

PERFORMANCE STUDY OF
MULTIRATE CIRCUIT SWITCHING
IN QUANTIZED CLOS NETWORK



BY

VINCENT WING-SHING TSE

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF INFORMATION ENGINEERING

THE CHINESE UNIVERSITY OF HONG KONG

DECEMBER 1997



Acknowledgement

May all glory and praises dedicate to my Almighty God. Without the intimate relationship with my Lord, Jesus, definitely I will not have the courage strength and intelligence to accomplish my research. I also sincerely express my gratitude to my supervisor Professor Tony T. Lee for his continuous support, guidance, inspiration and invaluable suggestions during my studies and research at CUHK.

Special thanks and appreciation I give to my helpful and affectionate friend—Dr. Philip Pak-tung To for revising my grammatical errors and furnishing my thesis with better organization. I would like to express my gratitude to my dear co-worker, Mr. Thomas Wai-hung Kwok for working unanimously for the studies of "Multirate Circuit Switching in Quantized Clos Network". Surely the insights and clear structure of this studies help me develop my research in better way. I am also very grateful to Dr. Check-hung Lam, Mr. Soung-yue Liew, Mr. Oo Tang for providing insightful opinion form continuously discussing with me and to Mr. Hanford Hang-fung Chan for his technical help in writing the simulation programming. I also wish to thank those who helped in many different ways during my studies, especially Dr. Quan-long Ding, Dr. Xiao-lei Guo, Ms. Cathy Wai-chun Chan, Mr. Walter Chi-woon Fung, Mr. Mai Jin, Mr.

Ngai Li, Mr. Raymond Hon-man Lin, Mr. Patrick Chi-kong Wu and Mr. Alan Yeung for their various kinds of help.

This thesis is dedicated to my respectable parents Mr. Hon-chai Tse and Mrs Mei-yee Tse, to partially reward their abiding love and concern, powerful encouragement and support during my studies. Moreover, special thanks and appreciation I give to my lovely girlfriend, Christine Ka-Kei Tung, for her continuous support, enthusiastic encouragement and untired prayer that have made my study meaningful. Finally, I would like to extend my gratitude to my church-mates, especially my respectable pastor, Rev. Vincent C.P. Lau, and his cherishing wife, Mrs Donna K.C. Lau, and all my lovely and affectionate fellows in the Rejoice fellowship. I really thank for their unconditional love, untired prayer and encouragement.

Abstract

The performance of multirate circuit switching in quantized Clos network is studied in this thesis. Based on bandwidth quantization and connection splitting, multirate traffic can be managed effectively in the Clos network. Moreover, the number of central modules required to achieve non-blocking is reduced compared to existing results. In the broadband ATM network, multirate traffic are in packet (or cell) format, we proposed to use a time slot assignment scheduling algorithm to schedule cell transmission in the Clos network so that there will not be any cell contention inside the Clos network. In addition, the cell delay is bounded so that the Quality of Services (QoS) of each connection is guaranteed. In addition, we propose two construction schemes for the Clos network with bufferless switch modules and buffered switch modules. We compare the computation time complexity and the network space complexity between the two schemes. The cell delay performance of these two schemes is shown by simulation.

摘要

這一篇論文研究多速率電路交換在量化的洛多斯網絡中。包定免更。交我。
性能有條件或非或演細胞限制因了沒兩者設計
的能塞在形的排何被我們提出及了兩設

Contents

1	Introduction	1
2	Principles of Multirate Circuit Switching in Quantized Clos Network	10
2.1	Formulation of Multirate Circuit Switching	11
2.2	Call Level Routing in Quantized Clos Network	12
2.3	Cell Level Routing in Quantized Clos Network	16
2.3.1	Traffic Behavior in ATM Network	17
2.3.2	Time Division Multiplexing in Multirate Circuit Switching and Cell-level Switching in ATM Network	19
2.3.3	Cell Transmission Scheduling	20
2.3.4	Capacity Allocation and Route Assignment at Cell-level	29
3	Performance Evaluation of Different Implementation Schemes	31
3.1	Global Control and Distributed Switching	32
3.2	Implementation Schemes of Quantized Clos Network	33
3.2.1	Classification of Switch Modules	33
3.2.2	Bufferless Switch Modules Construction Scheme	38

3.2.3	Buffered Switch Modules Construction Scheme	42
3.3	Complexity Comparison	44
3.4	Delay Performance of The Two Implementation Schemes	47
3.4.1	Assumption	47
3.4.2	Simulation Result	50
4	Conclusions	59
	Bibliography	62

List of Figures

1.1	Three different configurations to provide connectivity among four users in a network.	2
1.2	A 4×4 non-blocking switch using 16 cross-bar switching elements	3
1.3	A 3-stage Clos network with k switching modules in first stage and last stage, m switching modules in central stage	4
2.1	Capacity Allocation Matrix	14
2.2	Flow chart of the routing algorithm at connection-level	15
2.3	Example of non-splitting connection setup	16
2.4	Example of connection setup with splitting	16
2.5	Different level time scale in ATM network	18
2.6	Realization of multirate circuit switching in cell-level	20
2.7	Example of a 6×6 multirate quantized Clos network	22
2.8	The CAM of the 11 connections in Figure 2.7	22
2.9	Clos network model for the TSA problem	23
2.10	Correspondence Clos network model of the TSA problem which using M time slots out of $2M - 1$ time slots	24
2.11	The time slot Clos network model of figure 2.7	25

2.12	Graph model $G(X, Y, E)$ corresponding to module S and E of figure 2.11	27
2.13	Flow chart of the routing algorithm at cell-level	30
3.1	Input and Output buffered switch	34
3.2	The modular Knockout switch[8]	35
3.3	Sunshine architecture[9]	36
3.4	Block diagram of a shared buffer memory switch	37
3.5	Bufferless switch modules construction scheme	38
3.6	The routing of a 4×8 expansion network at first stage	39
3.7	The routing of a 8×4 concentration network at last stage	40
3.8	The routing of a 8×8 Benes Network at central stage	41
3.9	Buffered switch modules construction scheme	44
3.10	Example of cell arrival at different time unit	50
3.11	Mean cell delay performance of bufferless construction scheme with network size 256×256 , quantization level = 100	52
3.12	Mean cell delay performance of bufferless construction scheme with network size 64×64 , quantization level = 100	53
3.13	Mean cell delay performance of output-buffered module construction scheme with network size 256×256 , quantization level = 100, group size=6	54
3.14	Mean cell delay performance of output-buffered module construction scheme with network size 64×64 , quantization level = 100, group size=6	55

3.15	Mean cell delay performance of input-buffered module construction scheme with network size 256×256 , quantization level=100, look-ahead-window size=20	56
3.16	Mean cell delay performance of input-buffered module construction scheme with network size 64×64 , quantization level=100, look-ahead-window size=20	57
3.17	Comparison of the mean cell delay performance between three different construction scheme	57
3.18	Maximum cell delay performance of using output-buffered module construction scheme	58

Chapter 1

Introduction

Telecommunication plays an important role in this age of information explosion. The telecommunication industry is evolving quite rapidly due to the rapid advances in the VLSI technology and the optical technology in past few decades[1][20][22]. As a result, the demand of multimedia services such as video, audio and data are growing up rapidly. The challenge that telecommunication engineers face nowadays is how to design an efficient method to support a variety of services with diverse traffic characteristics. The design problems include how to build a *universal* network can support this huge amount of information and can efficiently manage the flow on the network.

The communications between users in traditional telephone network are done mainly with a *circuit switching* network[20]. Switching is very important since it saves the cost of transmission links between different users. In Figure 1.1, we show three different network configurations to connect four users. Obviously, the number of transmission links used in Figure 1.1(c) is the least. Switching provides an economical means to connect the source and the destination nodes

by providing a path through the network.

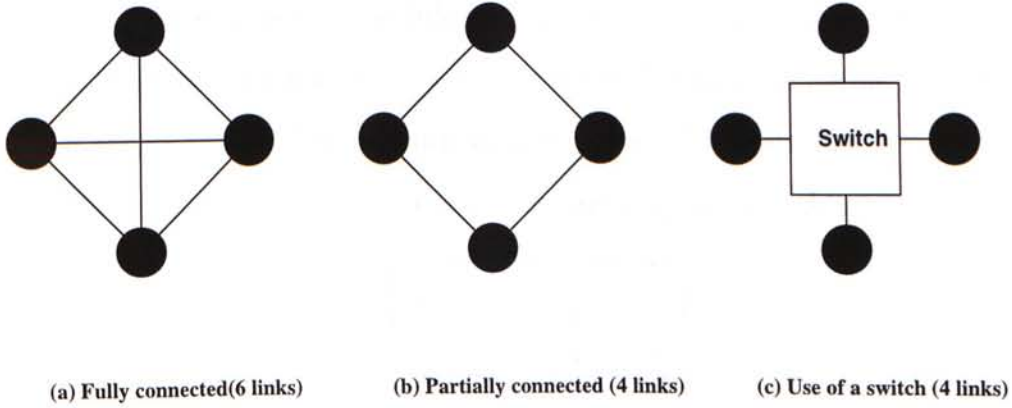


Figure 1.1: Three different configurations to provide connectivity among four users in a network.

The simplest way to implement an $N \times N$ *non-blocking* switch is to use N^2 cross-bar switching elements. A switch is said to be *blocking* if it fails to satisfy a connection requirement even if the two terminals involved are idle[20]. A switch is *strictly non-blocking* (SNB) if a connection can always be set up between any idle input and output without the need to rearrange the paths of the existing connections[15]. A switch is *rearrangeably non-blocking* (RNB) if a connection can always be set up between any free input and output, although it may be necessary to rearrange the existing connections[15]. Figure 1.2 shows an example of a 4×4 non-blocking switch using 16 cross-bar switching elements. We can see that the number of switching elements increases rapidly when the number of ports of the switch increases. In 1953, C. Clos proposed a 3-stage interconnection network which requires fewer number of switching elements and yet it is non-blocking even though the size of the switching network is large[3]. Figure 1.3 shows a 3-stage Clos network $C(m, n, k)$ with k switching modules in the first stage and the last stage, and m switching modules in central stage.

Each first stage switching module is an $n \times m$ non-blocking switching module. Each central stage switching module is an $k \times k$ non-blocking switching module. Each last stage switching module is an $m \times n$ non-blocking switching module. It has been shown that a Clos network with $m \geq 2n - 1$ is a strictly non-blocking network[3]. If $m \geq n$, the network is a rearrangeably non-blocking network[4].

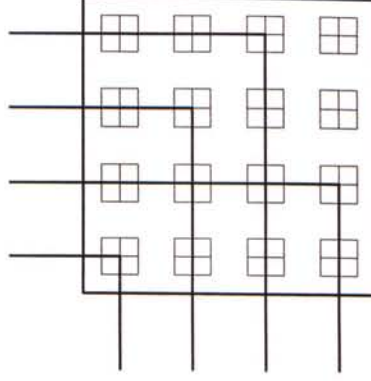


Figure 1.2: A 4×4 non-blocking switch using 16 cross-bar switching elements

The challenging problem of circuit switching in Clos network is how to assign a central module to establish the route for each connection. In 1971, Opferman and Wu proposed a looping algorithm to route the connections inside the Clos network[5]. However, when the network is large, the connection set-up time is long and the algorithm is not suitable for high-speed multimedia broadband network. In 1996, Lee and Liew proposed a parallel algorithm, which can route the connections in the Clos network in parallel if the number of central modules is a power of two[25]. The algorithm has the advantages that the input-output mapping need not be full permutation and output-contention can be resolved during the route assignment. Although parallel processing can reduce the connection set-up time, the bandwidth of each connection is limited to a single basic rate.

In future broadband communications network, a multitude of services with

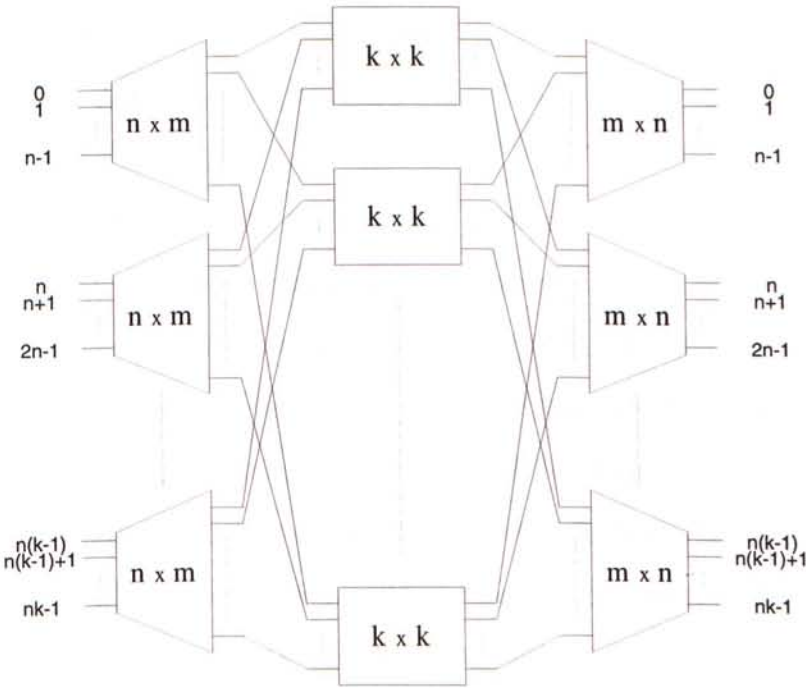


Figure 1.3: A 3-stage Clos network with k switching modules in first stage and last stage, m switching modules in central stage

diverse traffic characteristics ranging from constant bit-rate (CBR) to highly bursty, variable bit-rate (VBR) will be integrated together. We want to use the available resources efficiently so that all the available resources in the network can be used by all kinds of services, and an optimal statistical sharing of the resources can be obtained.

In circuit switching, the entire link bandwidth is assigned to a call for the whole period of conversation. In this way, the Quality of Services(QoS) of a connection such as data loss and end-to-end delay can be easily guaranteed. However, if a link can only hold one connection at a time, there will be a waste of bandwidth which results in poor utilization. This is especially significant as each optical channel nowadays can easily support bitrate up several hundred

megabits per seconds. To solve this problem, some form of multiplexing must be introduced. The Asynchronous Transfer Mode (ATM) is a promising technology to achieve this goal.

In an ATM network, data transmission is in cell format in that each data stream is divided into fixed length cells to transmit through the network. Each connection is treated as a Virtual Circuit (VC). The QoS parameters of a VC is set during call setup and is guaranteed along the entire call duration. The ATM network design and optimization at the call-level may be formulated in the framework of multirate, circuit-switched, loss networks with effective bandwidth encapsulating cell-level behavior[30]. Therefore, the design of multirate circuit-switched network is a foundation of ATM network.

The idea of multirate circuit switching is similar to traditional circuit switching except that the connections can request for different transmission rates in a transmission link. A number of connections can share one link given that the total rate of the connections do not exceed the link capacity. In addition, each connection will retain the bandwidth of the path assigned to it by the network from start to finish. By assigning the peak bandwidth requirement to the connection, the QoS can be guaranteed so that there will be no data loss and the delay is fixed for the connection.

In 1989, Melen and Turner laid out the foundation for the study of multi-rate networks[6] by derived a sufficient non-blocking condition for Clos network. Later, Chung and Ross showed a simple necessary and sufficient non-blocking condition for the Clos network in which the bandwidth requirement can be continuous or discrete[10]. The derivations of non-blocking condition in [6] and [10] are only considered in the space-domain in the sence that they just showed the

required number of central modules to route the connections inside the Clos network. The possibility of contention in the time-domain such as the case of simultaneous arrival of different connections within same link is not considered. Also their study was concerned with the theoretical scheduling condition of a multirate switch rather than the performance aspect of a switch.

In this thesis, we propose a switch network architecture call *quantized Clos-network* which can support multirate circuit switching efficiently. In [13], Lea and Alyatama stated that continuous bandwidth requirement implies an infinite dimensional analytical model for the routing problem and they proposed a *bandwidth quantization* method to reduce the state of the analytical model of the routing problem. This leads to a simple model for performing the optimal routing in the multirate network. In real world ATM networks, the bandwidth used by a certain connection is calculated as the average number of cells of the connection that can be transmitted through the network within a time slot. Quantization simplifies routing and capacity allocation by converting the continuous scale to a finite, discrete scale.

Using bandwidth quantization, the bandwidth requirement of a connection can be represented in discrete form. We can use time-division-multiplexed (TDM) switch architecture[3][4] to handle the multirate environment. In a TDM switch, each input or output port carries multiplexed data signals from several sources. These data signals are divided into repetitive frames, each of which is further partitioned into a fixed number of time slots (or slots for short). In our case, the number of slots represents the *quantization level* and it the number of rates supported in the network. Each connection may be allocated more than one slot according to their bandwidth requirement. The call blocking probability

and the throughput of this switch can be obtained by solving a stochastic knapsack model of this particular switch architecture[11]. In [29], Kim computed the call blocking probability of heterogeneous circuit switched traffic in a multirate multicast time-multiplexed switch environment using arrival modulation technique. It states that a call may be blocked not only in overflow blocking, but also in slot contention blocking. *Slot contention blocking* is the blocking due to mismatched idle slots given that the capacity constraints of a call are satisfied. One of the solutions of resolving the slot contention blocking is to schedule the transmission of each connection. In this thesis, we adopt the incremental algorithm for TDM switching time slot assignment in [28] to schedule the transmission of connections. The complexity of the algorithm is $O(\lambda M)$ where λ is the connection bandwidth requirement and M is the quantization level.

The purpose of routing is to find a suitable path inside the network such that the capacity of the path fulfills the bandwidth requirement of the connection[6][10]. Since the bandwidth of the network is not quantized and the path of the connection is not allowed to split, the link utilization of the switching module in central stage of the Clos network is low. With bandwidth quantization, *connection splitting* can be performed so that a connection can be routed through different paths inside the network. Consequently, an increase in link utilization results since fragmented bandwidth available on different links can now be filled up by the splitted connection requests. A direct consequence is the reduction of central switching modules which is needed to achieve non-blocking for multirate traffic when compared with the results of [6]and[10].

The quantized Clos network we propose here is based on the 3-stage Clos network. However, there will be M virtual Clos networks instead of one in the

quantized Clos network where M is the quantization level of the network. In this thesis, we will focus on the scheduling of cell transmission and the routing of connection in the quantized Clos network at cell-level since most data traffic transmitted in ATM network is in cell format. In [27], there is a detail discussion on routing of connection in quantized Clos network at connection-level. In addition, [27] also proposed a routing algorithm based on edge-coloring of weighted bipartite multigraph.

The performance of the multirate circuit switching in quantized Clos network will be the main focus on this thesis. We propose different implementations of quantized Clos network using different switching modules such as input-buffer switching modules, output-buffer switching modules and no-buffer switching modules. We will investigate which combination of switching modules gives the best performance of the quantized Clos network.

The organization of this thesis is as follows. In Chapter 2, we will describe the principle of multirate circuit switching in quantized Clos network. The definition of bandwidth quantization and connection splitting will be given. Then we will show the non-blocking conditions, the capacity allocation and the route assignment of the quantized Clos network at connection-level. In addition, we will state the difference between connection-level switching and cell-level switching and discuss the no rescheduling condition, the capacity allocation and the route assignment of the quantized Clos network at cell-level. The modified incremental algorithm of time slot assignment will also be presented. Chapter 3 will focus on the performance of different implementation schemes of the quantized Clos network. We use three different types of switching modules to implement each stage of the quantized Clos network : input-buffer switch, output-buffer

switch and no-buffer switch. We will evaluate the performance such as the time complexity of the algorithm used on different schemes, connection delay and cell delay inside the network when buffered switching modules are used. Conclusions and suggestions for future research will be given in Chapter 4.

Chapter 2

Principles of Multirate Circuit Switching in Quantized Clos Network

In this chapter we will show the principle of multirate circuit switching in quantized Clos network. We will describe the multirate environment in Clos network such that how the Clos network can handle multirate circuit switching. We will also introduce the concepts of bandwidth quantization and connection splitting in a Clos network. We will show the difference between routing in the call level and the cell level. We propose to use a time slot assignment scheduling algorithm to schedule transmission of cells in the Clos network so that there will be no cell contention inside the Clos network.

2.1 Formulation of Multirate Circuit Switching

In traditional circuit switching networks, the bandwidth of a link is dedicated wholly to a connection over the entire duration of conversation. Even though the call may be idle from time-to-time (e.g., when the person pauses speaking), the unused bandwidth cannot be allocated to another call, resulting in low network utilization.

To increase the utilization of the network, packet switching can be used. In packet switching, information is divided into a number of data blocks called packets for transmission. The packets may be of fixed length or of variable length. With packet switching, several connections can share the bandwidth of a particular link, provided that their aggregate bandwidth requirement does not exceed the link capacity. In this way, the network resources can be better utilized. In addition we can allocate different bit-rate to different connections so that no unnecessary bandwidth will be wasted. Under this environment, the Clos network must be able to switch circuits with different bit-rates. This is referred as the multirate circuit switching problem. Basically, a connection from input i to output j with bandwidth request ω will be blocked externally if there is no excess capacity in the input and the output link to accommodate the request. Mathematically, the call with bandwidth request ω will be rejected if

$$\sum_j \lambda_{Aj} + \omega > I_A \quad \text{or} \quad \sum_i \lambda_{iB} + \omega > O_B ,$$

where λ_{ij} is the total aggregate data rate from input i to output j and I_A and O_B are the available capacities of input A and output B respectively.

There has been a lot of work on the non-blocking conditions of the Clos network in the multirate environment. In [6], Melen and Turner proved that

by limiting the external link utilization β to be less than or equal to 0.5, the Clos network will be non-blocking if the number of middle-stage modules m is greater than or equal to $2n - 1$. The restriction on β can be relaxed if connection splitting is permitted. With connection splitting, the path through the switch followed by the packets belonging to the same connection need not be the same. In this way, the residual bandwidth of the internal links can be better utilized. A drawback of this approach is that packets may arrive in the wrong sequence and packet queuing will be required.

For the multirate Clos network to be practical, however, a routing algorithm must be derived which can route the calls quickly and efficiently. It was shown in [13] that if each connection is allowed to have a continuous bandwidth requirement, the routing problem will become unmanageable. To overcome this problem, we can quantize the bandwidth requirements into a number of discrete levels. In [27], Kwok studied the problem of applying bandwidth quantization in a multirate Clos network. Since the results are related to the work in this thesis, we will discuss them in the following sections.

2.2 Call Level Routing in Quantized Clos Network

The concept of bandwidth quantization is very simple. Bandwidth quantization is equal to dividing the bandwidth into several parts and allocate appropriate number of parts to the connections according to their bandwidth requirement. Suppose the bandwidth of every link is normalized to one, under uniform quantization with M quantization levels, the bandwidth is equally divided into

M parts and each part consists of $1/M$ of the whole bandwidth. The bandwidth request of each connection is then quantized from continuous bandwidth requirement to discrete bandwidth requirement and the number of bandwidth parts are allocated to the connection according to its quantized value. In order to guarantee the connection bandwidth requirement, the allocated bandwidth may be more than what the connection has requested since the quantized value may be larger than its actual request value. Connection splitting is to assign different routes to the packets of the same connection to go through in the network. It increases the flexibility of choosing the route for the connection because the bandwidth request can be splitted and more than one central module can share part of the request. Consequently, the number of central modules to achieve non-blocking in multirate environment can be reduced and the internal link utilization can also be increased. With bandwidth quantization and connection splitting, Kwok has derived the non-blocking conditions for the Clos network under multirate traffic condition at call-level[27] with bandwidth quantization and connection splitting, if $m \geq n$ where m is the number of central modules in the Clos network and n is the number of input ports in the first stage modules, the Clos network is rearrangeably nonblocking under multirate traffic. If $m \geq \lceil \frac{2Mn-1}{M} \rceil$ where M is the quantization level, the Clos network is strictly nonblocking[27]. In the special case where $M = 1$, the condition becomes $m \geq 2n - 1$ which matches the well known results[3, 4] of classical switching theory. For $M > 1$, only $2M$ central modules will be sufficient. In addition to the derivation of non-blocking condition in multirate environment, routing algorithm at call-level is also presented in [27]. The routing algorithm is based on edge-coloring of a weighted bipartite multigraph representing the connection

configuration and implemented using a specially designed matrix called *Capacity Allocation Matrix*(CAM). Figure 2.1 shows a CAM in which each row of the matrix represents an input or output module of a quantized Clos network. The matrix is divided into two parts, the resources matrix R and the allocation matrix A . The resources matrix stores how much capacity of the quantized Clos network can still be used. It indicates the available bandwidth in each central module. The allocation matrix records the route assignment and capacity assignment of the existing connections inside the quantized Clos network.

		1	2	3	...	C_0	C_1	C_2	...
Input Module 0	I_0	$4C_0$				6	10		
Input Module 1	I_1		$3C_0$			3	8		
	.				.				.
	.				.				.
Output Module 0	O_0	$4C_0$				7	10		
Output Module 1	O_1		$3C_0$			5	7		
	.				.				.
	.				.				.
Allocation Matrix, A						Resources Matrix, R			

Figure 2.1: Capacity Allocation Matrix

The routing and capacity assignment of connections are done simultaneously using the CAM. When a connection arrives, the algorithm will check if the bandwidth request exceeds the link capacity. If there is enough bandwidth for the new connection, the algorithm will follow the flow chart shown in Figure 2.2 to setup the connection.

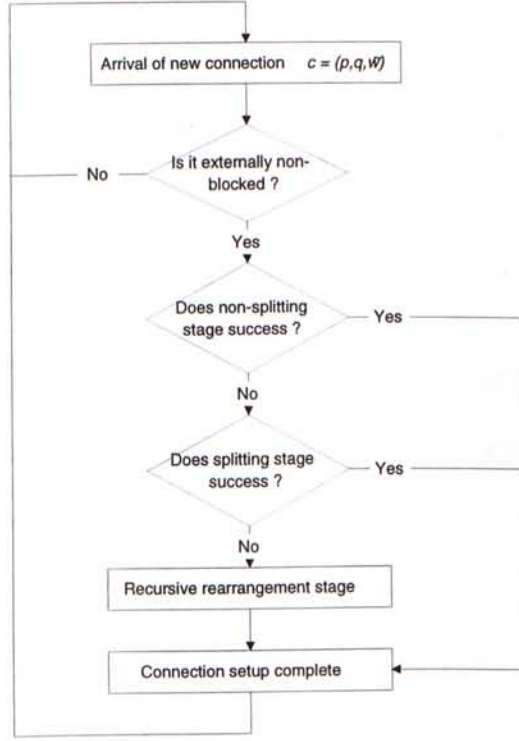


Figure 2.2: Flow chart of the routing algorithm at connection-level

In Figure 2.2, the non-splitting stage tries to assign exactly one central module to setup connection c . The splitting stage tries to setup connection c using more than one central module. The route assignment and capacity assignment in these two stages are done by searching the corresponding rows and columns of the CAM. Figures 2.3 and 2.4 are examples to show how the algorithm assigns the route and capacity to connections in the non-splitting stage and the splitting stage respectively. In the splitting stage, if the bandwidth request of the connection cannot be satisfied, bandwidth request will be fulfilled by rearranging existing connections. The proposed algorithm takes the recursive approach to traverse the weighted alternate tree and interchange the two colors simultaneously[27].

	1	2	3	4	5	C_0	C_1
I_0	$2C_1$	$1C_1$				4	1
I_1			$1C_0$			3	4
I_2				$3C_0$	1	1	4
O_0	$2C_1$					4	2
O_1		$1C_1$			1	4	3
O_2			$1C_0$	$3C_0$		0	4

Before connection setup

	1	2	3	4	5	C_0	C_1
I_0	$2C_1$	$1C_1$				4	1
I_1			$1C_0$			3	4
I_2				$3C_0$	$1C_0$	0	4
O_0	$2C_1$					4	2
O_1		$1C_1$			$1C_0$	3	3
O_2			$1C_0$	$3C_0$		0	4

After connection setup

Figure 2.3: Example of non-splitting connection setup

	1	2	3	4	5	6	C_0	C_1
I_0	$2C_1$	$1C_1$					4	1
I_1			$1C_0$			4	3	4
I_2				$3C_0$	$1C_0$		0	4
O_0	$2C_1$						4	2
O_1		$1C_1$			$1C_0$	4	3	3
O_2			$1C_0$	$3C_0$			0	4

Before connection setup

	1	2	3	4	5	6	C_0	C_1
I_0	$2C_1$	$1C_1$					4	1
I_1			$1C_0$			$3C_0, 1C_1$	0	3
I_2				$3C_0$	$1C_0$		0	4
O_0	$2C_1$						4	2
O_1		$1C_1$			$1C_0$	$3C_0, 1C_1$	0	2
O_2			$1C_0$	$3C_0$			0	4

After connection setup

Pass	x	$\min(R(I_p, C_x), R(O_q, C_x), \omega)$	Assignment	ω after step 2
1	0	$\min(3, 3, 4) = 3$	$3C_0$	1
2	1	$\min(4, 3, 1) = 1$	$1C_1$	0

Details of splitting connection setup

Figure 2.4: Example of connection setup with splitting

When any connection $C_R = (s, t, \tilde{\omega}_R)$ is released, the algorithm will examine existing splityrf connections and will re-route them through a single central module to make them non-splitting, if possible. Due to the structure of the Clos network, additional central module capacities are only available through switching modules I_s and O_t . Therefore, the algorithm will only consider those splitted connections sharing the same input or output modules with C_R .

2.3 Cell Level Routing in Quantized Clos Network

The routing algorithm at call does not indicate how to multiplex different connections within a input link or demultiplex different connections within a

output link. In this section, we will focus on the realization of multirate circuit switching on ATM cell-level switching network and using the propose quantized Clos network as the switch architecture.

2.3.1 Traffic Behavior in ATM Network

In a circuit switching network, the transmission rate of each connection is the same and it remains constant such that once the route of the connection is assigned and the transmission time is scheduled, the traffic of the connection can be simply managed by the switching network.

However, the behavior of traffic in an ATM network is very different from circuit switching network and can be considered in several levels: *network level*, *call level*, *burst level* and *cell level* as shown in Figure 2.5. At the network level, a number of connections are carried. At the call level, a call (or session) lasts for the duration of the connection between the end users. A call in turn can be partitioned into a sequence of alternate burst (*ON*) and silence (*OFF*) periods. These periods affect the burst level performance of an ATM network. During the *ON* period (T_{ON}), a steam of ATM cells is emitted at regular intervals. During the *OFF* period(T_{OFF}), no cells are transmitted. An important attribute of each level is its time scale. This is governed by the mean inter-arrival time of entities in that level during an activity period in the upper level. Usually, time scales of different levels are substantially different.

Implementation of circuit switching in ATM network is simple since the traffic behavior of circuit switching is regular and periodic. Therefore, we can combine the burst level and cell level altogether as one level and we assign the peak bandwidth capacity to each connection. The multiplexing can be done on

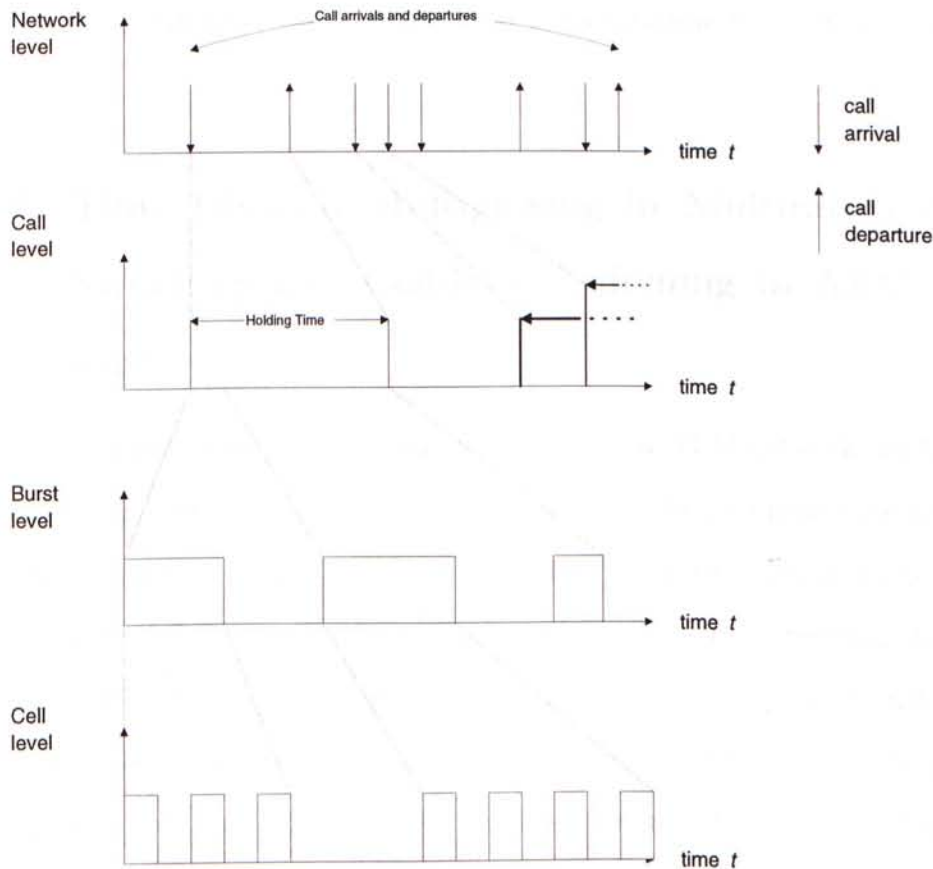


Figure 2.5: Different level time scale in ATM network

burst level instead of cell level and simplify the switch complexity.

Although the algorithm proposed in [27] can assign route and capacity to each multirate connection at call-level, it cannot solve the problem of simultaneous arrival of traffic from different connections at burst level. This is because it has not considered the time scale difference between the call level and the cell level. Capacity may be available on call level for different connection but it may not be available on cell level since cells from different connections may arrive simultaneously. Cells may be blocked at cell level so that the non-blocking condition at call level may not hold on cell level. The only way to make the routing

assignment at call level work at cell level is to schedule the cell transmission time.

2.3.2 Time Division Multiplexing in Multirate Circuit Switching and Cell-level Switching in ATM Network

In the realization of multirate circuit switching in ATM network, we have to place a multiplexer at the front of the switching network to multiplex different connections together to share a single link. In order to avoid cell loss due to cell contention inside the switch, the multiplexer has to do the scheduling work for the connections or buffer is needed inside the switch. Using buffer switch will reduce the scheduling work of the multiplexer but the delay cannot be guaranteed for each multirate connection. Therefore, the switch we used is bufferless so that we have to schedule the cell transmission to avoid any loss inside the switch. Since the traffic characteristic of multirate circuit switching is periodic, we can use a frame structure for aligning the cells in switching after they enter the multiplexer. There is no frame structure outside the switch such that the incoming multirate traffic will arrive independently. The frame size is equal to the quantization level so that the bandwidth requirement of each connection in each frame is the same. The switch pattern is repeated from frame to frame until any connection release or new connection arrive.



Figure 2.6: Realization of multirate circuit switching in cell-level

2.3.3 Cell Transmission Scheduling

We adopt the TDM scheme to implement our algorithm in cell-level switching in ATM network. As mention in Section 2.2, we quantized the bandwidth capacity uniformly in M levels, we set the TDM frame size equal to M such that there are M time slots within one transmission time unit. We transform the time slot unit into cell unit so that one time slot equal to $\frac{C}{M}$ cells where C is the number of cells transmitted within one transmission time unit. For simplicity, we will use the term “time slot” to represent the cell in this chapter.

To implement our algorithm in time slot level, we have to schedule time slot to each connection. However, we cannot set the time slot to connection without conflict when we get the route assignment and the capacity assignment from the Capacity Allocation Matrix(CAM) because the algorithm did not consider the time slot level route assignment. Time slot contention may occur even though we have sufficient number of central modules to make the network non-blocking. The time slot contention may occur when the capacity mismatch with time slot between different stage switching modules.

Consider a 6×6 multirate quantized Clos network with quantization level

equal to 6(Figure 2.7) and there are 11 connections exist in the network. If a new connection request from input link 3 to output link 0 with bandwidth request 2, there exist a output contention when allocating this new connection since the only available slots in input link 3 is 1 and 2 but the time slot 1 and 2 in output link 0 have already occupied by connection 1. In this situation, although there are sufficient capacity to establish a connection strictly non-blocking, we still cannot establish such connection without any time slot rearrangement. In addition, we cannot intelligently allocate time slot to connections to avoid any output conflict for the future.

From the previous example, we can see that scheduling of time slot is important and must be performed before assigning the route to connection. There are two methods to prevent the time slot contention The first one is to restrict the link capacity such that the internal link capacity is larger then the external link capacity. The second way is to reschedule the time slot assignment of existing connection to avoid contention to new connection.

The scheduling of time slot can be done by time slot interchanger(TSI). Kwok proposed using time slot interchanger at each stage in the Clos network when realizing the route assignment in packet level[27]. This will avoid the internal and external conflict but the delay will be 3 frames time because each time slot interchanger has to store up a frame of traffic before it interchanges the time slots. In here, we propose a global scheduling algorithm base on an incremental algorithm for TDM switching assignment which proposed in [28]. An incremental time slot assignment(TSA) algorithm computes the time slot assignment by making incremental changes to a previous assignment to accommodate the changes in traffic. This kind of algorithm is suitable for high-speed

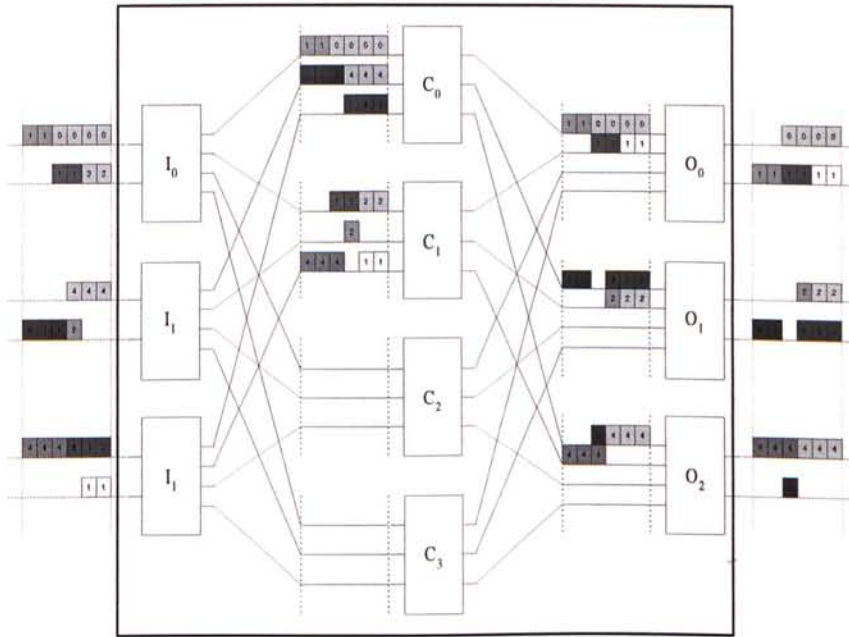


Figure 2.7: Example of a 6×6 multirate quantized Clos network

	1	2	3	4	5	6	7	8	9	10	11	N	C_0	C_1	C_2	C_3
I_0	$4C_0$	$2C_0$			$2C_1$			$2C_1$					0	2	6	6
I_1						$3C_0$	$1C_1$		$1C_0$		$2C_0$	2	0	5	6	6
I_2			$2C_1$	$3C_0$						$3C_1$			3	1	6	6
O_0	$4C_0$	$2C_0$	$2C_1$					$2C_1$				2	0	2	6	6
O_1				$3C_0$	$2C_1$		$1C_1$				$2C_0$		1	3	6	6
O_2						$3C_0$			$1C_0$	$3C_1$			2	3	6	6

Figure 2.8: The CAM of the 11 connections in Figure 2.7

switching network since it doesn't need to recompute the entire TSA for each frame. The incremental algorithm is based on the correspondence between the problem of computing an incremental TSA and the rearrangement problem in a 3-stage Clos network. To illustrate the correspondence between the two problem consider the Clos network in Figure 2.9, where the middle stage consists of M switch modules, each of size $N \times N$. The outer stages consist of switch modules of size $M \times M$ and there are exactly N of them in each. The middle stage M

switch modules represent the M time slots patterns of the $N \times N$ switching network. The interchange of time slots is equivalent to rearrange a connection assignment from a central module to another central module.

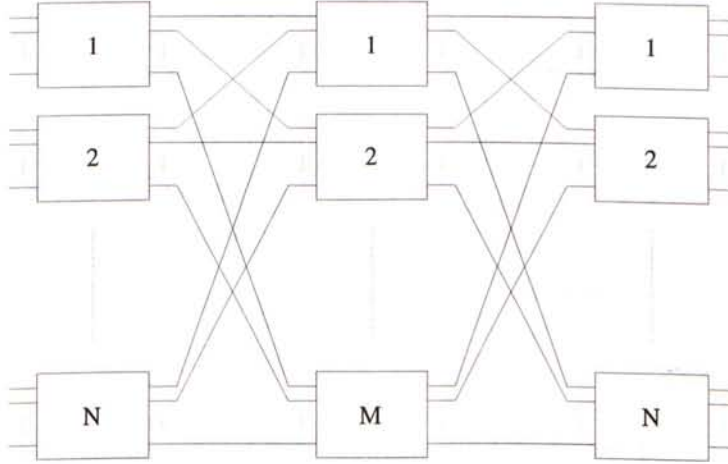


Figure 2.9: Clos network model for the TSA problem

In order to reduce the complexity of the time slot assignment algorithm such that we do not need to reschedule the existing connection TSA, we can restrict the external link capacity. Consider the corresponding Clos network model of the TSA problem in which the first stage modules are of size $M \times (2M - 1)$ and the third stage modules are of size $(2M - 1) \times M$. The middle stage modules are of size $N \times N$. This is a strictly non-blocking network such that a connection from any free pair input-output port can be established without rescheduling any existing connections[3]. Therefore, if we only use M time slots out of $(2M - 1)$ time slots, we can schedule the connection TSA without reschedule any existing connection TSA. The TSA algorithm under this condition can be simply search for a free time slot only without any time slot rescheduling. However the link utilization is limited to about 50%.

Although the computation complexity is low when rescheduling is not need

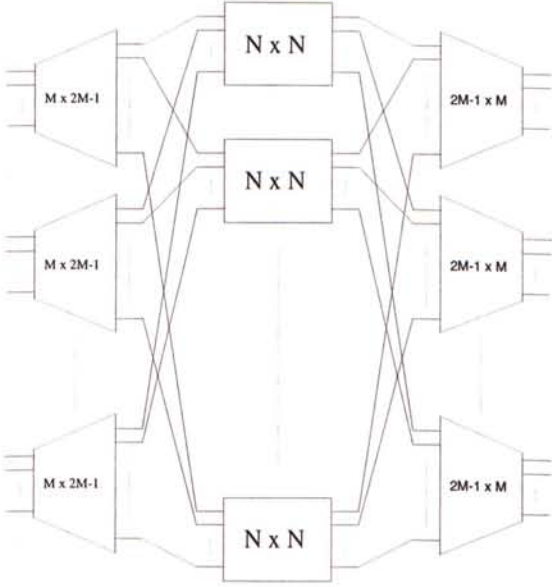


Figure 2.10: Correspondence Clos network model of the TSA problem which using M time slots out of $2M - 1$ time slots

to perform, the link utilization is also low. Rescheduling of time slot assignment can use up the whole link bandwidth capacity. The rescheduling algorithm is introduced here. Where a new connection comes in with (x, y, ω) , where x is the input port number, y is the output port number and ω is the bandwidth request, we have to find ω central modules in the Clos network model which shown in figure 2.9 to connection the x^{th} module in the first stage and the y^{th} module in the last stage. The allocation algorithm is shown below. We first search for central modules which input port x and output port y are free and allocate 1 unit of bandwidth to each of them. After all modules are searched and all bandwidth are allocated, the time slot assignment is done. Otherwise we have to rearrange the time slot assignment of connections to accommodate the unallocated bandwidth.

Time slot assignment algorithm

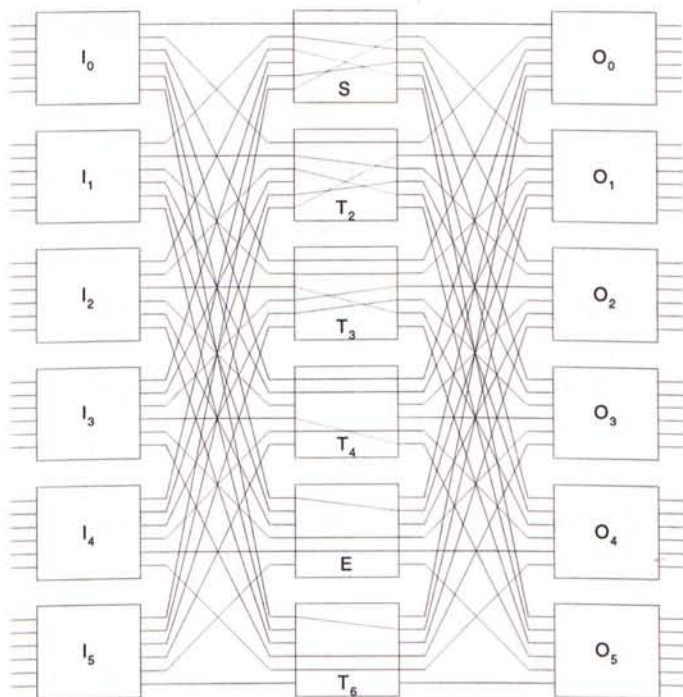


Figure 2.11: The time slot Clos network model of figure 2.7

$\text{tsa}(x, y, \omega)$

1 $i \leftarrow 1$

2 while $(i < M)$ and $\omega \neq 0$ do

2.1 if the x^{th} input and y^{th} output on i^{th} central module are free, allocate
 i^{th} module to $\text{connection}(x, y, \omega)$ and $\omega \leftarrow \omega - 1$

2.2 $i \leftarrow i + 1$

3 if $\omega = 0$, end.

4 while $\omega \neq 0$ do

4.1 Find central module S such that the x^{th} input of it is free.

- 4.2 Find central module E such that the y^{th} output of it is free.
- 4.3 $n \leftarrow y, m \leftarrow x, LIST \leftarrow 0$
- 4.4 Find the output port n which connect to input port m in module E .
If n does not exist goto 4.8
- 4.5 $LIST \leftarrow LIST \cup (m, n)$ (add edge (m, n) to the list).
- 4.5 Find the input port m which connect to output n in module S . If m does not exist goto 4.8
- 4.6 $LIST \leftarrow LIST \cup (m, n)$ (add edge (m, n) to the list).
- 4.7 goto 4.4.
- 4.8 while($LIST \neq 0$) do
 - 4.8.1 Find an element (m, n) in $LIST$
 - 4.8.2 If (m, n) is in module S change it from module S to E . Otherwise change it from module E to S .
 - 4.8.3 $LIST \leftarrow LIST - (m, n)$ (remove edge from $LIST$).
- 4.9 Allocate central module S to connection(x, y, ω) and $\omega \leftarrow \omega - 1$.

The rearrangement procedure is modified from the algorithm *allocate* proposed in [28] and it is elegantly modeled by a graph-theoretic formulation. The algorithm will search for two central modules S and E such that input port x on module S is free and output port y on module E is free and it will construct a bipartite multigraph $G = (X, Y, E)$, where X and Y are the two sets of vertices and E is the set of edges from the two central modules. The vertex set X represents the input ports of the central module and vertex set Y represents the output ports of the central module. An edge exists from vertex $i \in X$ to

vertex $j \in Y$ if and only if there is a connection assigned to module S or module E from input port i to output port j . The edges corresponding to module S and module E are distinguished by coloring them distinctly. Observe that the degree of each vertex in this graph model is at most two and no vertex has two incident edges with identical colors.

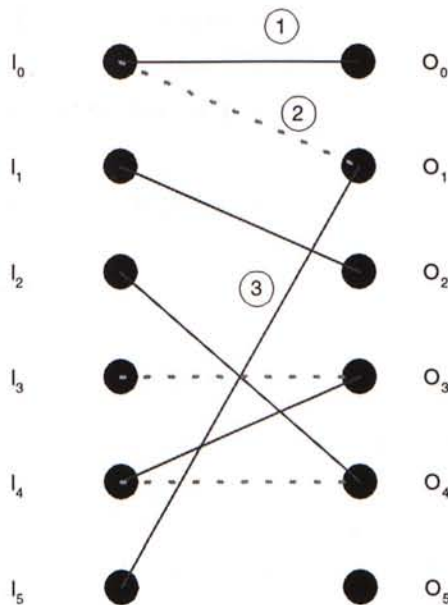


Figure 2.12: Graph model $G(X, Y, E)$ corresponding to module S and E of figure 2.11

To illustrate the above graph model, figure 2.12 shows the graph corresponding module S and module E of example in figure 2.11. Edges corresponding to module S are shown by solid lines and those corresponding to module E are shown by broken lines. Assume that the color of solid lines and broken lines are red and green respectively. Allocating the new connection $(x, y, 1)$ is equivalent to adding a new edge in the multigraph G between vertex $x \in X$ and $y \in Y$. Since input port x in module S is free and output port y in module E is free, the vertices $x \in X$ and $y \in Y$ each has exactly one incident edge but in different

color. We have to modify the coloring of some of the edge such that the new edge can be added without violate the coloring scheme.. To achieve this objective, we find a path (sequence of non-repeating vertices):

$$P = v_1 v_2 \cdots v_k$$

in multigraph with the following properties.

1. The start vertex v_1 is the vertex $y \in Y$.
2. The ending vertex v_k has degree one.
3. $(v_i, v_{i+1}) \in E$, for all $1 \leq i < k$.

Such a path S is guaranteed to exist because there is at least one vertex other than $y \in Y$ with degree one and every vertex has a degree of at most two. Vertices v_i with $i = 1, 3, \dots$ along the path P belong to the set Y and vertices v_i with $i = 2, 4, \dots$ belong to X . In addition, if (v_1, v_2) is an edge with red color, then (v_i, v_{i+1}) is an edge with red color if i is odd and green otherwise. The new edge can be introduced by simply flipping the color of each edge along the path P . This re-coloring ensures that the only edge that was incident on vertex $y \in Y$ is now colored green. If we now introduce the new edge between vertex $x \in X$ and vertex $y \in Y$ and color it red, then every vertex in the multigraph would have at most one incident edge of each color.

To illustrate the rearrangement process with an example, consider the multigraph in Figure 2.12. Assume that we need to allocate one unit of traffic from input port 3 to output port 0. This corresponds to introducing a new edge between vertex 3 on the left-hand side and vertex 0 on the right-hand side of the

multigraph. Note that the edges already incident to these vertices are of opposite color, so the new edge cannot be added without re-coloring the multigraph. It is easy to see that a path of alternating edge colors exists in the multigraph starting from the vertex $0 \in Y$ and ending at the vertex $? \in X$. The edges along this path are labeled ①, ②, ③ in Figure 2.12. Flipping the color of each of these three edges leaves vertices $3 \in X$ and $0 \in Y$ each with a single incident edge of red. The new edge can, therefore, be introduced between them and colored green.

The algorithm implicitly used the graph model $G(X, Y, E)$ described earlier to perform the reallocation. The loop beginning in step 4.4 of the algorithm follows the path P of alternating edge colors until it terminates in a vertex of degree one. While traversing the path, an unordered list $LIST$ is used to maintain a list of the edges traversed. Once the entire path is known, a separate loop, beginning in step 4.8, is used to flip the colors of the edges in $LIST$. This flipping of colors is achieved by exchanging the corresponding entries between central module S and E .

2.3.4 Capacity Allocation and Route Assignment at Cell-level

The routing algorithm in the quantized Clos network at cell-level is modified from the routing algorithm at connection-level which proposed in [27]. First of all, when a connection $c(p, q, \tilde{\omega})$ arrival, without external blocking, we schedule its time slot assignment by the $tsa()$ algorithm. The routing assignment and capacity allocation is then done by the Capacity Allocation Matrix (CAM) which

we search for suitable central modules for the connection. The searching of central modules is in condition of the time slot assignment of the connection such that the target central module must be available in the time slot assignment of the connection. The routing algorithm will not perform the recursive rearrangement procedure which proposed in [27] because of the complexity of time and space rearrangement. In addition the algorithm will not re-route the existing split connection to make them non-splitting because it will cause the rearrangement of time slot assignment and route assignment of existing connections and affect the existing switching pattern of each switching module.

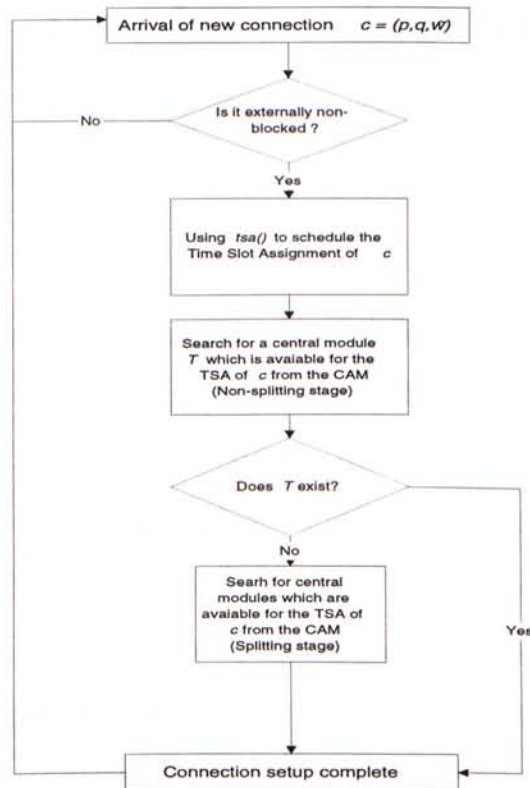


Figure 2.13: Flow chart of the routing algorithm at cell-level

Chapter 3

Performance Evaluation of Different Implementation Schemes

In this chapter, we will investigate the performance of different implementation scheme of the quantized Clos network. The quantized Clos network is constructed by switching modules so that different kinds of switching modules such as bufferless switching modules, input-buffered switching modules and output-buffered switching modules can be used. Using different buffered switching module design will have different performance of cell-delay and requirement of buffer size to prevent cell loss. In contrast to buffered switch, bufferless switch will require more complicated algorithm to control the switching pattern of the modules in the Clos network. Therefore the performance of these two kinds of switch design will be different. We will compare the cell-delay performance of using bufferless switch modules, input-buffered switch modules and

output-buffered switch modules. In addition, the computation time complexity and the network space complexity will also be investigated. All the cell delay performance results are based on simulation.

3.1 Global Control and Distributed Switching

In many proposed Clos network type large scale ATM switch architecture[7, 26], the switch processes are performed distributively in each switching module. Each individual module is able to make the routing decision according to the local information. The switching control of such network can be completely distributed over the three-stage so that the network do not need to globally control the switching pattern of the modules in each time slot. This reduces the global control complexity when the size of the switching network becomes large. The global control of the network is the traffic control to ensure the Quality of Service(QoS) requirement of each connection.

In the quantized Clos network, the global control is done by the routing algorithm we proposed in the pervious chapter. The algorithm search for the route such that the bandwidth is guarantee to all connections. In addition, the routing and the capacity allocation of each connection is assigned by the CAM and the switching patten of each stage switch module is also pre-determined. The switching pattern of each module is repeated for every M time slots unless there is a new connection arrival. Therefore, every switch module are independent of each other and the switch module design of every stage module can also be independently different.

3.2 Implementation Schemes of Quantized Clos Network

The implementation scheme of the quantized Clos network can be divided into two groups. The first one is using bufferless switch modules to construct the three stage modules. The second one is using both bufferless and buffered switch modules to construct. The difference between these two groups is the control complexity and the cell delay performance. The detail discussions will be presented in Section 3.3. In addition, the number of central modules in all the implementation models is double of the number of input link per first stage modules whether the routing of connection is without any rearrangement of existing connections.

3.2.1 Classification of Switch Modules

Input-buffered Switch

Cells are queued at the input in an input-buffered switch. Each input has its own buffer allocation size and the buffers are mostly served as a first-in-first-out(FIFO) so that at the beginning of each time slot only the cells at the heads of the FIFO's content for access to the switch outputs. The cells that have lost contention will block other cells which are queued behind it in the FIFO, causing head-of-line blocking. Therefore, under uniform random traffic, the maximum throughput of an input-buffered switch is limited to 0.586[16]. The throughput can be increased by various switch design strategies. The look-ahead contention resolution was proposed in [17] and it removed the FIFO constraint on the input

queues so that the blocked cells can also access to the switch lost contention output. The throughput is improved as the look-ahead window size increases but the increase of the windows size will also increase the contention-resolution overhead and the implementation complexity of the switch circuit design[14].

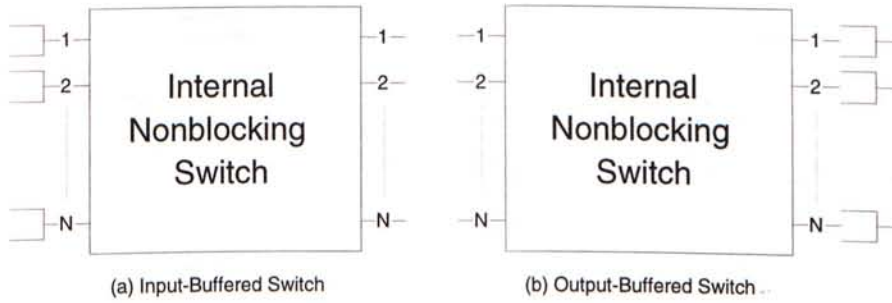


Figure 3.1: Input and Output buffered switch

Output-buffered Switch

Figure 3.1(b) shows an output-buffered switch. Buffers are placed at each output port and cells with same destination output port but different inputs can be simultaneously buffered at the output port. The throughput-delay performance of output-buffered switch is better than input-buffered switch under uniform random traffic assumption[16] but the implementation complexity is higher than the input-buffered switch where in earlier multistage space switch architectures, output queueing cannot be used unless switch speedup is utilized[2]. However the Knockout switch[8] and the Sunshine switch[9] can overcome the internal switch speed requirement and can also be self-routing.

Figure 3.2 shows the basic architecture of a Knockout switch. It uses n buses to broadcast all input cells to all outputs. At each output, filters are used to select the cells addressed to it. Contention resolution is performed by a

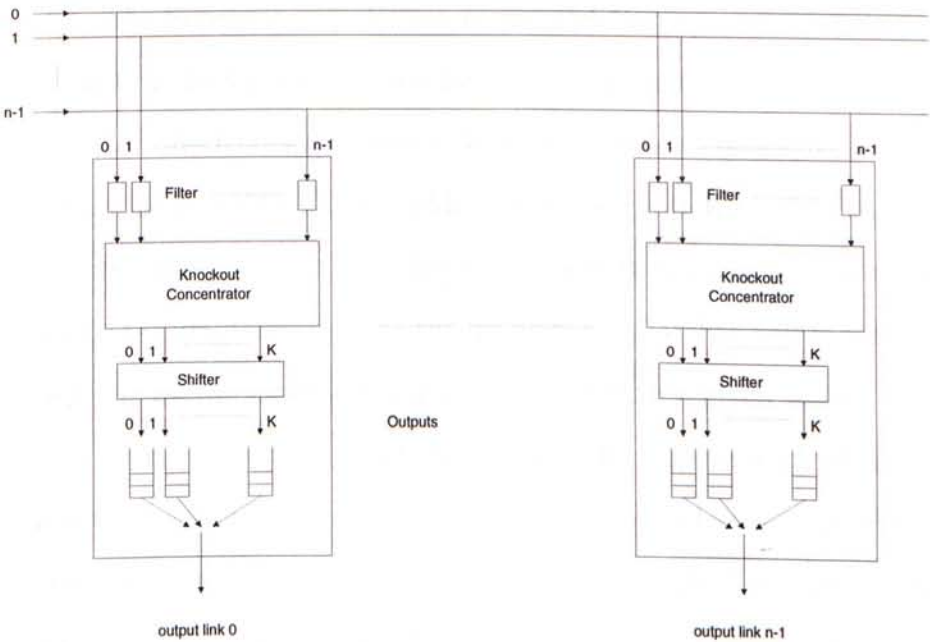


Figure 3.2: The modular Knockout switch[8]

Knockout concentrator at the output that selects at most K cells to feed into a logical FIFO queue. The excess cells are dropped in the knockout concentrator. The Knockout concentrator is composed of K sections, each corresponding to a tournament that selects one winning cells. The n inputs of the concentrator are connected to the first section. The losing cells at each section are feed into the next section so that after K sections, K or fewer cells are selected as winners. Each section is a binary-tree interconnection of 2×2 switch elements, and some delay elements are used to synchronize cells. The number of switch elements in the i -th section is $n - i$. The total umber of switching elements per concentrator is thus $(n - 1) + (n - 2) + \dots + (n - K) = nK - K(K + 1)/2$. In addition, the buffer is placed at each output port for buffering the simultaneously arriving cells.

The block diagram of the Sunshine Batcher-Banyan ATM switch is shown

in figure 3.3. The Sunshine switch uses K parallel Banyan routing networks to provide multiple paths for cell destined for any output port. This is similar to the technique used in the Knockout switch. With K Banyans connected in parallel, up to K cells with common output address may be routed, by the Banyans, to the appropriate output. Up to T additional cells with the same output address are trapped and recirculated back to the input of the Batcher Sorter for another pass through the overall switch. Recirculated cells from a given source can thus be routed in order of initial arrival at the switch. The trap network identifies the cells required to be recirculated; the concentrator and selector networks group them together and direct them to the recirculating network. The buffer is placed at each output port controller(OPC) for buffering the simultaneously arriving cells.

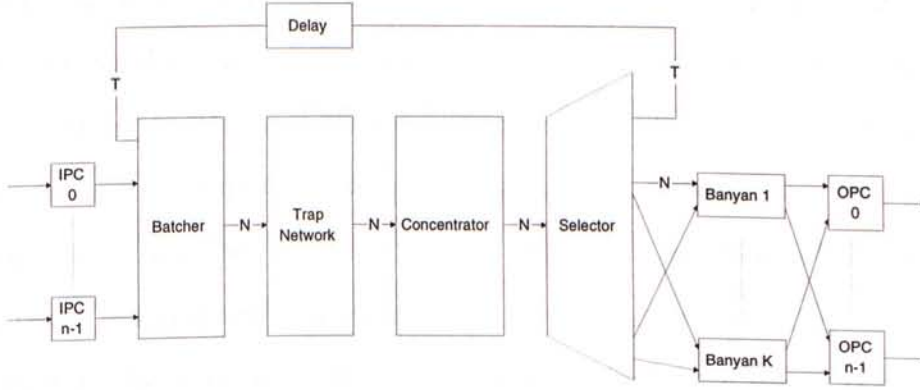


Figure 3.3: Sunshine architecture[9]

Assume the basic switch element in both Knockout switch and Sunshine switch is 2×2 switching nodes. The complexity of the Knockout switch and the Sunshine switch is of order KN^2 and $KN \log_2^2 N$ respectively[15, 2].

Shared-buffer Memory Switch

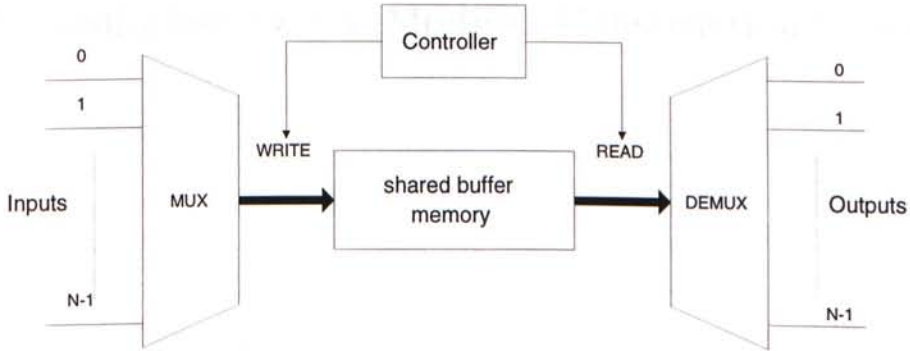


Figure 3.4: Block diagram of a shared buffer memory switch

The switch design which we discussed above are mainly space-division switch. In contrast to space-division switch, switch module can also be implemented by using memory switch will shared-buffer capability[18, 19, 31]. Figure 3.4 shows the block diagram of a shared-buffer memory switch. It consist of a single memory shared and accessed by all input and output links and managed by a central controller. In every time slot, the cells arrive on all input links are multiplexed and written into the common memory. The cells inside the memory are logical organized as separate queues on for each output links. Since all the memory can be randomly accessed, there will not be any head-of-line blocking which occur in input-buffered switch. The contention resolution can be done by programming the central controller. Therefore, the logical queue is not FIFO and the throughput-delay performance is raised[18]. However, the internal speed of the switch must be increased to $2N$ times the input link speed so that the cells at each input link can be written into the memory simultaneously and each logical queue inside the memory can be retrieved simultaneously in one time slot.

3.2.2 Bufferless Switch Modules Construction Scheme

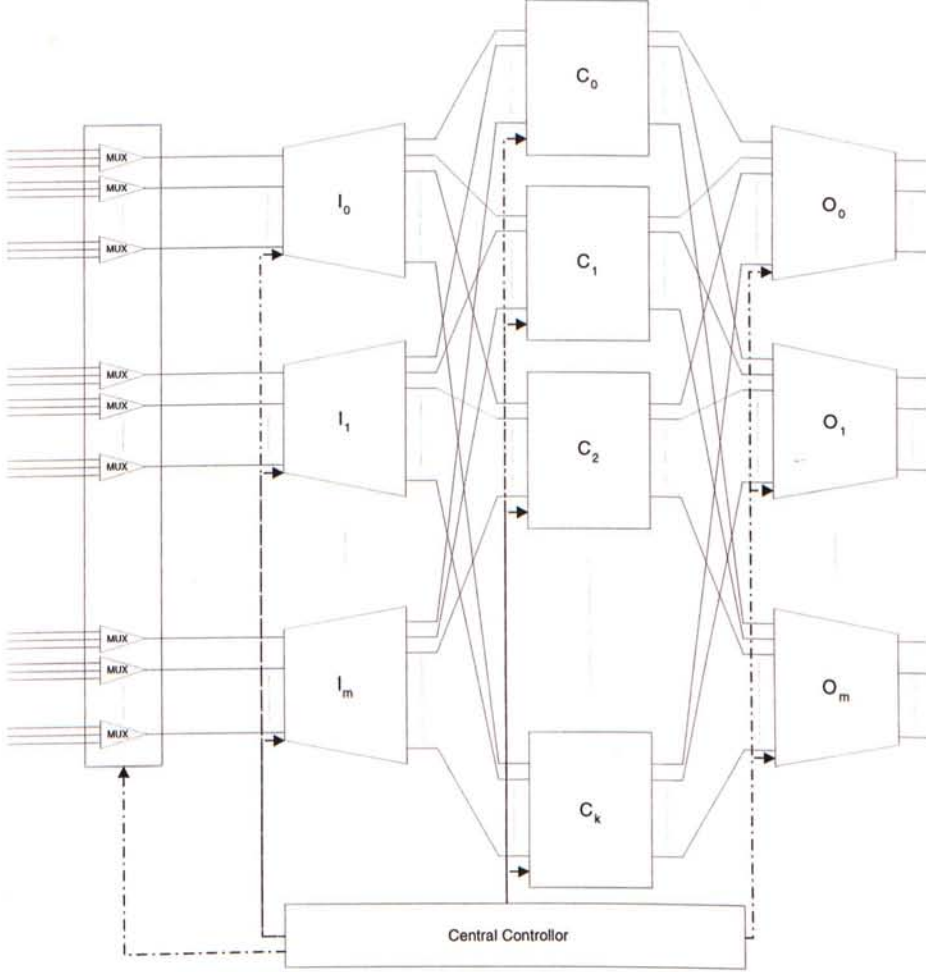


Figure 3.5: Bufferless switch modules construction scheme

In this scheme, all the switch modules used are bufferless. As we are considering the quantized Clos network under strictly non-blocking condition, the first stage modules are expansion networks where the number of output ports are double of the number of input ports. Similarly, the third stage modules are concentration networks where the number of output ports are half of the number of input ports. The central modules are Benes network with size of $\frac{N}{n} \times \frac{N}{n}$ where

N is the total numbers of inputs and n is the number of input ports in the first stage modules. Since there is no buffer in each module, cells may be dropped inside the network. In order to prevent cell loss inside the network, multiplexer must be placed in the front of each input link of the first stage modules and must have the scheduling capability in the sense that the cells enter the network will not have any contention over the three stages. The scheduling can be done by the algorithm which we proposed in the previous chapter.

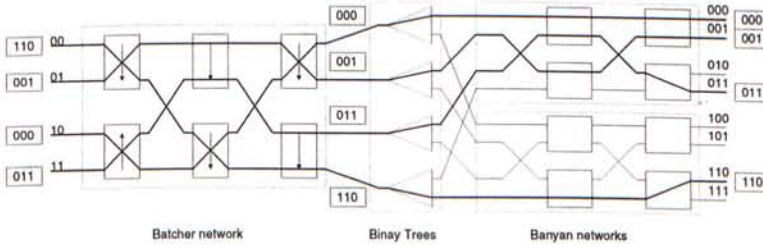


Figure 3.6: The routing of a 4×8 expansion network at first stage

We can use Batcher-banyan self-routing network to construct the first stage expansion network modules. The expansion network is composed by an $n \times n$ Batcher network followed by two $n \times n$ Banyan networks. According to the nonblocking property of expansion network [24] and the internal non-blocking property of the Banyan network[23], if the set of input cells to the expansion network is concentrated and has monotonic outputs, then every subsets of input cells to each banyan subnetwork of the expansion network is also monotonic and concentrated. Therefore, the routing of each cell can be performed by its own route assignment without any blocking inside the switch module. The route assignment bits is assigned from the central controller and is added by the multiplexer when the cell enters the network.

Similarly, the third stage modules can also be constructed by Batcher-banyan

network. The Batcher network is size of $2n \times n$ where the last $\log n$ stage n -bitonic sorter can be reduced. The banyan network is size of $n \times n$. Since the output packets of the batcher network is guaranteed to be monotonic and concentrated, the packets can self-route through the Banyan network without internally blocking.

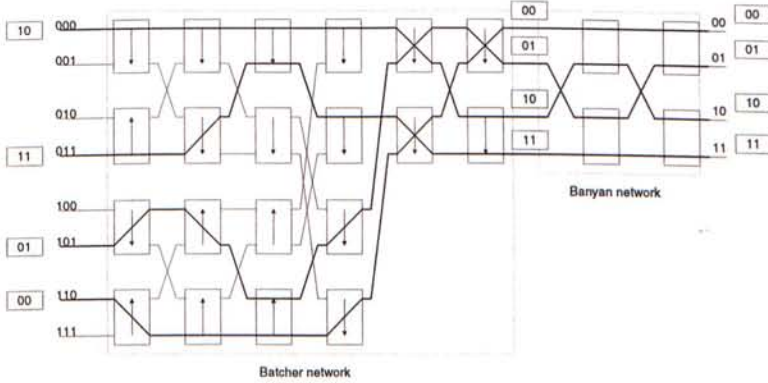


Figure 3.7: The routing of a 8×4 concentration network at last stage

We use the bufferless Benes network type switch module to construct central stage modules of the quantized Clos network. A Benes network is a rearrangeably non-blocking network such that any permuted input-output mapping can be realized on it[4]. In addition, Benes network can also be a self-routing network in the sense that the switch elements make use of the routing tag in the cell to perform switching. The routing tag of a cell in a Benes network is composed of two parts: the last $\log_2 k$ bits are the output address and the first $\log_2 k - 1$ bits are representing the routing path and can be calculated by some existing algorithms such as the looping algorithm in [5] or the parallel algorithm which proposed in [25]. The calculation of the routing tag can be perform according to the cell arrival at each time slot and is added to the header of the cells when they enter the network.

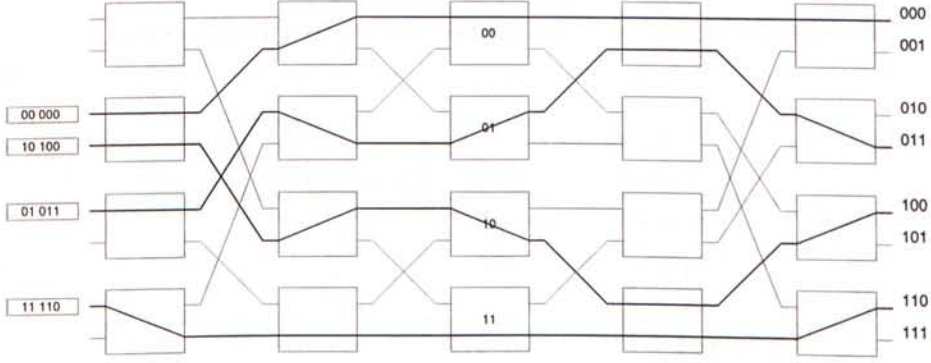


Figure 3.8: The routing of a 8×8 Benes Network at central stage

The network space complexity of the bufferless switch module implementation scheme depends on the size of input ports we use in the first stage modules. The total number of switch elements(SE) used in the quantized Clos network is

$$\begin{aligned}
 & N/n \times \text{no. of SE in each first stage module} \\
 & + 2n \times \text{no. of SE in each central stage module} \\
 & + N/n \times \text{no. of SE in each third stage module}
 \end{aligned}$$

Follow table 3.1, the total number of 2×2 switch elements used in an $N \times N$ quantized Clos network is

$$\frac{N}{2} \log_2 2n(2 + \log_2 n) + N + \frac{3N \log_2 n}{2} + N(2 \log_2 \left(\frac{N}{n}\right) - 1)$$

and the number of stages where cells need to pass are

$$(1 + \log_2 n)^2 + 2 \log_2 N$$

where n is the number of input ports in the first stage modules.

In addition to using multistage bufferless switch module to construct the Clos network, we can also use memory switch to construct. Since the cells are

Table 3.1: The complexity of different stage module in bufferless implementation scheme

	Dimension	Number of Modules	Number of SEs per module	Number of stages per module
1 st stage	$n \times 2n$	$\frac{N}{n}$	$\frac{n}{4} \log_2 n (1 + \log_2 n) + n + 2 \times \frac{n}{2} (\log_2 n)$	$\frac{\log_2 n}{2} (1 + \log_2 n) + 1 + \log_2 n$
2 nd stage	$\frac{N}{n} \times \frac{N}{n}$	$2n$	$\frac{N}{2n} (2 \log_2 \frac{N}{n} - 1)$	$2 \log_2 \frac{N}{n} - 1$
3 th stage	$2n \times n$	$\frac{N}{n}$	$\frac{n}{4} (1 + \log_2 n) (4 + \log_2 n) + \frac{n}{2} \log_2 n$	$\frac{(1 + \log_2 n)}{2} (2 + \log_2 n) + \log_2 n$

scheduled to transmit without blocking, no cell is needed to be buffered in the memory switch. The memory size of the switch is fixed and is dependent on the number of input ports and the cell size. The requirement of the memory size is

$$\text{no. of input ports} \times \text{cell size (plus extra header)}$$

Using memory switch to construct the Clos network can also improve the transmission time of the cells inside the network since cells do not have to pass through multistage switch elements. However, the internal speed of the memory switch must be $2n$ times of a multistage switch where n is the number of input ports.

3.2.3 Buffered Switch Modules Construction Scheme

The second implementation scheme is to replace the bufferless switch modules in the third stage by those with buffers. The advantage of this construction is that the third stage modules can be independently controlled without requiring getting any information from the central controller. When cells come in the third stage module from the central modules, the modules perform the contention resolution and buffer cells which have lost contention. The central controller only needs to schedule the cell transmission time in the first-stage and central

stage. This reduce the computation time complexity of the central controller at the expense of having extra buffer to prevent cell loss inside the network and an increase in the cell delay in the third stage. The buffered switch modules are not used in the first and central stage because the cell delay cannot be controlled. The cells from the same connection may have different routing paths in the network and each may have different queueing delay time according to each central stage module. Therefore, buffer is also needed at each output link and re-sequencing must be performed. The more buffered-switch module is used, the more buffer storage size is needed and the longer cell delay time it experiences. Consequently the QoS cannot be guaranteed.

The construction of the first two stage modules is the same as the bufferless implementation scheme. The buffered switch module in the third stage can be constructed by input-buffered Batcher-banyan network with 3-phase contention resolution scheme[21], output buffered Knockout switch[8], output buffered Sunshine switch[9] or shared buffer memory switch[18]. The overall cell throughput-delay performance and the buffer size requirement are depended on which buffered switch module we used. In addition, in order to prevent cell loss inside the module the group size of the Knockout switch and the Sunshine switch must be large enough. The sufficient group size K is $\min(2n, M)$ since there are $2n$ central modules which cells can come from. However, the cells with the same destination output port must less than or equal to M due to the link capacity so that the sufficient group size is $\min(2n, M)$. In general, the quantization level M is larger then $2n$ because when M is large, more services with different transmission rate can be supported and minimize the waste of quantization lost of bandwidth quantization. The internal switch speed, network space complexity

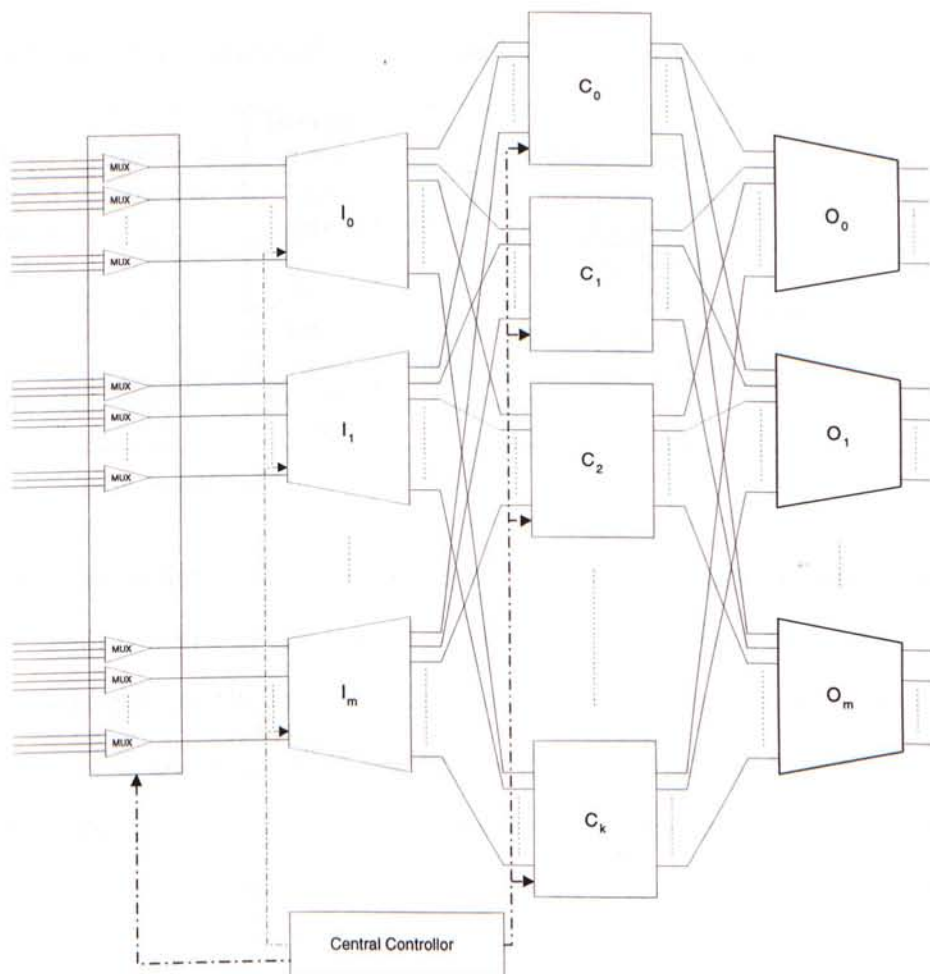


Figure 3.9: Buffered switch modules construction scheme

and the throughput performance of these switch are shown in table 3.2.

3.3 Complexity Comparison

The complexity comparison between the two proposed schemes can be considered in the computation time complexity due to the algorithm and the network space complexity due to the requirement of switching elements. In addition the buffer memory storage requirement to prevent cell loss is also a concern. In

Table 3.2: Comparison of different buffered switch modules used in stage 3

Module Type	Dimension	Internal Speed	Number of Switching Elements	Number of Stages per module
Input-buffered Batcher-banyan	$2n \times n$	Line rate	$\frac{n}{4}(1 + \log_2 n)(4 + \log_2 n) + \frac{n}{2} \log_2 n$	$\frac{(1+\log_2 n)}{2}(2 + \log_2 n) + \log_2 n$
Output-buffered Knockout Switch	$2n \times n$	Line rate	$n(2nK - \frac{K}{2}(K + 1))$	$2(n + 1)$
Output-buffered Sunshine Switch	$2n \times n$	Line rate	$\frac{n}{2}(1 + \log_2 n)(2 + \log_2 n) + 3n(1 + \log_2 n) + \frac{Kn}{2} \log_2 n - K + 1$	$\frac{(1+\log_2 n)}{2}(2 + \log_2 n) + 3 \log_2 n + 3$
Shared-buffered Memory Switch	$2n \times n$	$4n \times$ line rate	-	-

K: group size n: number of input ports

the first one bufferless switch module scheme, the computation time complexity is the main concern in the time slot assignment when each connection arrived. The complexity of the algorithm $tsa()$ which is used to schedule the time slot assignment is $O(\lambda M)$. After the time slot assignments, the routing path also needs to be found. The complexity of the routing assignment is $O(2n)$ because we search for the routing path within the $2n$ central modules. Therefore the overall complexity is $O(\lambda M + 2n)$.

In the second scheme, since we do not need to schedule the output link time slot, we can skip the $tsa()$ algorithm and just simply assign the time slot assignment in available slots in the input link of the first stage module. The complexity of the searching is $O(M)$ and the routing assignment complexity is the same as $O(2n)$. Therefore the overall complexity reduce to $O(M + 2n)$.

The network space complexity of the first scheme depends on the size of the first stage module. The network space complexity of the second scheme depends on which buffered switch we used. Tables 3.3 and 3.4 show the numbers of 2×2 switch elements used and the numbers of stages which the cells have to pass in

the network respectively in the bufferless switch module construction schemes with different N and n . Table 3.5 to table 3.8 show the numbers of 2×2 switch elements used in the buffered switch module construction schemes using Knockout switch module and Sunshine switch module with different N , n and quantization level M . Tables 3.9 and 3.10 show the numbers of stages which cells have to pass in the network when using the Knockout switch and Sunshine switch respectively in the buffered switch module construction scheme. Obviously, the complexity of the buffered switch module construction scheme is higher than the bufferless switch module construction scheme and the complexity of using Knockout switch is higher than using Sunshine switch. From the tables we can also observe that choosing a small n can reduce the number of switch elements to construct the network. In addition, the number of stages is also reduced when n is small so that the transmission time of cells inside the network can also be reduced.

Table 3.3: Switch elements required in bufferless switch module scheme

Number of 2×2 switch elements required						
N	n					
	256	128	64	32	16	8
32	-	-	-	-	736	592
64	-	-	-	1952	1600	1312
128	-	-	4992	4160	3456	2880
256	-	12416	10496	8832	7424	6272
512	30208	25856	22016	18688	15872	13568
1024	62464	53760	46080	39424	33792	29184

Table 3.4: Number of stages in bufferless switch module scheme

N	n					
	256	128	64	32	16	8
32	-	-	-	-	35	26
64	-	-	-	48	37	28
128	-	-	63	50	39	30
256	-	80	65	52	41	32
512	99	82	67	54	43	34
1024	101	84	69	56	45	36

Table 3.5: Switch elements required in buffered switch module scheme

Number of 2×2 switch elements required when using Sunshine switch module at stage 3 $M=100, K=\min(2n, M)$					
N	n				
	128	64	32	16	8
128	-	-	27332	13704	7440
256	-	94196	55176	27920	15392
512	222324	189416	43120	56864	31808
1024	446696	380880	224800	115776	65664

3.4 Delay Performance of The Two Implementation Schemes

3.4.1 Assumption

The delay performance we discuss here is the cell queueing delay in the Clos network. Although there is no buffer in the first implementation scheme, cells are needed to be buffered at the multiplexer until the network is available for their transmission. In the following discussion, we will ignore the transmission delay inside the same network configuration since every cell will experience same

Table 3.6: Switch elements required in buffered switch module scheme

Number of 2×2 switch elements required when using Knockout switch module at stage 3 $M=100, K=\min(2n, M)$					
N	n				
	128	64	32	16	8
128	-	-	260160	65408	17152
256	-	1489248	520832	131328	34816
512	16534480	3979520	1042688	263680	70656
1024	21070848	7961088	2087424	529408	143360

Table 3.7: Switch elements required in mixed switch module scheme

Number of 2×2 switch elements required when using Sunshine switch module at stage 3 $M=10, K=\min(2n, M)$					
N	n				
	128	64	32	16	8
128	-	-	10268	8248	6384
256	-	25436	21048	17008	13280
512	61404	51896	43120	35040	27584
1024	124856	105840	88288	72128	57216

transmission delay inside the same network configuration.

We assume that the cell arrival process is a periodic arrival process such that every connection will generate fixed number of cells in every M slots, where M is the quantization level, depending on their bandwidth requirement but the pattern of cells arrival may be different. As we schedule the time slot and allocate the bandwidth capacity to each connection, cells may not arrive at the time slot that we have scheduled for them. As shown in figure 3.10, cells may arrive at any time slot and have to wait for its transmission time slot. We use this process to model multirate circuit emulation services on ATM cell switching

Table 3.8: Switch elements required in mixed switch module scheme

Number of 2×2 switch elements required when using Knockout switch module at stage 3 $M=10, K=\min(2n, M)$					
N	n				
	128	64	32	16	8
128	-	-	76992	35840	15232
256	-	318848	154496	72192	30976
512	1295360	638720	310016	145408	62976
1024	2592768	1279488	622080	292864	128000

Table 3.9: Number of stages in mixed switch module scheme

Number of stages when using Sunshine switch module					
N	n				
	128	64	32	16	8
128	-	-	63	50	39
256	-	80	65	52	41
512	99	82	67	54	43
1024	101	84	69	56	45

network. Cells from the same connection will depart in sequence such that the early arrival cell has high priority to transmit when the time slot is scheduled for that connection. Therefore, cell may have to wait for next available time slot even it arrive at the right time of transmission. In addition, the connections we generated in this simulation model are randomly distributed over the N links and their bandwidth requirement ω is uniformly distributed. We simulate the cell level switch by given a set of connections existing in the network. We analyze the delay performance in cell level under different link loading. We generate fixed connection patterns in the network such that the link loading is constant over the simulation. Otherwise the connection arrival and departure will affect

Table 3.10: Number of stages in mixed switch module scheme

Number of stages when using Knockout switch module					
N	n				
	128	64	32	16	8
128	-	-	90	54	35
256	-	161	92	56	37
512	297	163	94	58	39
1024	299	165	96	60	41

the cell level link loading.

Cells are stamped when it enters the multiplexer. They will queue in the multiplexer and in the buffer at stage three until they can transmit at their scheduled time. Cells will be stamped when they come out of the network. Therefore the delay time is the difference between its departure stamped time and arrival stamped time.

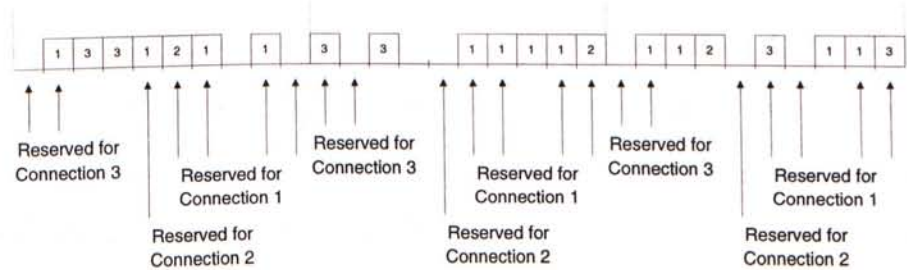


Figure 3.10: Example of cell arrival at different time unit

3.4.2 Simulation Result

We simulate the cell delay performance of the bufferless construction scheme with different network size, 256×256 and 64×64 , and different input module size, n . From the simulation result (figure 3.11 and 3.12), we see that the cell

delay performance is the same when the network size is fixed but different input module size. We can see that the cell delay performance of this scheme is mainly limited in the queuing in the multiplexer. No matter the complexity of the switching network is high or low, in order to the input size of first stage module, there will be nonblocking for the cell to transmit through the network. So that the maximum delay for the cell is $M - 1$ time slots and is the worst case delay time of a cell.

The network configuration does not affect the waiting time of cell in the multiplexer since the cell and the network are all scheduled. Therefore different n with same N have same mean cell delay performance. In addition, we observe from figure 3.11 and 3.12, that the mean cell delay is approximately linear proportion to the input link load in our simulation model since the cells periodically arrive and depart at fixed rate. The reason for this phenomenon is due to the correlation of the cells in heavy load. In heavy loading situation, connections may request a high bandwidth such that cells are correlated. Cells may bursty arrive at the multiplexer but not arrive at their scheduled time slot. Although some of them are matched with the transmission time slot, they still have to queue at the multiplexer because the early arrival cells have to transmit first. Therefore, when the loading is high, this situation will occur more frequently and the mean cell delay is approximately linear increasing with the loading.

In the simulation results of buffered construction scheme, we observe that the performance is the same when using the output-buffered modules with group size is equal to n or greater than n since the special case did not occur in the simulation. However, we still need to have group size of $\min(M, 2n)$ to guarantee the cell will not be loss inside the network. From figure 3.13 and figure 3.14,

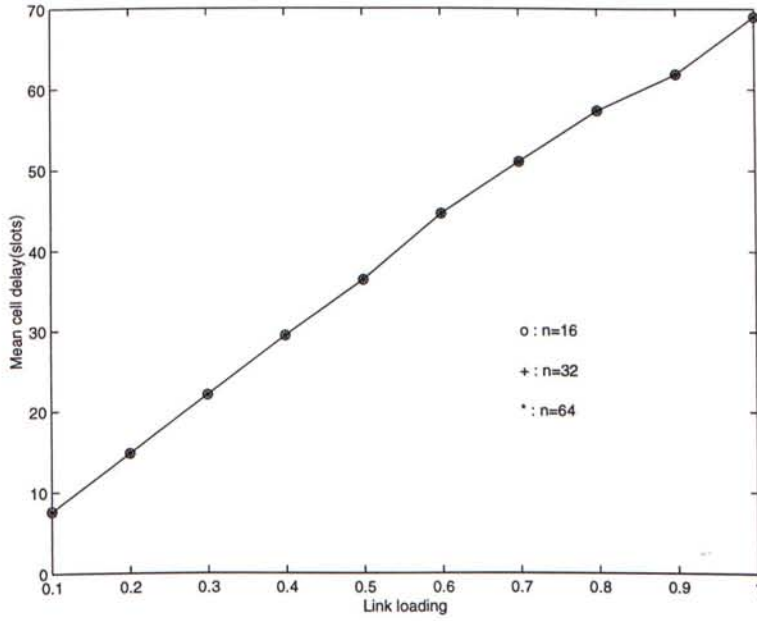


Figure 3.11: Mean cell delay performance of bufferless construction scheme with network size 256×256 , quantization level = 100

we can observe that the mean cell delay performance in small input size n is better than large n . In addition, the situation is more significant when using input-buffered modules (figure 3.15 and 3.16) with look-ahead windows scheme. The reason for this result is due to the contention problem of large size switch modules. Cells contend for same output frequently when the switch size is large. Therefore, the head-of-line blocking affect the throughput of the switch and the delay become longer. In output-buffered modules construction scheme, the larger the input switch size in first stage module, the larger input switch size in the third stage module because the input switch size is twice of n . The probability of cells going to same destination increases when the number of input increase. Therefore, the queuing time of cells queue in the output buffer of a large size switch is longer than a small size switch.

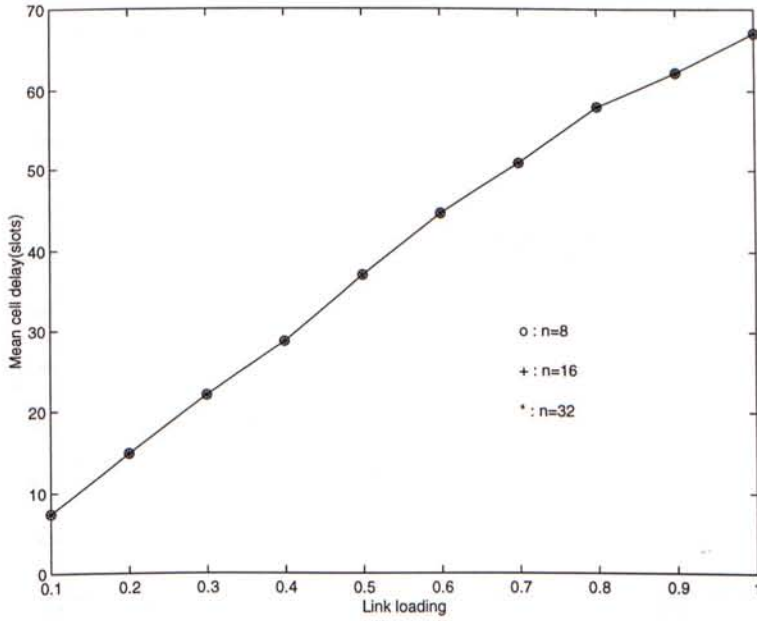


Figure 3.12: Mean cell delay performance of bufferless construction scheme with network size 64×64 , quantization level = 100

Comparison of using input-buffered module and output-buffered module which used in stage 3, the cell delay performance of using output buffered module is better than using input buffered module since the HOL blocking effect is significant when loading increase (figure 3.17). Although, look-ahead window scheme can improve the performance but the delay can not be bounded in heavy load such that the QoS of the connection can not be guaranteed. The mean cell delay in output-buffered module construction scheme is less than M time slots even in heavy loading and the maximum delay time is less than $2M$ times slots (figure 3.18).

In summary, we observe the following from the simulations performed:

1. The cell delay performance of the bufferless construction scheme is the best over the others.

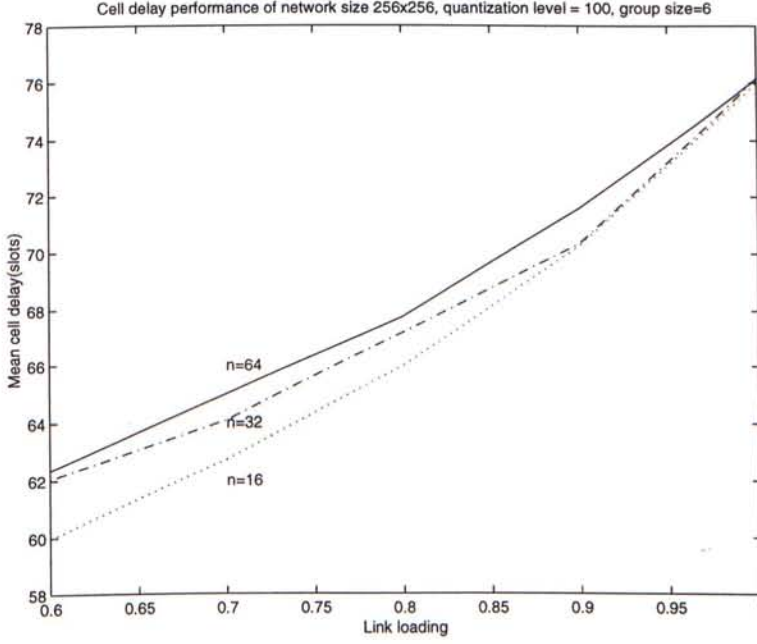


Figure 3.13: Mean cell delay performance of output-buffered module construction scheme with network size 256×256 , quantization level = 100, group size=6

2. Choosing a small input size of module can get better cell delay performance in buffered switch modules construction scheme.
3. In the buffered switch modules construction scheme, using output-buffered module results in better cell delay performance than using input-buffered module. In addition, the maximum cell delay time can be guaranteed when using output-buffered module even in heavy loading.

Regarding the computation time complexity, network space complexity and the delay performance, we can see that there are two main tradeoffs between bufferless switch module construction scheme and buffer switch module construction scheme. To minimize the computation time complexity, we choose the buffer switch module construction scheme. For best delay performance and low

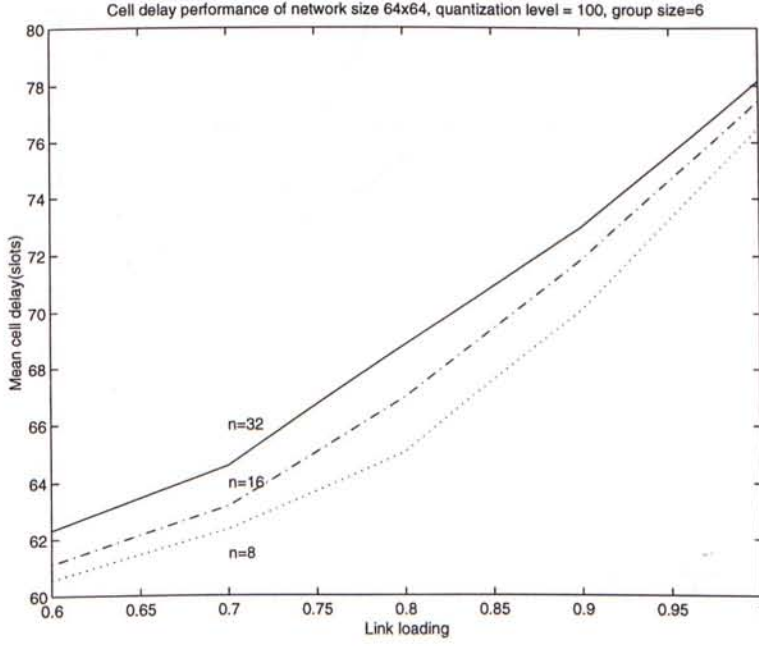


Figure 3.14: Mean cell delay performance of output-buffered module construction scheme with network size 64×64 , quantization level = 100, group size=6

complexity on building the switching network, we choose the bufferless switch module construction scheme. The network space complexity of the buffered switch module construction scheme also shows that using a small input size of the first stage module can reduce the number of switch elements used in the network and reduce the stage that the cells have to transmit inside the network. In addition, using a small input size of the first stage module can also get a better cell delay performance. However, if the input size n is too small, the switch module size in the central stage will become large and the routing in the central module will become complex. Therefore, we better to choose $n \simeq \sqrt{N}$ so that the size of the switch module over the three stage will not have a large difference. In stead of considering the complexity of the multistage switch module, the complexity of the memory switch module is also depending on the input

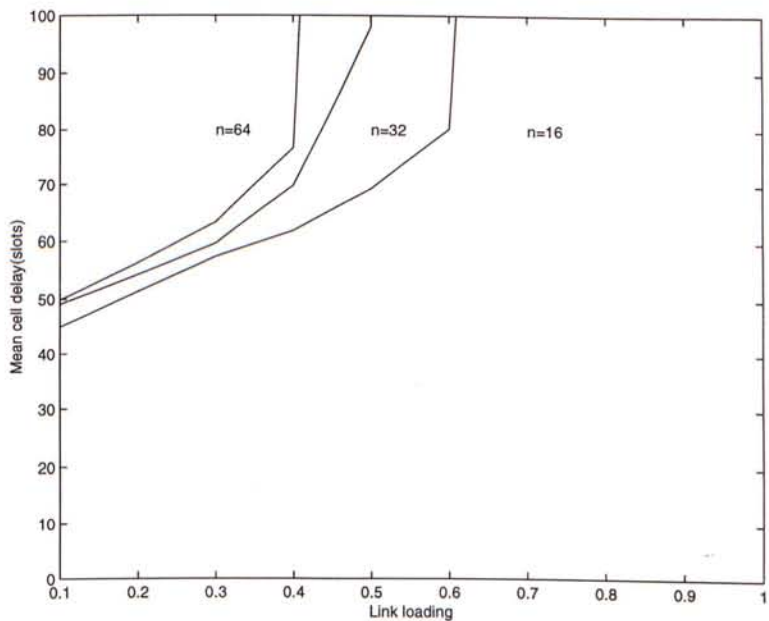


Figure 3.15: Mean cell delay performance of input-buffered module construction scheme with network size 256×256 , quantization level=100, look-ahead-window size=20

size since the core processing speed of the memory switch is limited. The larger the switch module size is, the slower the transmission speed of cells the switch module has.

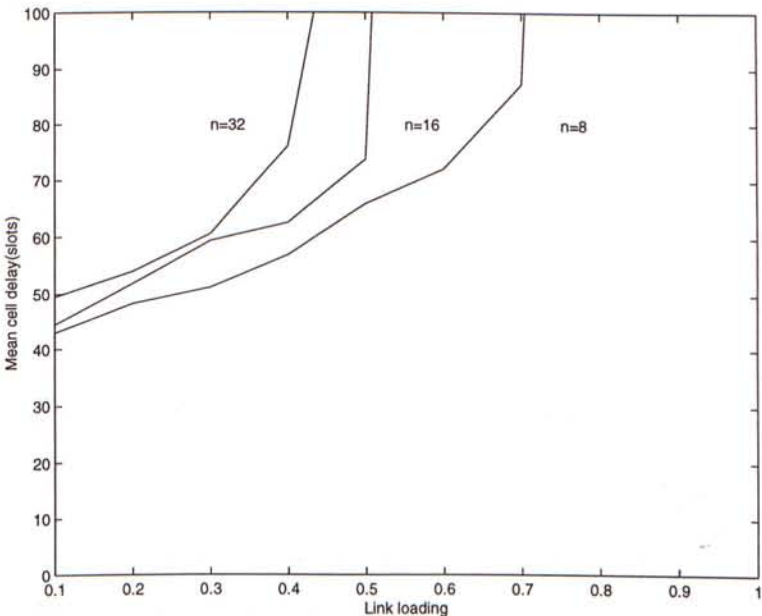


Figure 3.16: Mean cell delay performance of input-buffered module construction scheme with network size 64×64 , quantization level=100, look-ahead-window size=20

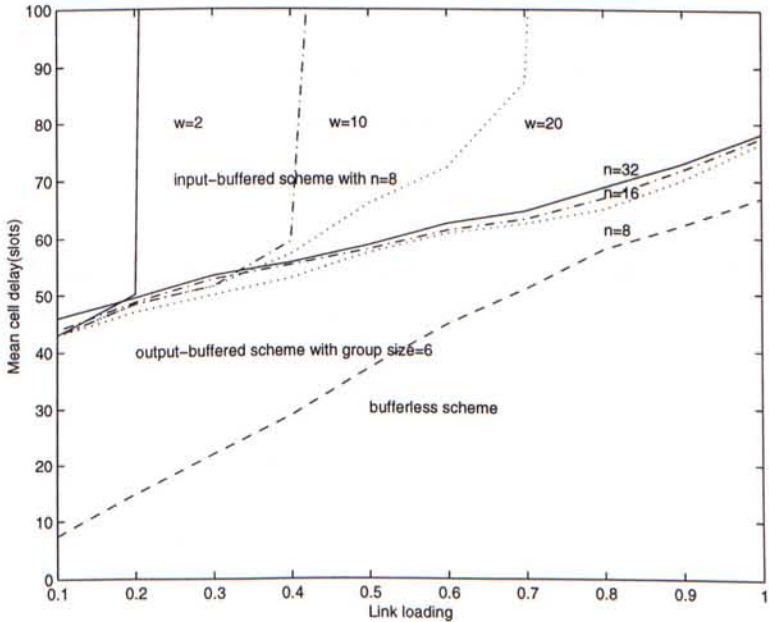


Figure 3.17: Comparison of the mean cell delay performance between three different construction scheme

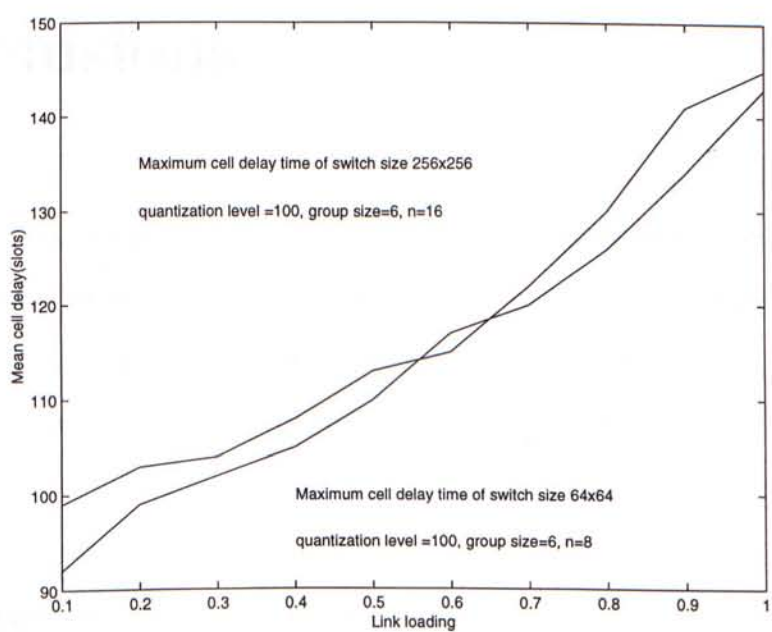


Figure 3.18: Maximum cell delay performance of using output-buffered module construction scheme

Chapter 4

Conclusions

In this thesis we performed an overall study of multirate circuit switching in quantized Clos network. In multirate circuit switching, a transmission link is shared by a number of connections with different transmission rates and each connection is guaranteed to be no data loss and fixed delay time. Clos network is a well studied switching network for the circuit switching in which the non-blocking conditions and routing algorithms have been derived by researchers. However, since multirate traffic are in cell format in ATM networks, the routing algorithms for using the Clos network as the switch network is not yet proposed. In addition, the nonblocking condition for the Clos network in multirate environment is not valid in cell level without any cell scheduling. This is the motivation of this thesis study.

We have proposed to use bandwidth quantization to convert the continuous scale bandwidth requirement to a finite discrete scale. With bandwidth quantization, we can make the calculation of routing and capacity allocation in multirate environment simpler and connection can be splitted such that the

bandwidth requirement can be shared by different paths inside the Clos network. This reduces the number of central modules required to make the Clos network nonblocking in multirate environment. In Chapter 2, we have stated the differences between the call level routing and the cell level routing in the quantized Clos network. The main difference is that the call level routing have not considered the simultaneous arrival of traffic and the cell contention in each stage modules. We have proposed to use a time slot assignment (TSA) scheduling algorithm which is based on the correspondence between the problem of computing an incremental TSA and the rearrangement problem in a 3-stage Clos network. We show that no rescheduling of existing connection time slot assignment is needed for any new connection time slot assignment when limiting the link utilization to 50%. Finally, the routing algorithm at cell level has been proposed which is modified from the routing algorithm at call level with TSA scheduling algorithm.

In Chapter 3, we have proposed two construction schemes for the quantized Clos network which using bufferless switch modules and buffered switch modules. We have investigated the cell delay performance of these two schemes by simulation and compared their computation time complexity and space complexity. The computation time complexity in the bufferless switch module construction scheme is high but the space complexity is low when compared with the buffered switch module construction scheme. From the simulation results, we conclude that using bufferless switch modules can have the best cell delay performance. Using output-buffered switch modules at the third stage in the buffered switch module construction scheme can have better cell delay performance than using input-buffered switch modules and the delay can also be bounded. In addition,

choosing a suitable n where $n \simeq \sqrt{N}$ so that the size of the switch module over the three stages will not have a large difference leads to a better cell delay performance in the buffered switch modules construction scheme.

Finally, some related directions for further research are outlined in the following. The TSA scheduling algorithm can be improved so that the TSA can be executed in parallel instead of sequentially scheduled. Another issue that is worthy of study is the rearrangement algorithm for the routing assignment and the time slot assignment of connections such that the number of central modules can be reduced to achieve non-blocking in the Clos network.

Bibliography

- [1] A.S. Acampora, *An Introduction to Broadband Networks*, Plenum Press, New York and London, 1994.
- [2] M. Schwartz, *Broadband integrated networks*, Prentice Hall, 1996.
- [3] C. Clos, "A Study of Non-Blocking Switching Networks," *Bell Sys. Tech. Jour.*, Mar. 1953, pp.406–424.
- [4] V.E. Benes, "On Rearrangeable Three-Stage Connecting Networks," *Bell Sys. Tech. Jour.*, Sept. 1962, pp.1481–1492.
- [5] D.C. Opferman and N.T. Tsao-Wu, "On a Class of Rearrangeable Switching Networks Part I: Control Algorithm," *Bell Sys. Tech. Jour.*, May-June 1971, pp.1579–1600.
- [6] R. Melen and J.S. Turner, "Nonblocking Multirate Networks," *SIAM Journal on Computing*, Vol. 18 Apr. 1989, pp.310–313.
- [7] K.Y. Eng, M.J. Karol and Y.S. Yeh, "A Growable Packet (ATM) Switch Architecture: Design Principles and Applications," *IEEE Trans. on Comm.*, Vol. 40, No.2, Feb. 1992, pp.423–430.
- [8] Y.S. Yeh, M.G. Hluchy and A.S. Acmpora, "The Knockout Switch: A Simple Architecture for High-performance Packet Switching," *IEEE JSAC*, Vol. 5, No.8, Oct. 1987, pp.1274–1283.
- [9] JN. Giacomelli, J.J Hickey, W.S. Marcus, W.D. Sincoskie, and M. Littewood, "Sunshine: A High-Performance Self-routing Broadband Packet Switch Architecture," *IEEE JSAC*, Vol. 8, No. 9, Oct. 1991, pp.1289–1298.
- [10] S. Chung and K.W. Ross, "On Nonblocking Multirate Interconnection Networks," *SIAM Journal on Computing*, Vol. 20 N0.4 1991, pp.726–736.

- [24] T.T. Lee, "A Modular Architecture for Very Large Packet Switches," *IEEE Trans. on Comm.*, Vol.38, No.7, Jul. 1990, pp.1097–1106.
- [25] T.T. Lee and S.Y. Liew, "Parallel Algorithm for Benes Networks," *Proc. INFOCOM '96*, 1996, pp.279–286.
- [26] T.T. Lee and C.H. Lam, "Path Switching - A Quasi-Static Routing Scheme for Large-Scale ATM Packet Switches," *to be appeared in IEEE Journal on Selected Area in Communications*
- [27] W.H. Kwok, "Routing Algorithm for Multirate Circuit Switching in Quantized Clos Network," *Master Thesis*, 1997, Division of Information Engineering, Chinese University of Hong Kong.
- [28] A. Varma and S. Chalasani, "An Incremental Algorithm for TDM Switching Assignments in Satellite and Terrestrial Networks," *IEEE Journal on Selected Area in Communications*, Vol.10, No.2, Feb. 1992, pp.364–376.
- [29] C.K. Kim, "Blocking Probability of Heterogeneous Traffic in a Multirate Multicast Switch," *IEEE Journal on Selected Area in communications*, Vol. 14, No.2, Feb. 1996, pp.374–385.
- [30] D. Mitra, J.A. Morrison and K.G. Ramakrishnan, "ATM Network Design and Optimization: A Multirate Loss Network Framework," *IEEE/ACM Trans. on Networking*, Vol. 4, No. 4, Aug. 1996, pp.531–543.
- [31] H.Kondoh, et.al., "A 622-Mb/s 8×8 ATM Switch Chip Set with Shared Multi-buffer Architecture," *IEEE Journal of Solid-State circuits*, Vol.28, No.7, July 1993, pp. 808–815.

- [11] K.W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer, 1995.
- [12] C. Partidge, *Gigabit networking*, Addison-Wesley, 1993.
- [13] C.T. Lea and A. Alyatama, "Bandwidth Quantization and Staes Reduction in the Broadband ISDN," *IEEE/ACM Trans. on Networking*, Vol.3, No.3 June 1995, pp.352–360.
- [14] S.C. Liew, "Performance of Various Input-buffered and Output-buffered ATM Switch Design Principles under Bursty Traffic: Simulation Study" *IEEE Trans. on Comm.*, Vol.42, No. 2/3/4, Feb/Mar/apr 1994, pp.1371–1379.
- [15] S.C. Liew and T.T. Lee, *Principles of Broadband Switching and Networking: Draft 3*, Lecture Notes of the course *Telecommunication Network*, Chinese University of Hong Kong.
- [16] M.J. Karol, M.G. Hluchyj and S.P. Morgan, "Input Versus Output Queueing on s Space-Division Packet Switch," *IEEE Trans. on Comm.*, Vol. 35, No.12, Dec 1987, pp.1347–1356.
- [17] M.G. Hluchyj and M.J. Karol, "Queueing in High-Performance Packet Switching," *IEEE JSAC*, Vol.6, No.9, Dec. 1988, pp.1587–1597.
- [18] T. Kozaki, N. Endo, Y. Sakurai, O. Matsubara, M. Mizukami, and .I. Asano, "32 × 32 Shared Buffer Type ATM Switch VLSI's for B-ISDN's," *IEEE JSAC*, Vol. 8, No.9, Oct. 1991, pp.1239–1247.
- [19] M. Devault, J. Cochenec and M. Servel, "The Prelude ATD Experiment: Assignments and Future Prospects," *IEEE JSAC*, Vol.6, No.9 Dec. 1988, pp. 1528–1537.
- [20] J.Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers.
- [21] Y.N.J. Hui and E. Arthurs, "A Broadband Packet Switch for Integrated Transport," *IEEE JSAC*, Vol.6, No.9 Dec.1988, pp. 1264–1273.
- [22] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN, Second Edition*, Ellis Horwood, 1993.
- [23] T.T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE JSAC*, Vol.6, No.9, Dec. 1988, pp.1455–1467.

CUHK Libraries



003704287