $x_{n-2}\ldots x_1 01$ if the two routing bits are 01 and so on.

Let the source and the destination addresses of a packet be $S = s_n\ldots s_1$ and $D = d_n\ldots d_1$, respectively. The destination address will be used for routing starting from the most significant bit to the least significant bit. Initially, the packet occupies link $s_n\ldots s_1$, at the entrance to the shuffle-exchange network. After the first shuffle, its link label is $s_{n-2}s_{n-3}\ldots s_1 s_n s_{n-1}$ at the input

to the first-stage switching node. Bit $d_n d_{n-1}$ is used to switch this packet outgoing link $s_{n-2}s_{n-3}\ldots s_1 d_n d_{n-1}$. We see that $s_n s_{n-1}$ has been replaced by $d_n d_{n-1}$. By another shuffle and exchange, the packet would occupy link $s_{n-4}s_{n-5}\ldots s_1 d_n d_{n-1} d_{n-2} d_{n-3}$. Repeating the process, we see that the output links of the successive switching stages traversed by a packet is

$$S = s_n \ldots s_1 \rightarrow s_{n-2}s_{n-3}\ldots s_1 d_n d_{n-1} \rightarrow \ldots$$

$$\rightarrow s_{n-i}s_{n-i-1}\ldots s_1 d_n d_{n-1}\ldots d_{n-i+2}d_{n-i+1} \rightarrow$$

$$\rightarrow d_n \ldots d_1 = D$$

5.6 Comparison between Quarter shuffle and doubled links approaches

In this section, we are going to compare these two new designs analytically. For both designs, the transition diagrams are the same as shown in Figure 3-8 with number of states changed to 8. The limiting probabilities for each state are

$$\pi_i = \frac{\binom{8}{i}\mu^{8-i}\alpha^i}{(\alpha+\mu)^8 - \alpha^8} \qquad (5.5)$$

The differences between these two designs are the deflection probability in each node and the number of routing bits that would be processed in one stage.

For the second design, as each output link is independent, the deflection probability is

$$\begin{aligned} d &= 1-\left(\sum_{i=0}^{7}\frac{(8-i)}{8}\pi_i\right) \\ &= 1-\frac{\mu(\alpha+\mu)^7}{(\alpha+\mu)^8-\alpha^8} \\ &= \frac{32\rho-\rho^8}{64-\rho^8} \end{aligned} \tag{5.6}$$

As each stage can process 2 bits, n is reduced by half. The delay of the switch is

$$T_n = \frac{n(64-\rho^8)}{2(64-64\rho+\rho^8)} \tag{5.7}$$

For the first design with all the links doubled, as outputs are not independent, it is harder to arrive on the deflection probability. Define D as the output port number that the new arriving burst destined. Define $P(i)$ as the probability that for an arriving burst can reach D when there are i bursts in the switching node. Trivially, as there are two output ports for each port number

$$P(0)=P(1)=1 \tag{5.8}$$

$$P(2)=1-\left(\frac{1}{4}\right)^2=\frac{15}{16} \tag{5.9}$$

It happens when all the two previous bursts are destined to D. For i=3, burst is deflected when all the three previous bursts have the same destination D or any two of them destined to D

$$P(3) = 1 - \frac{1 + {}_3C_2 3}{4^3} = \frac{54}{64} \tag{5.10}$$

Things start tricky when i is equal to 4, as other bursts would have been deflected to D also. Our strategy is to choose the port number with less busy ports for deflection.

$$P(4) = 1 - \frac{1 + {}_4C_3 \times 3 + {}_4C_2 \times 3^2 + 1}{4^4} = \frac{188}{256} \tag{5.11}$$

The value 1 in the end of (5.11) is happened when the first three bursts destined to the same address other than D ($\left(\frac{1}{4}\right)^3$) and the last one of them is deflected to D ($\frac{1}{3}$). Then the fourth one also wants to reach D ($\frac{1}{4}$). The first three bursts have 3 values to choose from. Therefore, this probability equals

$$3 \times \left(\frac{1}{4}\right)^3 \times \frac{1}{3} \times \frac{1}{4} = \frac{1}{4^4} \tag{5.12}$$

In the same way, $P(5)$ and $P(6)$ can be obtained

$$P(5)=\frac{632}{1024}, P(6)=\frac{1026}{2048} \tag{5.13}$$

For $i = 7$, we have only one output port available, therefore

$$P(7)=\frac{1}{4} \tag{5.14}$$

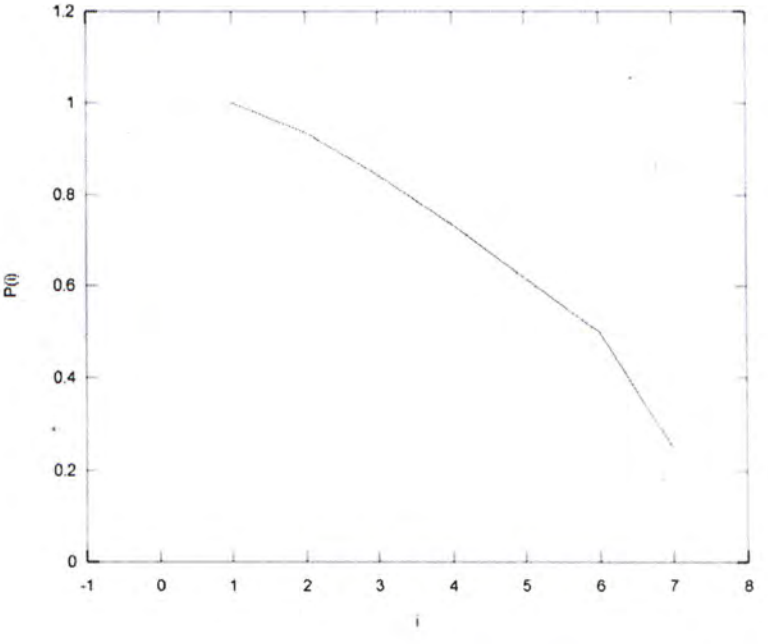


Figure 5-7 Probability that for an arriving burst can reach its destined output when there are *i* bursts in the switching node.

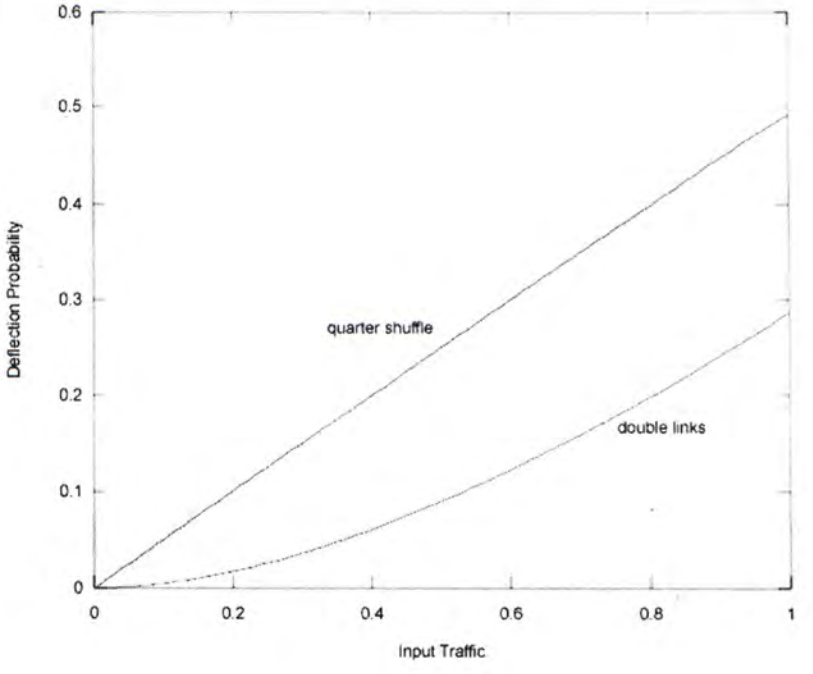


Figure 5-8 Deflection probability difference between quarter shuffle and doubled links

Figure 5-7 shows $P(i)$ against i. The deflection probability can be calculated by multiplying $P(i)$ with π_i. Deflection probabilities of the two designs are shown in Figure 5-8. The average delay can then be obtained and is shown in Figure 5-9. We can see that both designs of DSN with 8 × 8 nodes perform better than the DSN with 4 × 4 nodes. The quarter shuffle is better when load is smaller as this implies that there is not so much deflection and therefore it simply visits its minimum stages it needed, which is smaller. On the other hand, when load is high, that means a lot of deflection would occur. The doubled links design has smaller deflection probability; therefore the delay on this design is smaller. Simulation result supporting the analysis is shown in Figure 5-10.

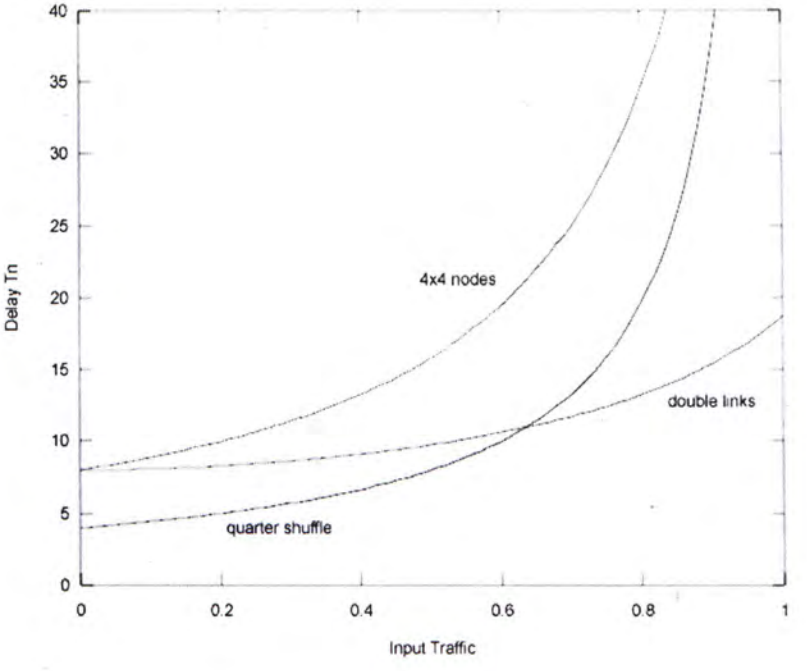


Figure 5-9 Comparing the delays of different designs using 8 × 8 nodes with DSN with 4 × 4 nodes

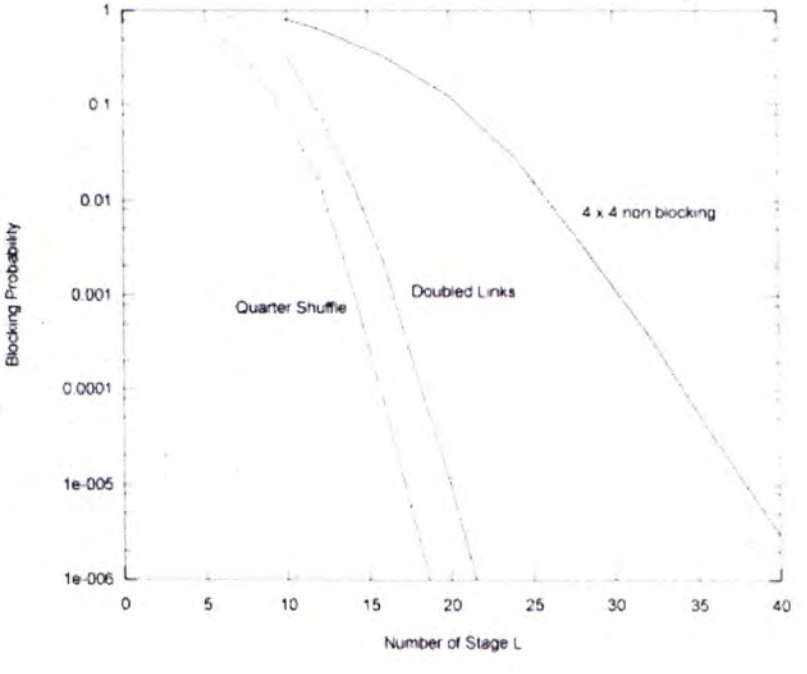


Figure 5-10 Simulation on the delays of different designs using 8 × 8 nodes with DSN with 4 × 4 nodes.

Chapter 6

Conclusion

In this thesis, we propose a novel approach to implement Optical Burst Switching (OBS) using the Dual Shuffle-exchange Network (DSN) as the core switching fabric. DSN possesses the self-routing property which allows major simplifications on the complex crossbars setup mechanisms. In addition, its asynchronous and buffer-less natures are highly preferable in the optical environment. We also show that with an appropriate error-correcting routing algorithm, the output wavelength contentions can be reduced by means of internal deflection routing.

We have studied the performance of three DSN switch schemes for use in OBS routers. These schemes differ mainly from how they handle output multiplexing. The first one requires the less central controlling unit. We assign each incoming burst with a routing tag, instructing it to reach one (but not a specific one) of wavelength channels of its destined output fiber. However, this wavelength channel might have already been occupied. In this way, the burst would be deflected to other wavelength channel until it can find a free wavelength channel or it is dropped at the end of the switch.

The second scheme introduces the use of Tunable Wavelength Converter to handle output multiplexing. When a burst has reached an output port of its destined output fiber, it immediately converts to one of the free wavelength available in the destined output fiber. This scheme gives a much better result than the first scheme but Tunable Wavelength Converters are still immature and expensive. In the third scheme, rather than instructing each arriving burst to reach its output fiber, we instruct each burst to reach a particular wavelength channel of its destined output fiber. In this way, the bursts do not need to be deflected in order to find a free wavelength channel. However, a central controller is needed to record which wavelength channel is being occupied so that it could easily find a free wavelength channel for a newly arrived data burst.

We also propose vertical expansion architecture based on DSN. We find that with vertical expansion of 1.33, we can further reduce the blocking probability especially for high input traffic rates. As 8 × 8 MEMS switches are available, we have designed two schemes to make use of these switches and have improved the performance of switch.

Bibliography

[1] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) – A New Paradigm for an Optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69-84, Jan. 1999.

[2] M. Yoo, C. Qiao, S. Dixit, "Qos Performance of Optical Burst Swtiching in IP-Over-WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 2062-2071, Oct. 2000.

[3] J. Ramamirtham and J. Turner, "Design of wavelength converting switches for optical burst switching," In *proceedings of INFOCOMM*, volume 1, pages 362-370, 2002.

[4] V.M. Vokkarane, J.P.Jue, and S.Sitaraman, "Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks," *Proceedings, IEEE, ICC 2002.*

[5] E. Haselton, "A PCM frame switching concept leading to burst switching network architecture," *IEEE Communications Magazine, vol. 21, pp. 13-19, June 1983.*

[6] *S. Amstutz, "Burst switching - an introduction,"* IEEE Communications Magazine, vol. 21, pp. 36-42, Nov. 1983.

[7] M. Yoo and C. Qiao, "A Novel Switching Paradigm for Buffer-Less WDM Networks," *IEEE Communications Magazine,* 1999.

[8] L. Xu, H.G. Perros, and G Rouskas, "Techniques for optical packet switching and optical burst switching" *IEEE Communications Magazine,* pp.136-142, Jan. 2001.

[9] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration," *IEEE Communications Magazine*, vol. 38, no. 9, pp. 104-114, Sept. 2000.

[10] M. Yoo and C. Qiao, "Supporting Multiple Classes of Service in IP over WDM Networks," *Proceedings, IEEE Globecom '99*, pp. 1023-1027, Dec. 1999.

[11] I. Widjaja, "Performance Analysis of Burst Admission Control Protocols," *IEEE Proc. Commun.,* vol. 142, Feb. 1995, pp. 7-14.

[12] S. L. Danielsen et al., "Analysis of a WDM Packet Switch with Improved Performance Under Bursty Traffic Conditions Due to Tunable Wavelength Converters," *J. Lightwave Tech.,* vol. 16, no. 5, May 1998, pp. 729-35.

[13] Liew, S.C.; Lee, T.T., "NlogN Dual Shuffle-Exchange Network with Error-correcting Routing," *IEEE Communications.*, vol. 42, pp. 754-766, Feb-Apr 1994.

[14] J.Y. Wei and R.I. McFarland, "Just-in-time Signaling for WDM Optical Burst Switching Networks," *J.Lightwave Tech.*, vol.18, no. 12, Dec. 2000, pp. 2019-37.

[15] G. Shen, Tee Hiang Cheng, Sanjay K. Bose, Chao Lu, Teck Toong Chai, "Architectural Design for Multistage 2-D MEMS Optical Swtiches," *J.Lightwave Tech.*, vol.20, no. 2, Feb. 2002, pp. 178-187.

[16] David J. Bishop, C. Randy Giles, Gary P. Austin, Lucent Technologies, "The Lucent LambdaRouter: MEMS Technology of the Future Here Today," *IEEE Communication Magazine*, Mar 2002, pp. 75-79

[17] Patrick B.Chu, Shi-Sheng Lee, Sangtae Park, Tellium, Inc, "MEMS: The Path to Large Optical Crossconnects," *IEEE Communication Magazine*, Mar 2002, pp. 80-87.

[18] Peter De Dobbelaera, Ken Falta, Li Fan, Steffen Gloeckner, Sussant Patra, OMM Inc, "Digital MEMS for Optical Switching," *IEEE Communication Magazine*, Mar 2002, pp. 88-95.

[19] S. Hardy, "All-optical-switching groundswell builds," *Lightwave*, pp. 45-47, May 2000.

[20] Light Reading. (2000) Optical Switching Fabric. [Online]. Available: http://www.lightreading.com/document.asp?doc_id=2254