A Quasi-Static Routing Scheme for Cross-Connected Storage Area Network

YANG Qin

A Thesis

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Philosophy

in

Information Engineering

July 2001

Acknowledgement

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

First of all, I would like to express my deepest gratitude towards my supervisor Prof Tony Lee for his continuous support, for his invaluable advice, and most importantly, for being a true teacher to me. I am grateful to him. Many thanks to Dr. M.C.Chan and Dr.S.Y.Liew for their valuable advice and sincere encouragement and all directions they provided.

I would also like to thank Mr.H.N.Lung, Mr.C.Siu for their continuous discussions with me and their various kinds of help. They are very kind and being good companion to me.

At last, I am grateful to my family, my mother and my elder brother and my niece, for morally supporting me and their understanding and encouragement during my two-year study.

Abstract

Recently, a highly scalable technology Fibre Channel was designed for supporting various size of Storage Area Network (SAN). Usually Fibre Channel switch has 16 I/O ports. The most promising interconnecting of Fibre Channel switches is a cross-connected configuration. The cross-connected SAN can be considered as a Clos network. Although this configuration provides the ultimate high bandwidth solution, the routing in it is yet a challenging issue to be solved. The existing routing scheme in cross-connected SAN is circuit switching, which is computational expensive and wastes a lot of bandwidth.

A novel quasi-static switching concept called Path Switching has been proposed recently for large-scale packet switching systems. It is a compromise of the static and the dynamic routing scheme in Clos network. Conceptually, there is virtual path connecting each input module and output module in a Clos network. The scheduling is based on providing QOS guarantee in all the virtual paths. The scheduling of Path Switching consists of two steps : Capacity Assignment and Route Assignment.

In this thesis, we propose to implement Path Switching in the cross-connected SAN. We will develop a complete set of traffic control strategies for the path-switched SAN to support various Fibre Channel traffic classes, the new scheme would be more efficient than the old circuit switching scheme and we give a theoretical upper bound of the blocking probability in Path Switching, which is testified by the simulation. Also, we introduce an effective way to produce the capacity matrix in Path Switching,

which reduces the computational complexity. Moreover, the choice of repetition rate in Path Switching is also discussed, it is a tradeoff between the memory requirement and the number wasted capacity. The result of this thesis is useful for systematic deployment of large-scale SAN.

摘要

近几年来，随着数据处理量的增加，存储局域网的运用范围越来越广泛，而光通道技术已成为存储局域网的一个热门话题。普通的光通道交换机拥有 16 个端口。 光通道交机的互联方式有多种，其中交叉互联最引人注目。交叉互联的拓扑结构可被认为同 Clos 网络想等同。目前，电路交换是交叉互联存储局域网的路由选择方式，但效率不高并且计算开销过大。

最近，基于Clos网络的虚通道交换非常受到关注，它假设在Clos网络的任何一个输入模块同任何一个输出模块间存在一条虚通道。虚通道交换采用准静态路由选择策略。虚通道交换机制包括两个部份：容量分配同路由分配。

在本篇毕业论文中，我们将把虚通道交换运用到交叉互联存储局域网中，与电路交换相比，效率将得到提高。我们将推导出虚通道交换中阻塞概率并用模拟实验证明。在虚通道交换的容量分配中，我们将提出一个新的算法已减少算法的复杂度。最后，虚通道交换的周期的长短也获得讨论，它同内存的多少及浪费的容量有关。

Contents

List of Figures

Chapter 1

Introduction

The Storage Area Network (SAN) is an emerging data communication platform that interconnects large amount of shared resources dispersed over a large area at Gigabaud speeds. As more and more data and transactions go online, early SAN is being challenged to support more and more device. Recently, a highly scalable technology called *Fibre Channel* was designed for supporting various size of SAN [1][2]. A challenging issue in the successful deployment of Fibre Channel SAN is to interconnect large amount of shared resources located over a large area. By combining LAN network models with the core building blocks of server performance and mass storage capacity, Fibre Channel eliminates the bandwidth bottlenecks and scalability limitations imposed by previous small computer system interface (SCSI) bus-based architecture. The simplest and least expensive implementation of Fibre Channel is a point-to-point connection between a server and a storage device [3]. While this implementation delivers speed, it does not offer scalability. For increased scalability, a number of Fibre Channel devices can be connected via fiber or copper cable on a shared-bandwidth arbitrated loop [4][5]. An arbitrated loop configuration increases the number of storage devices to which a server can connect and adds a hub for

arbitration. However, only one originator and one end-point can communicate at the same time with the maximum bandwidth. Therefore, it is inappropriate for fast data access. In large environments, a switched SAN may be the answer to achieve both speed and scalability. A switched SAN provides the flexibility of a fabric with full bandwidth to all nodes on the network simultaneously.

For large Fibre Channel SAN (more than 16 switches), switches should be connected and a SAN can be built to meet application requirements. Because multi-switch fabrics provide in-order delivery of frames through any number of links, several interconnection configurations are possible. These configurations can be classified into three categories, namely cascade, meshed and cross-connected. Cascade architectures are the least-expensive options and are similar to traditional network structures in which switches are daisy chained. It is most appropriate for low aggregate bandwidth requirements or for interconnecting just a few devices. Beyond these limited applications, cascaded architectures do not scale well. Moreover, since bandwidth and latency vary unpredictably, depending on where messages enter and exit the fabric, which means close attention must be put on the changing traffic patterns while designing and redesigning their system. For projects requiring a more predictable environment and flexible environments where data paths may change dynamically, meshed architectures offer a cost-effective entry to high-performance multi-switch topologies [6]. Because each switch is directly connected to every other switch in the fabric, the hop count remains low even as the fabric scales, which minimizes

bandwidth loss and the effects of compound latency. However, the high number of required inter-switch connections makes it impossible to add more I/O ports. The only way to grow a meshed configuration is to add larger switches. Therefore, for the sake of open-ended scalability and robustness, a cross-connected configuration is a better solution [7]. In fact, the topology of the whole cross-connect network can be treated as a Clos network except some minor difference.

In [8], a quasi-static routing scheme, called *path switching*, is proposed for building large scale packet switching systems. It is a compromise of the static and the dynamic routing scheme. The routing of path switching is based on the concept of *virtual path* within the Clos network [9]. We consider that there is a virtual path between an input module and an output module, which comprises all virtual circuits interconnecting any incoming port and any outgoing port on this pair of modules. The scheduling of path switching consists of two steps, the capacity assignment and the route assignment. The capacity assignment, which is also called capacity allocation, is to find the capacity $C_{xy} \geq \lambda_{xy}$ for each virtual path P_{xy} between input module I_x and output module O_y, where λ_{xy} denotes the aggregated constant bandwidth requirement of the virtual path P_{xy} in the unit of packets per time slot. The next step is to convert the capacity matrix $[C_{xy}]$ into a finite number f of regular bipartite multigraphs based on the time-space interleaving principle. By considering each input and output module as a node, the connection pattern in the middle stage of the Clos network can be represented by edge-coloring of a

bipartite multigraph. An edge-coloring of a bipartite multigraph is to assign *m* distinct colors to *m* edges of each node so that no two adjacent edges have the same color, where *m* is the number of central modules. Each color here corresponds to a central module, and the color assigned to an edge from input module I_x to output module O_y represents a path between them through the corresponding central module. If the switch is operated repeatedly in a finite number of timeslots *f* according to such a set of connection patterns, the capacity requirement on each virtual path can be satisfied in the long run, and the computation of route assignment on the fly can be avoided. In this thesis, we investigate the performance of path switching routing scheme in the SAN that has a cross-connected configuration.

Fibre Channel protocol supports three classes of service, namely dedicated connection (Class 1), multiplex (Class 2), and datagram (Class 3). Class 1 service is based on a dedicated connection between the communicating I/O ports, thus guaranteed bandwidth. While, Class 2/3 service is based on connectionless operation by multiplexed routing of frames between multiple I/O switches. To support such traffic classes, we will develop a measurement based algorithm for performing capacity assignment for each virtual path in path switched cross-connected configuration. This algorithm must be practical enough for real-time implementation.

To simplify the switch control and distribute it to different switches in path switched cross-connected configuration, we divide the traffic control into three levels, namely the switch level, the path level and the frame level. The switch level can be

viewed as the overall switching resources in the path switch. It is defined as the number of frames can be routed from input switch to output switch in one time slot, which is fixed and determined by the number of cross-connected switches. In this thesis, we show that the choice of the number of cross-connected switches is a tradeoff between the blocking probability of Class 1 traffic and the number of I/O devices supported.

The path level control deals mainly with virtual path capacity assignment, route assignment and reconfiguration at the cross-connected stage. The capacity of a virtual path reflects the frame transmission ability of the virtual path and can be adjusted by varying the connection patterns of the cross-connected switches. Recall that in the path-switched cross-connected fabric, the connection pattern at the cross-connected stage is deterministic and repeats in a fixed number *f* of time slots. Each cross-connected switch provides a path between a pair of I/O switches in each time slot. A frame that requests to depart an I/O switch must content a connection pattern of the central cross-connected switch with the same destination. The collection of the tokens at I/O switch S_x with destined I/O switch S_y divided by time can be interpreted as the capacity C_{xy} of the virtual path P_{xy} interconnecting them. For stability of the whole network, we should find the capacity $C_{xy} \geq \lambda_{xy}$ for each virtual path P_{xy} between I/O switch S_x and I/O switch S_y, where λ_{xy} denotes the aggregated admitted Class 1 traffic bandwidth requirement of the virtual path P_{xy} in the unit of

frames per time slot, and the left capacity $(C_{xy}\text{-}\lambda_{xy})$ will be distributed to the Class 2/3 traffic. Instead of the traditional modified simplex method solving objective function, a *measurement based algorithm* of capacity assignment is proposed to satisfy the objective function subject to $\sum_x C_{xy} = \sum_y C_{xy} = m$ (Row sum and Column sum), where m is the number of the central cross-connected switches. In measurement-based algorithm, firstly we form an intermediate capacity matrix easily in which the row sum condition $\sum_x C_{xy} = m$ is fulfilled. Secondly, we use this new algorithm to adjust former matrix to get the final capacity matrix, in which the row sum and column sum condition are both fulfilled $\sum_x C_{xy} = \sum_y C_{xy} = m$. Comparing with the traditional modified Simplex methods, the computation complexity of measurement based algorithm is reduced from an exponentially growing number to a polynomial growing number. Moreover, the measurement based algorithm is much simpler than the traditional modified simplex methods. In this thesis, we investigate the performance of measurement-based algorithm. We also develop a mathematical model to study the call blocking performance of Class 1 traffic in the Clos network by using M/D/1 model.

As mentioned previously, it is necessary to convert the capacity assignment into the connection pattern at each cross-connected switch. In an early study, we have shown that the more uniform the token distribution over time slots the smaller the delay jitter encountered by each frame [10]. Therefore, for better delay performance, we demand a route assignment

scheme to achieve uniform token distributions. In [11], Liew has proposed a route assignment algorithm for implementing stop-and-go queuing strategy in a path-switched Clos network. This algorithm can assure the distribution of tokens for each virtual path as uniformly as possible. In this thesis, we use this algorithm to convert the capacity matrix to the connection pattern at each cross-connected switch. Attention is put on the practicality and feasibility in the implementation of the algorithm.

1.1 Thesis Overview

This thesis is organized as follows:

- Chapter 2 gives an overview of Storage Area Network (SAN) and Fibre Channel, the new protocol proposed in SAN. It also presents the services type supported in Fibre Channel protocol, including Class 1 traffic, Class 2 traffic, Class 3 traffic. Three implementations of Fibre Channel are introduced, namely point-to-point, arbitrated loop and switched. We also study several topologies of SAN, cascaded, meshed, cross-connected. Moreover, we find that the architecture of the Cross-Connected SAN can be treated as Clos network. The existing routing scheme, circuit switching, in cross-connected SAN is also discussed.
- Chapter 3 presents a brief introduction about path switching. The scheduling scheme of path switching consists of two steps: capacity assignment and route

assignment. The computation complexity of capacity assignment is also studied, which shows that the computation complexity of modified simplex method is exponential.

- Path switching in cross-connected SAN is introduced in Chapter 4. The detail implementation is studied, including the time delay of Class 2/3 traffic, the blocking probability of Class 1 traffic, and total performance of mixed traffic. The simulation result is also presented. Beyond that, a new algorithm, measurement based algorithm, is proposed to reduce the computation complexity of capacity assignment into polynomial. The repetition rate *f* of path switching is also discussed. We find that the repetition rate *f* is a tradeoff between the number of wasted token and the memory requirement of the switch.
- The findings of this thesis are concluded in Chapter 5.

Chapter 2

Storage Area Network (SAN)

Because of the rapid growth in online transactions, more and more organizations, such as banks and companies, are storing an exploding data, resulting in the Storage Area Network (SAN) is widely implemented.

Traditionally, the storage devices are connected to the storage server with the channel protocol (Fig 2.1), namely SCSI (small computer standard interface), IPI (Intelligent Peripheral Interface), and HIPPI (High Performance Peripheral Interface). Since these channel protocols provide fast transmission speed by using parallel lines, the distances between the storage devices and servers can not be too long, usually 20-25 meters. These channel protocols can be considered as hardware intensive with low overhead. All storage servers are attached to LAN, WAN, or MAN through network protocols, namely IP (Internet Protocol), IPX (Internetwork Packet Exchange). With this approach (Fig 2.2), those storage servers do the conversion between the channel protocol and network protocol, acting as the conduit for data transfer, backup data and so on. Comparing with the channel protocol, the network protocol can be considered as software intensive with high overhead. Thus, there is a tradeoff between these two protocols in the traditional SAN: the channel protocol is fast but difficult to be expanded;

the network protocol is easy to be expanded but relatively slow. In addition, the performance of the storage server is the bottleneck of the whole SAN in most cases.

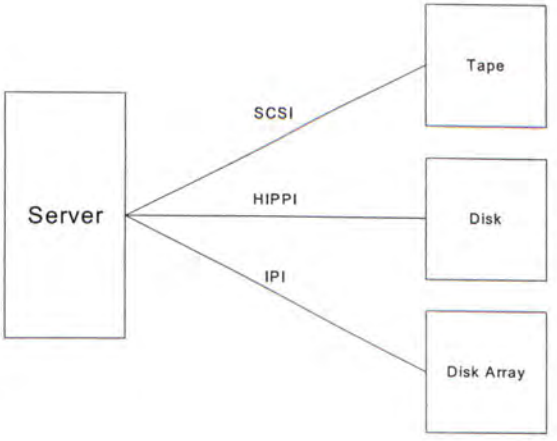

Figure 2.1 Channel Protocol

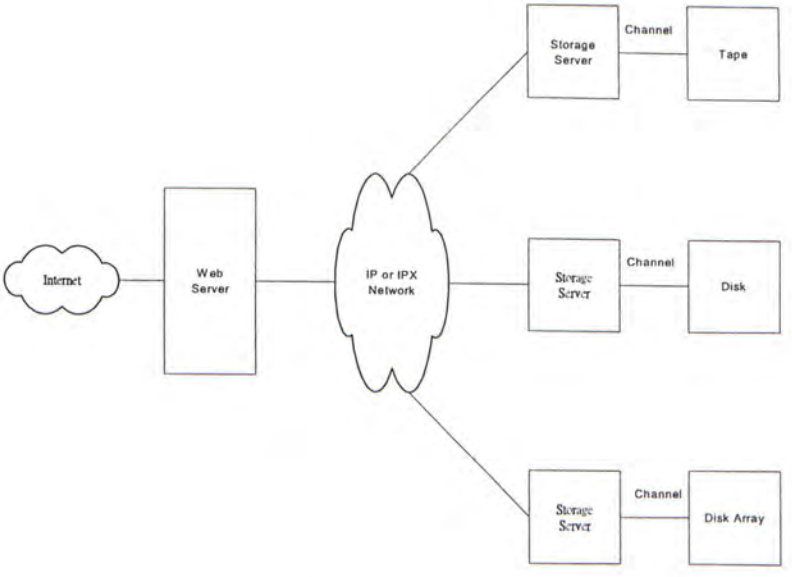

Figure 2.2 Traditional SAN

2.1 Fibre Channel Protocol

Recently Fibre Channel (FC) has been proposed [1][2] to allow SAN to transmit data at higher speed over greater distance, which should accelerate the adoption of SAN. In essence, FC allows all devices ,including disk, disk array, tape and server, to connect directly (Fig 2.3), which means all devices in FC SAN have equal status.

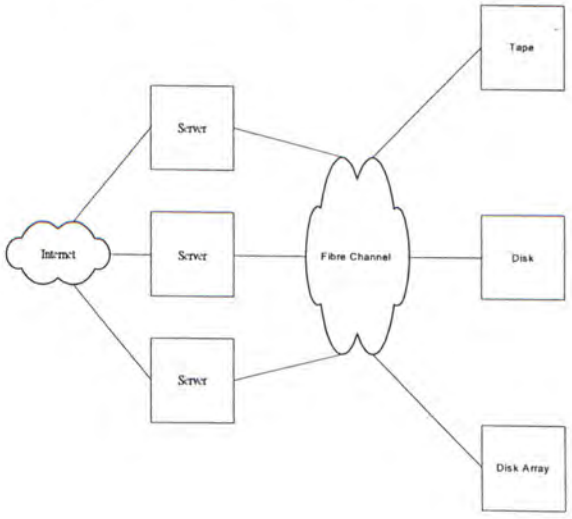

Figure 2.3 Fibre Channel SAN

The intention of FC is to develop practical, inexpensive, yet expandable means of quickly transferring data between workstations, mainframes, supercomputers, storage devices (including tape, disk array, disk). FC is the general name of an integrated set of protocol stacks being developed by American National Standards Institute (ANSI), which is shown in Fig 2.4.

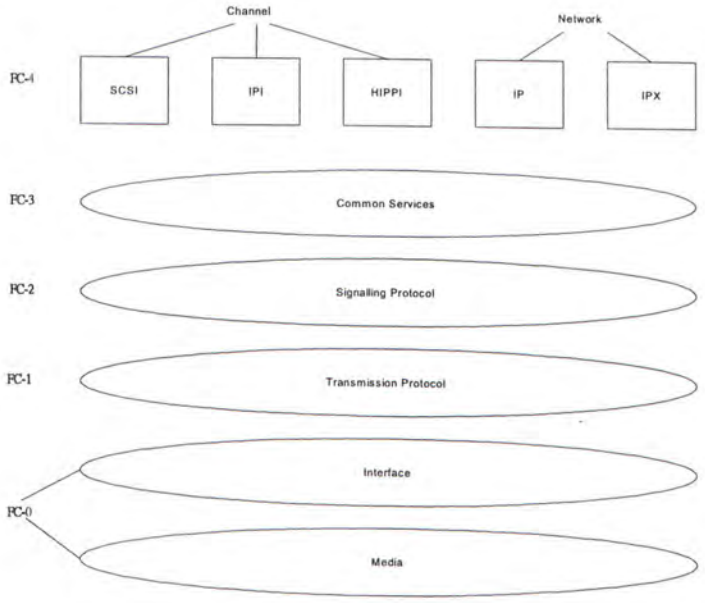

Figure 2.4 Fibre Channel Protocol

The lowest level of FC defines the physical link. The standard is rather rich here, enabling speeds of 265Mbit/s, 531Mbit/s, 1.06Gbit/s. As the transmission media, twisted pair, coaxial cable and optical fibre can be used depending upon the distance to be linked. While using the single mode fibre in FC, the maximum bit-distance product can be 10Gb.km and be expanded to 30Gb.km with an optical amplifier [1].

The interaction between two nodes in the Fibre Channel consists of a hierarchy of data units, the lowest of which is called *frame* [15][16][17]. The frame is the smallest indivisible unit of information transfer across the FC. In the Version 1.6, each frame consists of a 4-byte start-of-frame sequence, a 16-byte frame header, a variable-size data field with a maximum

length of 2112 bytes, 4 bytes of a cyclic redundancy sequence. Thus, the maximum length of a frame is 2140 bytes. The individual bytes are encoded by the IBM 8-bit/10-bit transmission code. The highest data-unit in the Fibre Channel that is of interest to us is *connection* [15][16][17]. Connection is the level at which two nodes establish communication while transferring large block data.

In Fibre Channel, the frame delimiters are used to specify the Class of Service required and are also used to make and break connections and to assist in the routing of frames. Three Classes of Service have been specified: connection oriented, connectionless with ACKs, connectionless datagram. Their principles are outlined below:

Class 1

Dedicated connection

End-to-end acknowledgement of frame delivery

Guaranteed sequential delivery

Guaranteed bandwidth

Class 2

Connectionless service

End-to-end acknowledgement of frame delivery

Class 3

Connectionless service

In summary, there are two kinds traffic in SAN: connection oriented (Class 1), connectionless (Class 2/3). In order to simplify the following analyze, in this thesis the Class 1 traffic request is a block of data with fixed length of 1500 frames [8],

which is roughly corresponding to one screenful of graphics data at 1024X1024 resolution with 24 bit per pixel (Fig 2.5). And the Class 2/3 traffic request has only a single Fibre Channel frame of 2140 bytes (Fig 2.6).

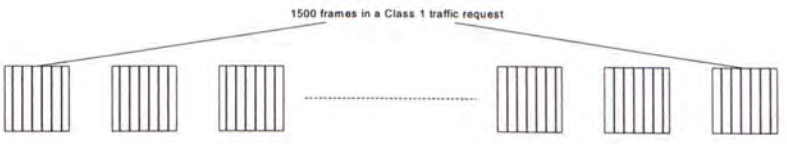

Figure 2.5 Class 1 Traffic

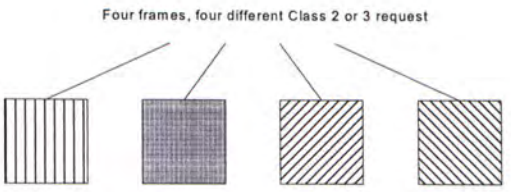

Figure 2.6 Class 2/3 Traffic

Fibre Channel SAN uses three topologies. Because point-to-point topology links only two nodes (Fig 2.7), it is not used so significant as the Fibre Channel Arbitrated Loop (FC-AL) (Fig 2.8) and Switched Fibre Channel (Fig 2.9).

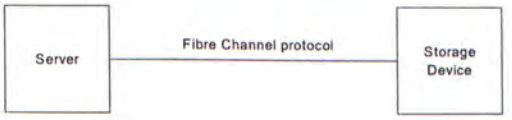

Figure 2.7 Point-to-Point Fibre Channel

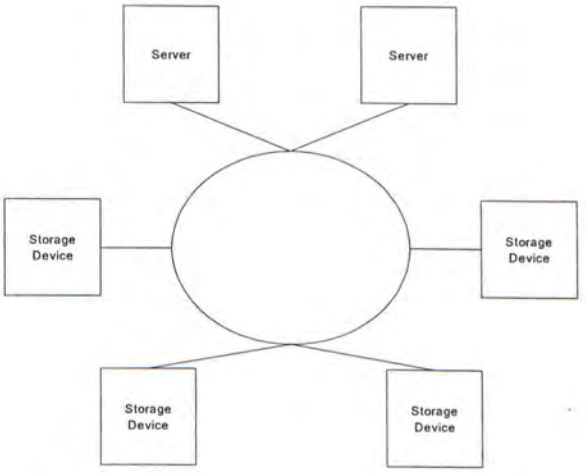

Figure 2.8 Arbitrated Loop Fibre Channel

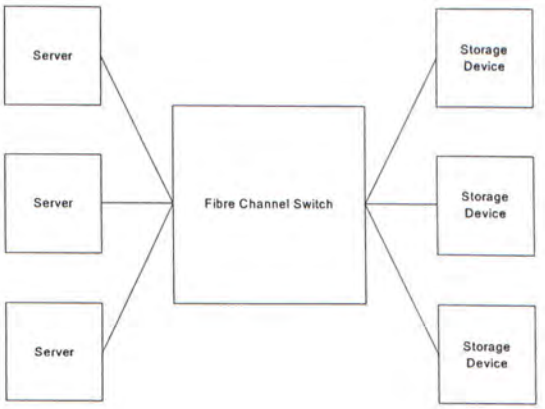

Figure 2.9 Switched Fibre Channel

Fibre Channel Arbitrated Loop: All devices share a hub, however, only one device on the loop can transfer data at one time. FC-AL is less expensive because of using hub. But as a shared technology, the FC-AL can not fully utilize the available throughput provided by Fibre Channel. Thus, it is best for the

projects where higher speed is not the highest priority but lower cost is.

Switched Fibre Channel: The switched topology is the best while high throughput is required. Switches can connect each port of device to the network directly, providing Fibre Channel's full duplex throughput in each transmission. Since its connections are direct, not shared, switched Fibre Channel is faster than FC-AL. This approach, however, is more costly and complex to operate and manage. Switched Fibre Channel would be best suited for users who have high-speed acquisition and real time applications. This thesis is focused on Switched Fibre Channel SAN.

2.2 Switched Fibre Channel SAN

Switched Fibre Channel SAN is designed to scale in capacity and performance, which will be necessary to cope with the future demands. Switched Fibre Channel SAN accommodates more devices than the traditional SAN adopting channel-network architecture and hence is more scalable. Usually a normal Fibre Channel Switch (FCS) has 16 ports [7][18], using crossbar architecture. If there are less than 16 devices in SAN, one FCS is enough. However, if there are more than 16 devices in SAN, the FCS should be connected to meet the requirement in larger environments. These topologies of the connections can be classified into three categories, namely cascaded, meshed, cross-connected.

2.2.1 Cascaded Topology

Cascaded topology (Fig 2.10) is the least expensive option and is similar to traditional network structures in which all hubs or switches are daisy chained. Like any topology that shares resources among devices, cascading is most appropriate when aggregate bandwidth requirements are low or when just a few devices need to be interconnected (e.g. in work groups, low-volume file sharing applications). Beyond these limited applications, cascaded architecture doesn't scale well. Adding switches significantly reduce the total performance due to compounded latency caused by multiple switch to switch hops and to the limited shared inter-link ports. Moreover, bandwidth and latency vary unpredictably, depending on where the traffic enters and exits network, which means network planners should pay close attention to changing traffic pattern while designing and redesigning their system.

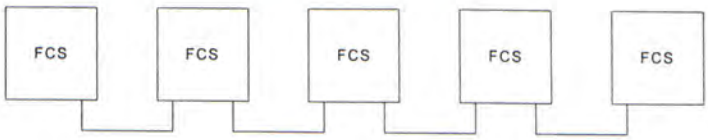

Figure 2.10 Cascaded Topology

2.2 Meshed Topology

For applications requiring a more predictable environment,

such as video servers, high volume, frequently accessed database, meshed topology (Fig 2.11) offers a cost effective entry to high performance SAN. Since each FCS is directly connected to every other FCS in the SAN, the hop count remains low even the SAN scales, minimizing the bandwidth loss and the effects of compounded latency. Because there is no single point failure in the SAN, this configuration is extremely resilient and is an excellent solution for data backup applications. Eventually, however, the scalability of meshed topology is limited to the number of ports in each FCS. As each FCS has 16 ports, the SAN of meshed topology could be no larger than 16 FCS. Obviously, one way to grow a meshed configuration is to add larger switches, but the meshed topology is not a better solution for applications with open-ended scalability and robustness.

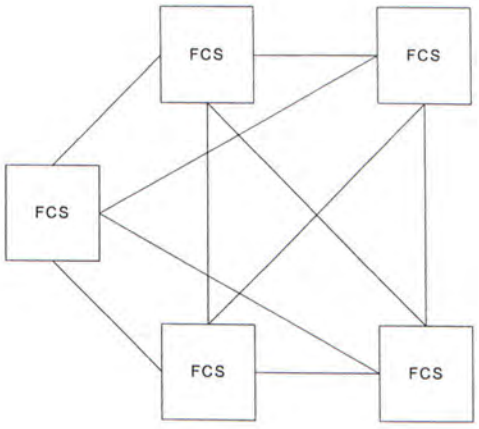

Figure 2.11 Meshed Topology

2.3 Cross-Connected Topology

Cross-connected topology (Fig 2.12), each FCS with a direct line to each cross-connected switch, offers the highest aggregate bandwidth by allowing network designers to control the ratio of I/O ports to the cross-connected ports. Although this architecture involves extra hardware cost, users can create as many redundant paths to achieve performance improvements.

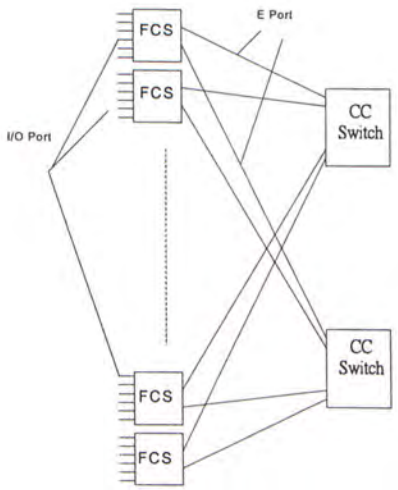

Figure 2.12 Cross-Connected Topology

A cross-connected SAN makes it possible to design systems with no single point failure, since each FCS can be linked to multiple cross-connected switches. In the unlikely event of total switch outage, redundant data paths keep the network functioning at normal or near normal capacity. Therefore, the cross-connected topology can provide the best return on investment for a wide range of large and midsize application

with the economic impact of scalability and resiliency.

Cross-connected SAN can be expanded in three ways: adding more E ports, adding more FCS, adding more CC switches, and they can be applied individually or in combination, depending on cost and performance. Suppose the total number of switches is fixed, there is a tradeoff between the total throughput and the number of I/O devices supported. The more CC switches in the SAN, the more cross-connected capacity will be provided, offering the more throughput, however, the less I/O ports will be left to the I/O devices; the more I/O ports are available to the I/O devices, the less ports left for the cross-connected function, thus, the total throughput will be reduced. This thesis is based on the cross-connected SAN.

In Fig 2.12, all I/O ports are duplex ports, including input port and output port. If the ports (switches) are treated as the combination of input port (input module) and output port (output module), the topology of cross-connected SAN can be considered as a Clos network (Fig 2.13).

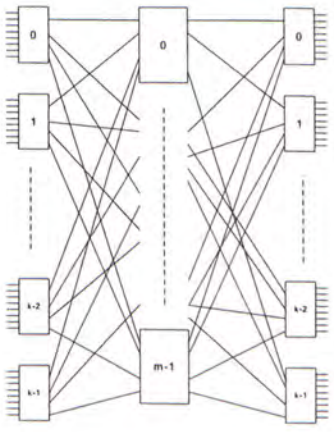

Figure 2.13 Clos Network

The only difference between the Clos network and the cross-connected SAN is that in the later case the internal traffic within a switch doesn't pass through the cross-connected switch, which should be routed within the switch itself. If we use matrix to represent the traffic between any input module and any output module in the Clos network derived from cross-connected SAN, the value in the diagonal line should be equal to zero (Fig 2.14).

$$\begin{bmatrix} 0 & \cdots & \\ & 0 & \cdots \\ \vdots & \vdots & \ddots \\ & \cdots & 0 \end{bmatrix}$$

Figure 2.14 Traffic Matrix

2.2.4 Routing Scheme in Cross-Connected SAN

Traditionally, circuit switching is adopted as the routing scheme in cross-connected SAN [8], which means dedicated circuits should be set up between the I/O ports involved with the accepted request, whether the accepted request is connection-oriented or connectionless. The new traffic request would be discarded in case that the output port is occupied by an existing connection or there are not enough CC switches. Circuit switching is easy to be implemented, however, there are several disadvantages of this scheme:

Suppose there is an accepted request from I/O port X to Y in cross-connected SAN, two dedicated circuits, namely forward link and backward link, should be set up in the Clos network (Fig 2.15) from input port X to output port Y and from input port Y to output port X. That is, the real traffic is sent on the forward link. It is easy to see that much bandwidth on the reverse link is wasted, since only the acknowledgement frame or flow control frame will be transferred on it. The definition of forward link and reverse link is based on the traffic: the link carrying the real traffic transmission is called forward link, the other one is called reverse link. For instance, suppose the server wants to fetch something from the storage device, the direction from the storage device to the server is called the forward link. On the other hand, if the server has the most fashionable film and wants to save it in the storage device, the direction from the server to the storage device is the forward link.

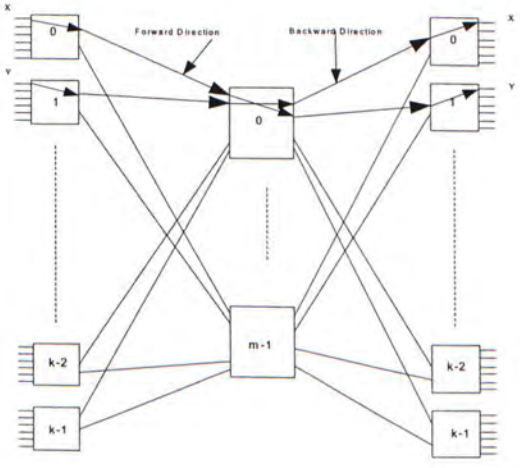

Figure 2.15 Circuit Switching

The second drawback of circuit switching is computational expensiveness: although the connection-oriented traffic (Class 1) occupies a large ratio of the total traffic, the number of connectionless traffic (Class 2/3) is much more than that of connection-oriented [8]. To most single frame traffic requests, dedicated connections must be set up. In most cases, the connection set up time is larger than the transmission time of a single frame. Hence, the routing scheme of circuit switching is computational expensive.

The main purpose of our research is to introduce a new routing scheme rather than the circuit switching scheme, which can save more bandwidth in cross-connect SAN and reduce the computation complexity.

Chapter 3

Path Switching

Traditionally, there are two routing schemes, namely static routing and dynamic routing, in Clos network [10], in which the most difficult issue is path and bandwidth assignment for each connection request. The static routing, such as circuit switching, does not fully exploit the statistical gain. In contrast, the dynamic routing, such as cell switching, requires slot-by-slot computation of route assignment. Recently, a quasi-static routing scheme, called path switching, is proposed to build large-scale ATM switch in Clos network [9], which is the compromise of the above two schemes. It uses a predetermined periodical connection pattern in the central stages, look-ahead scheme in the input stage, and output queuing in the last stage.

3.1 Cross-Path Switching Principle

The path-switched Clos network is called cross-path switch. A cross-path switch is a three-stage Clos network with the implementation of the quasi-static routing scheme, called path switching. Fig 3.1 shows a *NXN* Clos network. The routing of path switching is based on the concept of virtual path within the Clos network. We consider that there is a virtual path in Clos

network between an input module and an output module, which comprises all virtual circuits interconnecting any incoming port and outgoing port on this pair of modules.

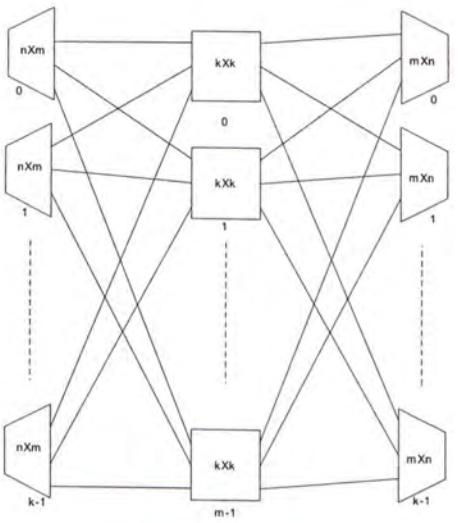

Figure 3.1 Clos Network

Instead of processing, scheduling, and routing for all incoming packets simultaneously by using a central controller, the routing scheme of cross-path switch is implemented in a distributed manner over three different stages with the whole Clos network. The predetermined connection pattern of the central modules will be used repeatedly (Fig 3.2). Storing the routing table in the local memory of every input module, the connection pattern of the central module is known in every time slot. Each connection pattern specified exactly how many packets can be delivered to a particular output module through which central module in that time slot. Based on this information, each input module will select those packets within local buffer

according to their destinations and priorities (Fig 3.3). The selection process will match the destination with the desired output modules only, hence the output module would have to handle the output port contention.

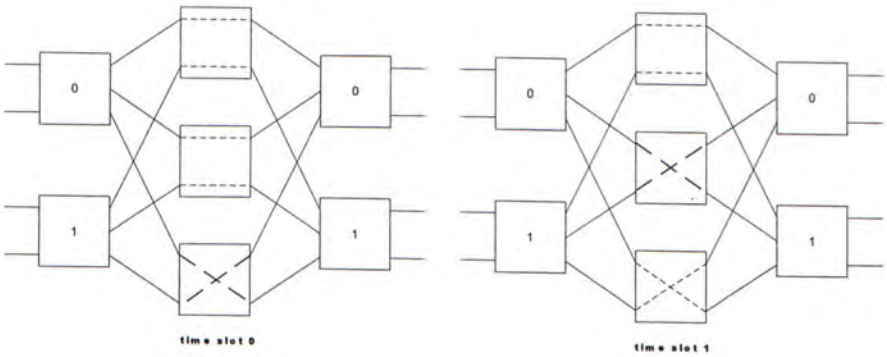

Figure 3.2 Connection Pattern of Central Module

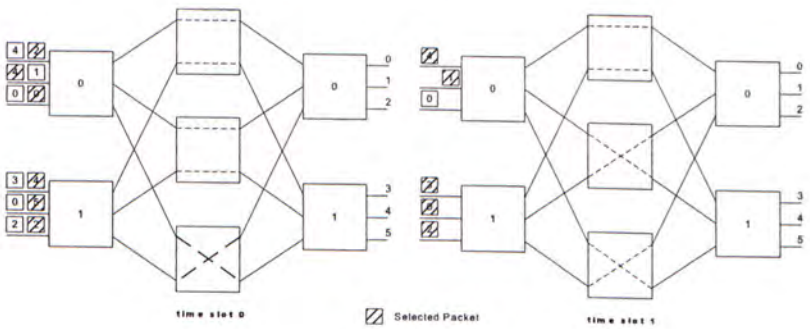

Figure 3.3 Routing Packets in Path Switching

If we consider each input module or output module as a node, a particular connection pattern in the middle stage of the Clos network can be represented by a regular bipartite multigraph as illustrated in Fig 3.4, where each central module corresponds to a group of edges. Each of them connects one distinct pair input-output modules. Suppose the routing scheme of the Clos

network adopts the dynamic cell switching, and the amount traffic from input module I_i to output module O_j is λ_{ij} cells per time slot. The connection pattern of the central module will change in every time slot according to arrival packets, and the routing table will be calculated on slot-by-slot basis.

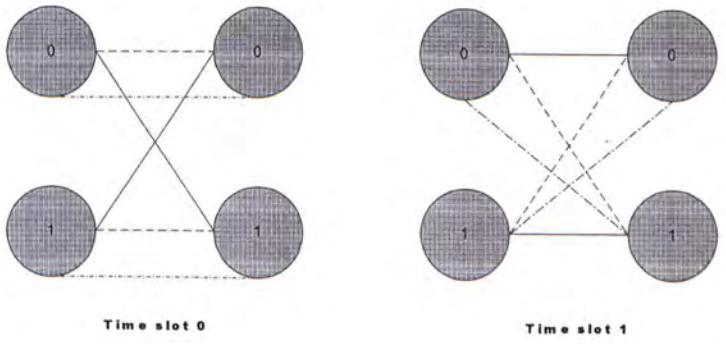

Figure 3.4 Bipartite Multigraph

Let $e_{ij}(t)$ be the number of edges from I_i to O_j of the corresponding bipartite multigraph in timeslot t. Then the capacity C_{ij} of the virtual path between I_i and O_j must satisfy:

$$C_{ij} = \lim_{T\to\infty} \frac{\sum_{t=1}^{T} e_{ij}(t)}{T} \geq \lambda_{ij}$$

On the other hand, the routing of circuit switching is fixed, and the connection pattern will be the same in every time slot. The capacity C_{ij} of the virtual path between I_i and O_j must satisfy:

$$C_{ij} = e_{ij}(t) = e_{ij} \geq \lambda_{ij}$$

which implies that the peak packet bandwidth C_{ij} is provided

for each virtual circuit at call setup time and doesn't take statistical multiplexing into consideration at all. We conceive that idea of quasi-static routing, called path switching, using a finite number of different connection patterns in the middle stage repeatedly, as a compromise of the above two extreme cases. For any given λ_{ij}, we can always find a finite number f of regular bipartite multigraph such that

$$C_{ij} = \frac{\sum_{t=1}^{f} e_{ij}(t)}{f} \geq \lambda_{ij}$$

where $e_{ij}(t)$ is the number of edges from input module *i* to output module *j* at time slot t. *f* is called the repetition rate of the path switching. And this finite amount of routing information is stored in the local memory of each input module to avoid slot-by-slot computation of route assignment. In two extreme cases, the path switching becomes circuit switching if *f* is equal to 1, and it is equivalent to cell switching if $f \to \infty$.

The scheduling of path switching consists of two steps, namely the capacity assignment and the route assignment, which will be explained detailedly in the following subsection.

3.2 Capacity Assignment

The capacity assignment is to find the capacity $C_{ij} \geq \lambda_{ij}$ for each virtual path from input module I_i to Output module O_j, where λ_{ij} is the aggregate CBR (Constant Bit Rate) [9]

between this pair of modules. In the example shown in Fig 3.1, we assume that the following capacity constraints are observed by the call admission procedure:

$$\begin{cases} \sum_j \lambda_{ij} \le m \\ \sum_i \lambda_{ij} \le m \end{cases} \qquad \textbf{(1)}$$

We can use a matrix λ to represent the CBR (Fig 3.5) in path switching. In general cases, the row sum $\sum_j \lambda_{ij}$ and the column sum $\sum_i \lambda_{ij}$ of λ is less than m, which means the capacity of central module can not be over supplied. Therefore, we should find the capacity matrix C (Fig 3.6) subject to following constraints:

$$\begin{cases} \sum_j C_{ij} = m \\ \sum_i C_{ij} = m \\ C_{ij} \ge \lambda_{ij} \end{cases} \qquad \textbf{(2)}$$

$$\lambda = \begin{bmatrix} \lambda_{0,0} & \lambda_{0,1} & \cdots & \lambda_{0,k-1} \\ \lambda_{1,0} & \lambda_{1,1} & \cdots & \lambda_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{k-1,0} & \lambda_{k-1,1} & \cdots & \lambda_{k-1,k-1} \end{bmatrix}$$

Figure 3.5 CBR Matrix

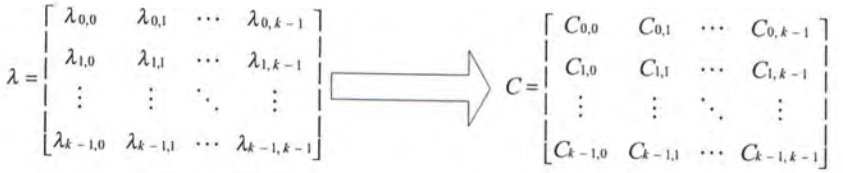

Figure 3.6 Capacity Assignment

The capacity assignment can be carried out by optimizing some objective function subject to constraint (2). The choice of objective function depends on the stochastic characteristic of the virtual paths and the quality of service requirements of connection [9]. For instance, in the cross path switch with two-dimensional constraint in (2), we can minimize the following total weighted offered load objective function:

$$z = \sum_{i,j} \frac{\lambda_{ij}^{2}}{C_{ij}}$$

Alternatively, each virtual path can be modeled as an independent *M/M/1* queue with arrival rate λ_{ij} and service rate C_{ij} ; then the average delay for the packets from input module i to output module j is given by:

$$Tij = \frac{1}{C_{ij} - \lambda_{ij}} \qquad \textbf{(3)}$$

The objective is to minimize the total weighted delay [19], [20]:

$$z = \sum_{i,j} \frac{\lambda_{ij}}{C_{ij} - \lambda_{ij}} \qquad (4)$$

There are several efficient ways that can be used to solve the optimization problem [21]. The objective function (3) and (4) are convex; each can be minimized subject to the linear constraint (2) by using the sequential-approximation algorithm [22], or the generalized reduced gradient (GRG) method [23] for convex programming. Moreover, since it is a sum of single-variable functions, and each of them can be approximated by a linear function, the problem can be transferred into a linear programming function, and solved by the modified simplex method [21] to reduce the computation complexity.

Although there are many ways to solve the objective function, the computation complexity is exponential [14]. In Chapter 4, a new algorithm will be proposed to reduce the computation complexity from exponential to polynomial. The good news is that the capacity assignment is not a slot-by-slot task, it would be carried out if and only if the CBR matrix changes significantly and the switch performance becomes unacceptable.

3.3 Route Assignment

Upon the completion of capacity assignment, the capacity matrix C should be converted into edge coloring of a finite number f of regular bipartite multigraph, each of which represents a particular connection pattern of central modules in the Clos network (Fig 3.7) [9]. In the Clos network shown in Fig

3.1, an edge coloring of a bipartite multigraph is to assign m distinct colors to m edges of each node such that no two adjacent edges have the same color. It is well known that a regular bipartite multigraph with degree m is m-colorable [24] [25]. Each color corresponds to a central module, and the color assigned to an edge from input module i to output module j represents a connection between them through the corresponding central module.

Suppose that we choose a sufficient large integer f such that fC_{ij} are integers for all i,j, and form a regular bipartite multigraph, in which the number of edges between node i and node j is fC_{ij}. Since the new multigraph is regular with degree fm, it can be edge colored by fm different colors [24]. Furthermore, it is easy to show that the any edge coloring with degree fm is the superposition of the edge coloring of f regular bipartite multigraph of degree m [9]. The coloring problem can be solved by time-space principle [9]. Coloring can be found be complete matching, which is repeated recursively to reduce the degree of every node one by one. One general method to search a complete matching is the so-called Hungarian algorithm with the worst time complexity $O(k^2)$ [26], or totally $O(fmk^2)$ since there are fm matching. If each of fm and k is a power of two, an efficient parallel algorithm proposed in [27] for conflict-free routing scheduling in three stage Clos network with time complexity of $O(\log^2(fmk))$ can be used.

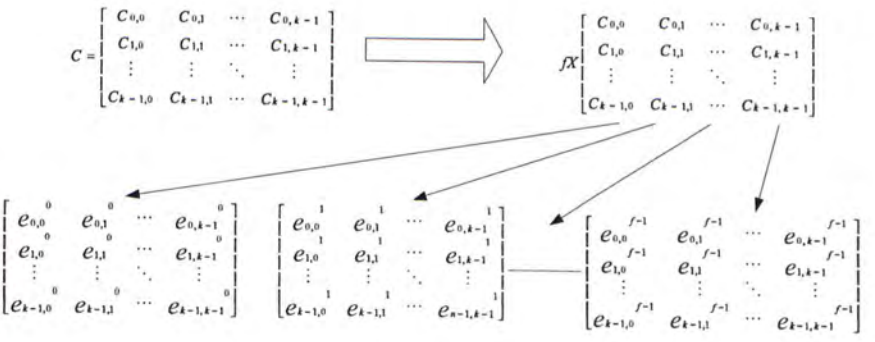

Figure 3.7 Route Assignment

For instance, Fig 3.8 shows the route scheduling of path switching in central modules for the Clos network under the uniform input traffic.

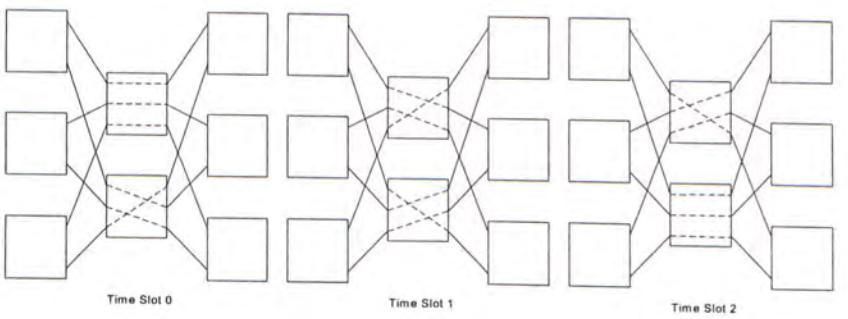

Figure 3.8 Route Scheduling

Chapter 4

Path Switching in Cross-Connected SAN

In this chapter, we will implement path switching in cross-connected SAN. Before introducing the new scheme, the size of the SAN should be determined. We will choose an environment in which there are 16 FCS. As mentioned in Chapter 2, the normal FCS has 16 ports. Suppose all the FCS are connected through a meshed topology, it is impossible to build a SAN with more than 16 FCS. Hence, 16 FCS are chosen as our studied model. In order to analyze the relationship between the performance of the whole SAN and the number of I/O devices supported, the number of cross-connected switch is 2,4,6,8, respectively. Furthermore, the traffic is considered as uniformly distributed, which means that the input traffic is equally likely to be destined to any one of its output address.

This chapter is organized as follows: in subsection 4.1, the performance under different traffic in SAN is studied in theory and simulation. A new algorithm will be proposed to reduce the computation complexity in subsection 4.2. The choice of repetition rate f is discussed in subsection 4.3.

4.1 Path Switching in SAN

Instead of using the circuit switching scheme or dynamic switching scheme in cross-connected SAN, the path switching

can be implemented in it, which means that the predetermined connection pattern of central cross-connected switches and the repetition rate *f* should be found according to given traffic. In this subsection, we will show how to find the connection pattern and the repetition rate *f*. Furthermore, the performance of path switching in different traffic is also studied.

Before studying path switching in SAN, a new term will be introduced: *token*. A token of a virtual path is mapped to a physical connection provided by the central module, connecting the corresponding input-output module pair in a time slot. This is illustrated in Fig 4.1. In path switching, when a new call arrives, the switch fabric will allocate the requested internal bandwidth (token) to the call. The packets of that call simply use the preassigned tokens to pass from the input module to the targeted output module.

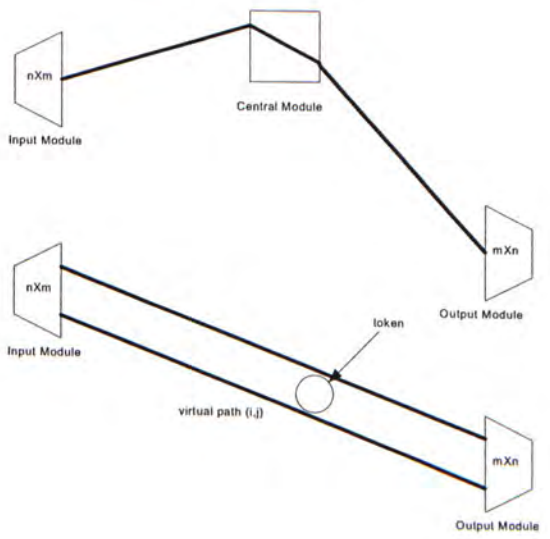

Figure 4.1 Token

Suppose the repetition rate *f* and the number of token in the virtual path is fixed in path switching, the token distribution has a large impact on the performance. That is, the tokens on a virtual path have to be distributed as uniformly as possible within the repetition rate because the burstiness of tokens may adversely affect the Quality of Service [11]. For instance, there are three tokens from input module 0 to output module 1 within the fixed repetition rate 3. The token distribution can have three different forms shown in Fig 4.2. To achieve better performance, the first case is the best choice. Thanks to the research of M.C.Chan and S.Y.Liew [11][12], an algorithm can be found to assign the tokens in each virtual path as uniformly as possible [12].

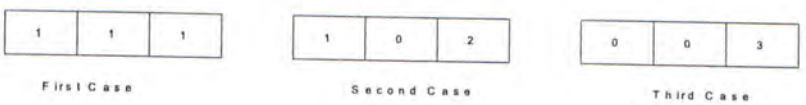

Figure 4.2 Token Assignment

4.1.1 Connectionless Traffic

In Chapter 2, it is known that there are two kinds of traffic in SAN: namely connection-oriented traffic (Class 1) and connectionless traffic (Class 2/3). Only the connectionless traffic (Class 2/3) will be considered in this subsection.

Given the connectionless traffic in SAN, traditionally deterministic dynamic routing or randomized dynamic routing is adopted [28], both of which require slot-by-slot routing calculation. In path switching, the routing pattern of central switches is predetermined. Therefore, the routing calculation on

the fly can be avoided.

Since the destination of connectionless traffic is uniformly distributed, the repetition rate *f*, is easily determined. *f* is the smallest integer to make $\frac{fm}{k}$ to be an integer, where *m* is the number of central cross-connected switch and *k* is the number of I/O FCS. In our example, *k* is equal to 16. *m* is equal to 2,4,6,8 respectively. After *f* is determined, the *fm* edges of each node in the multigraph can be evenly divided into k groups. Each of them contains $g = \frac{fm}{k}$ edges in every virtual path with the repetition rate *f*. The edges of the multigraph can be easily colored by the *Latin Square* given in table 4.1,

	O_0	O_1	O_2	...	O_{k-1}
I_0	A_0	A_1	A_2	...	A_{k-1}
I_1	A_{k-1}	A_0	A_1	...	A_{k-2}
⋮	⋮	⋮	⋮	⋱	⋮
I_{k-1}	A_1	A_2	A_3	...	A_0

Table 4.1 Latin Square Assignment

Where each A_i, $0 \le i \le k-1$, represents a set of distinct colors, e.g.:

$$A_0 = \{0, k, \cdots, (g-1)k\}; A_1 = \{1, k+1, \cdots, (g-1)k+1\};$$
$$\cdots; A_{k-1} = \{k-1, 2k-1, \cdots, kg-1\}$$

Since each number in the set $\{0,1,\cdots,kg-1\}$ appears only once in each column or row of table 4.1, this assignment is a legitimate edge coloring. After the repetition rate *f* is determined and the coloring method is found, the connection pattern of central cross-connected switches can be fixed. Concerning with

the uniform token distribution in every virtual path, we can consider a particular color assignment $a \in \{0,1,\cdots, fm-1\}(fm = kg)$ of an edge between input module I_0 and output module O_1 of the capacity graph, let

$$a = r \bullet m + t$$

where $r \in \{0,1,\cdots, f-1\}$ and $t \in \{0,1,\cdots, m-1\}$ are the quotient and the remainder of dividing *a* by *f*, respectively. The mapping $g(a) = (t,r)$ from the set $\{0,1,\cdots, fm-1\} \rightarrow \{0,1,\cdots, f-1\} X \{0,1,\cdots, m-1\}$ is one-to-one and onto, i.e.:

$$a = a' \Leftrightarrow t = t' \; and \; r = r'$$

That is, the assignment (*t,r*), of the edge between I_0 and O_1 indicates that the central module *t* has been assigned to a route from I_0 to O_1 in the rth time slot within each repetition rate *f*. With the above algorithm, the token distribution will be as uniformly as possible.

For instance, in the case of 6 central cross-connected switches, *f* is equal to 8 (8X6/16=3, 3 is an integer). Hence, *g* is equal to 3, which means there are equally 3 tokens in each virtual path within the repetition rate 8. From the above example, we can get the token distribution shown in Fig 4.3 in virtual path from input module I_0 to output module O_o, which is uniformly distributed.

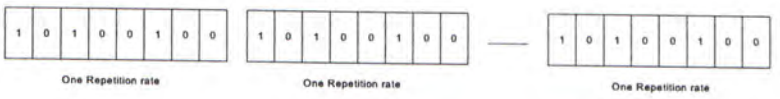

Figure 4.3 Token Distribution

For there is only connectionless traffic in this case, the time delay is the most important parameter to be examined. The simulation result is shown in Fig 4.4.

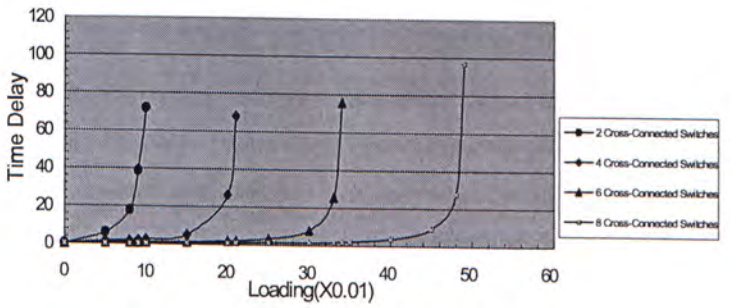

Figure 4.4 Delay Performance

From the results we can see that the SAN with 8 central cross-connected switches has the best time delay performance and the maximum input loading can reach 0.47, but only (16-8)X16=144 I/O devices can be supported; on the other hand, the SAN with 2 central cross-connected switches has the worst time delay performance and the maximum input loading could be only 0.09, but as much as 224 ((16-2)X16) I/O devices can be connected in this environment. So there is a tradeoff between the performance of SAN and the number of I/O devices supported.

4.1.2 Connection-Oriented Traffic

In this subsection, only the connection-oriented traffic (Class 1) is assumed to be existing in the SAN. Since Class 1 traffic is the traffic with a large block of data (Fig 2.5), the only way to transfer that kind of traffic is to set up dedicated connection, using the circuit switching scheme. Hence, the blocking probability is the most used parameter to describe the performance of connection-oriented traffic.

In Clos network, the new connection-oriented request will be blocked due to two reasons: the output port is occupied by an existing request or there are not enough central modules (Fig 4.5). In the following study, we want to find the blocking probability in theory. Therefore, we assume the Clos network studied is rearrangeable or strictly nonblocking [29], which means the new traffic request can not be blocked due to no enough central modules (Fig 4.6). With such assumption, the connection-oriented request will be blocked if and only if the output port is occupied by an established connection.

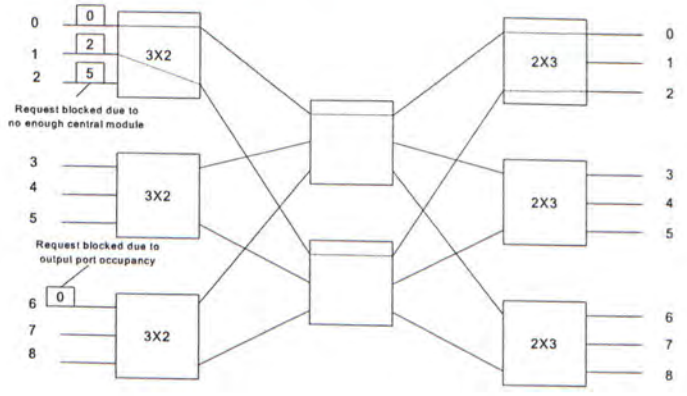

Figure 4.5 Blocking in Clos Network

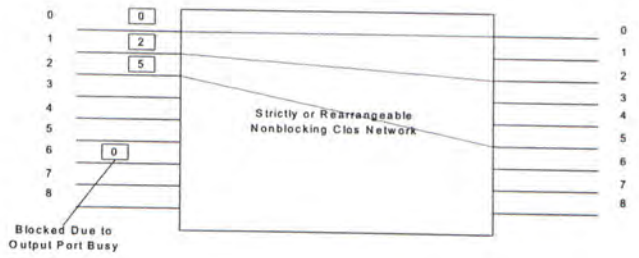

Figure 4.6 Blocking in nonblocking Clos Network

Assuming each Class 1 traffic has a fixed block size: L, given a traffic density *u* between 0 and 1, the input traffic of Clos network shown in Fig 4.6 are generated using a negative exponential distribution with an average inter-arrival time of *L*/*u* [13]. The new incoming request in input port a may be stored in the input buffer when the input port a is occupied by a former request and the buffer of a has not been full yet. If the input buffer has been full, the new incoming request will be discarded (not blocked) (Fig 4.7). Suppose each input link has *d* buffers, where $d \in (0,+\infty)$, according to the total request number every input port can have *d*+2 states from s(0), s(1) to s(d+1), which is shown in Fig 4.8.

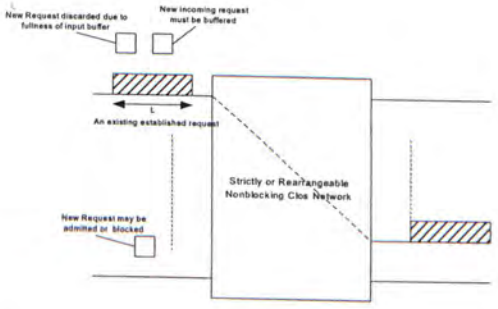

Figure 4.7 Sample (Buffer=1)

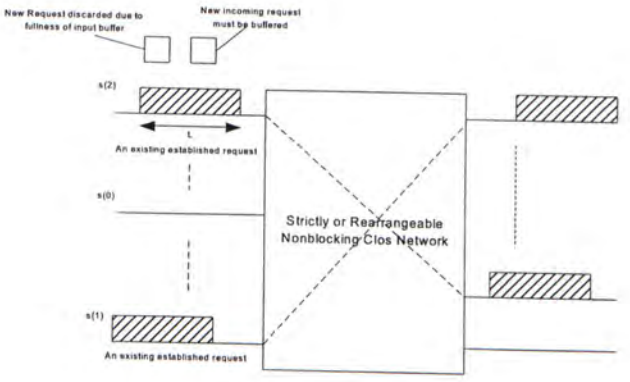

Figure 4.8 Input Port Status (d=1)

s(0) stands for the input port is idle;

s(1) stands for the input port is occupied by an admitted request and the buffer is empty

s(k) $\{2 \le k \le d+1\}$ stands for the input port is occupied by an admitted request and there are k-1 requests stored in the buffer.

Suppose there is an NXN (N is very large) strictly or rearrangeable nonblocking Clos network, P(k) $\{0 \le k \le d+1\}$ stands for the probability that the input port is in the state of s(k), P(I) (P(B)) stands for the probability that the output port is Idle (Busy), it can be seen that the P(I) is equal to P(0) since an input port in idle state implies that an output port is in the idle state also. Since N is very large, we can assume that the requested output port state is independent with the state of the input port issuing the request. The following equations can be got:

$$\begin{cases} P(I)+P(B)=1 & \textbf{(1)} \\ P(I)=P(0) & \textbf{(2)} \\ u\{P(0)+P(1)+\cdots+P(d)\}P(I)=P(B) & \textbf{(3)} \end{cases}$$

The equation (3) is explained as following: normally, the rate of incoming traffic is *u*, however, if the input buffer is limited, the incoming request will be discarded when the input port is in the state of *P(d+1)*. Hence, the effective rate of incoming traffic should be $u\{P(0)+P(1)+\cdots P(d)\}$. Furthermore, the admitted request rate is $u\{P(0)+P(1)+\cdots P(d)\}P(I)$, which means the request will be admitted if and only if the output port is idle. Beyond that, the probability that the incoming request is admitted should be equal to the probability that the output port is busy, because a request is admitted in the input port, there should have an output port to be in the Busy state at the same time.

From (1)(2)(3), we can have another equation:

$$u\{P(0)+P(1)+\cdots P(d)\}P(0)=P(B) \qquad \textbf{(4)}$$

Therefore, we can consider two extreme cases: $d=0 \; or \; d=+\infty$, which means there is no buffer in the input port or there is infinity buffer in the input port. In the first case, we can get:

$$uP(0)P(0)=P(B)=1-P(0) \Rightarrow P(B)=\frac{2u+1-\sqrt{1+4u}}{2u} \qquad \textbf{(5)}$$

In the second case, since $P(0)+P(1)+\cdots P(d)=1$, the next equation can be got:

$$uP(0) = P(B) = 1 - P(0) \Rightarrow P(B) = \frac{u}{1+u} \quad \textbf{(6)}$$

In fact, the equation (5) can be proved through the M/D/1 model also. For there is no buffer in the input link, the input port can be in the state s(0) or s(1), of which the state transition diagram from time t to time t+L is drawn in Fig 4.9.

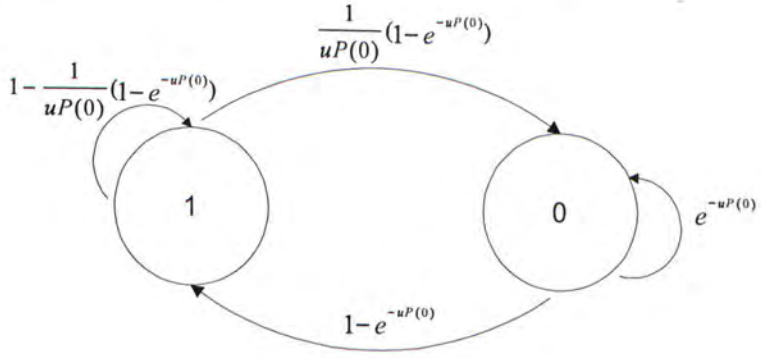

Figure 4.9 State Diagram (d=0)

P(0--1) is the transition probability that the input port is changed to state s(1) at time t+L if it is in the state s(0) at t. The transition probability can be calculated as following:

$$P(0--0) = e^{-u} + (1-P(0))u\,e^{-u} + (1-P(0))^2\frac{u^2}{2!}e^{-u} + \cdots + (1-P(0))^n\frac{u^n}{n!}e^{-u} = e^{-u}e^{u-uP(0)} = e^{-uP(0)} \quad \textbf{(7)}$$

In the above equation, the first term means that there is no new arrival in the interval (t, t+L). The second term means that

there is one new request in the interval but this request is blocked for the requested output port is busy. The last term means that there are n new arrivals but all of them are blocked due to output port occupancy. In the steady state, the sum of P(0--0) and P(0--1) should be one. Therefore,

$$P(0--1) = 1 - e^{-uP(0)} \tag{8}$$

P(1—0) is the transition probability from the state s(1) to s(0). From the graph 4.10, the following equation can be got:

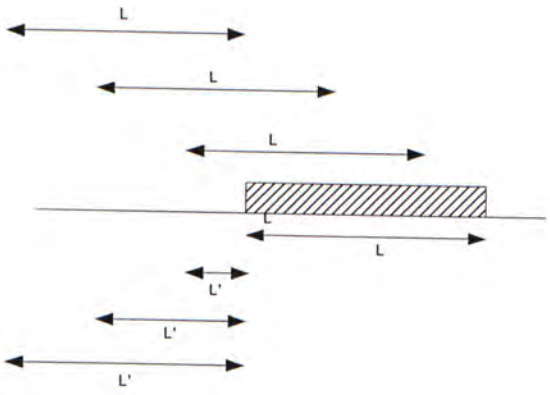

Figure 4.10 P(1--0)

$$P(1--0) = \frac{1}{L}\int_0^L (e^{-u\frac{L'}{L}} + (1-P(0))u\frac{L'}{L}e^{-u\frac{L'}{L}} + (1-P(0))^2\frac{(u\frac{L'}{L})^2}{2!}e^{-u\frac{L'}{L}} + \cdots + (1-P(0))^n\frac{(u\frac{L'}{L})^n}{n!}e^{-u\frac{L'}{L}})dL'$$

$$\Rightarrow P(1--0) = \frac{1}{L}\int_0^L (e^{-uP(0)\frac{L'}{L}})dL' = \frac{1}{uP(0)}(1-e^{-u}) \tag{9}$$

$$\Rightarrow P(1--1) = 1 - \frac{1}{uP(0)}(1-e^{-u}) \tag{10}$$

In the steady state, the transition probability from s(1) to s(0) should be equal to the probability from s(0) to s(1), therefore:

$$P(1)P(1--0) = P(0)P(0--1) \Rightarrow P(1) = uP(0)^2 \Rightarrow P(B) = uP(0)^2 \quad \textbf{(11)}$$

We can see equation (11) is the same as the equation (5). If B is used to represent the blocking probability of the request, in the case of infinity buffers,

$$B = P(B) = \frac{u}{1+u} \quad \textbf{(12)}$$

But in the case of zero buffer, B=P(I)XP(B), this is because the blocking probability is equal to multiply of the probability that input port is idle and the probability that the requested port is busy.

$$B = P(I)P(B) = (1 - P(B))P(B) = \frac{\sqrt{1+4u}-1}{2u}\frac{2u+1-\sqrt{1+4u}}{2u} \quad \textbf{(13)}$$

Fig 4.11 shows the simulation and theoretical results, which fit well and prove that our theoretical model is precise enough.

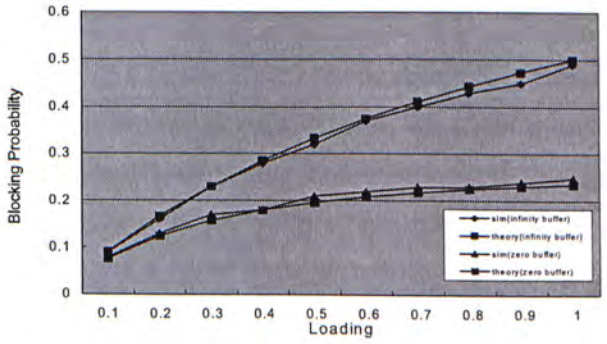

Figure 4.11 Blocking of Connection-Oriented Traffic

4.1.3 Mixed Traffic

In the former subsection, the connectionless traffic and connection-oriented traffic are introduced individually. But in the real environment, the two kinds of traffic, namely Class 1 and Class 2/3, exist simultaneously. In this part, we should study the performance of cross-connected SAN under the mixed traffic.

In the old circuit switching mentioned in Chapter 2, the two way dedicated connections should be set up for each accepted request, regardless that it is connectionless or connection-oriented. It is clear that this traditional routing scheme causes a lot of bandwidth consummation and is computational expensive.

In fact, we notice that after connections are set up a block of data with 1500 consecutive frames is transferred in the forward link, but only some acknowledge frames or flow control frames pass through the reverse link, resulting the waste of bandwidth. Therefore, if we can only assign just a certain percentage of bandwidth in the reverse link, not the whole bandwidth, to that connection, the efficiency will be greatly increased.

Now we are going to implement path switching in cross-connected SAN to achieve better blocking-delay performance, the working principle is very simple: we should build CBR matrix for the connection-oriented traffic first, and then the left traffic will be distributed to the connectionless traffic.

Firstly, we should construct the CBR matrix for the connection-oriented traffic, in which the circuit switching is still used. In the forward link we use **1** to represent the bandwidth requirement in the CBR matrix, however, in the reverse link, we will use **impact factor *a*, 0<*a*<1**, to represent the bandwidth of it in CBR matrix. In the extreme case of circuit switching scheme,

a is equal to 1. When this accepted connection-oriented request finishes transmission, both values will return to zero. Suppose **A** is used to represent the forward link CBR matrix, the CBR matrix of reverse link can be expressed as $\boldsymbol{aA}^T$, where $\mathbf{A}^T$ is the transpose of **A**. Then the total CBR matrix can be expressed as $\mathbf{A}+ \boldsymbol{aA}^T$ (Fig 4.12). After the CBR matrix is fixed, the capacity assignment of path switching should be found to determine the connection pattern of central cross-connected switches within the repetition rate. Therefore, it is easy to see that less reversed bandwidth will be consumed by the connection-oriented traffic (Class 1), and the left capacity will be distributed to the connectionless traffic (Class 2/3). Comparing with the traditional circuit switching scheme, path switching reduces reverse bandwidth requirement, hence the blocking probability of SAN will be no worse than that of the circuit switching. Furthermore, the left capacity for the connectionless traffic will be improved and computing route of each individual connectionless request on the fly can be avoided. Only when the aggregate CBR matrix of FCS is changed, the CBR matrix needs to be recomputed and the connection pattern of central cross-connected switches should be rearranged also. So the routing computation time is much less than that of the circuit switching scheme, whose routing computation is based on each incoming request.

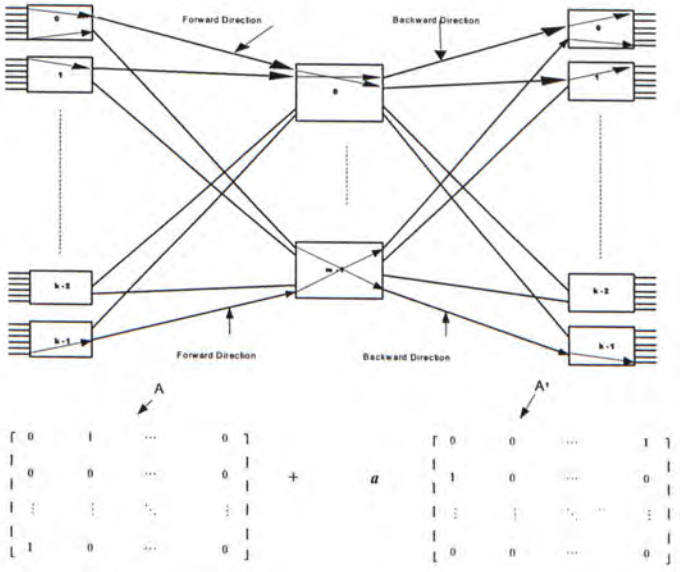

Figure 4.12 A+*a*A^T

Before comparing the simulation result, the traffic model must be introduced. Still assuming the environment of the SAN with 16 FCS, there will have 2,4,6,8 central cross-connected switches. The value of **reverse factor *a*** is chosen as 1 (Circuit switching), 0.4 (Path Switching), 0.2 (Path Switching). Moreover, 10 percent of the total request are Class 1, the other 90 percent is Class 2/3 [8]. The average length of the request is:

$$avg\_length = 1500 \bullet 0.1 + 1 \bullet 0.9 \approx 150$$

Given a traffic loading *u* between zero and one, the input traffic is generated by using a negative exponential distribution with an average inter-arrival time *avg_length/u*. The new request has 10 percent to be Class 1 traffic, and 90 percent to be Class 2/3 traffic. Furthermore, the destination of input traffic is uniformly distributed, which means the incoming request

chooses its destination uniformly among all the output ports and independently from all other requests. In real implementation, the request blocked will be regenerated, thus the input traffic can not be assumed to be uniform. This issue will be left for further study of this topic and surpass the scope of this thesis. The simulation results of different central cross-connected switches are shown in Fig 4.13, Fig 4.14, Fig 4.15, Fig 4.16 respectively.

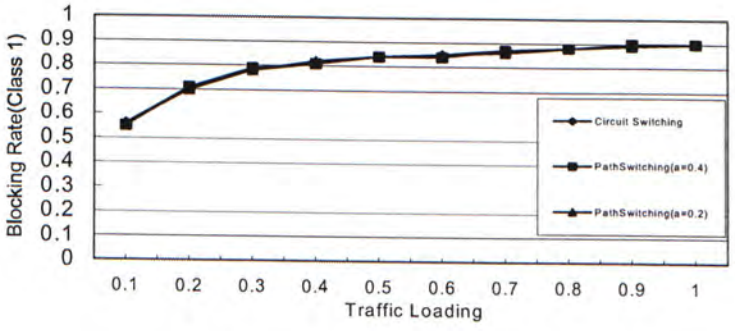

Figure 4.13 Two Cross-Connected Switches

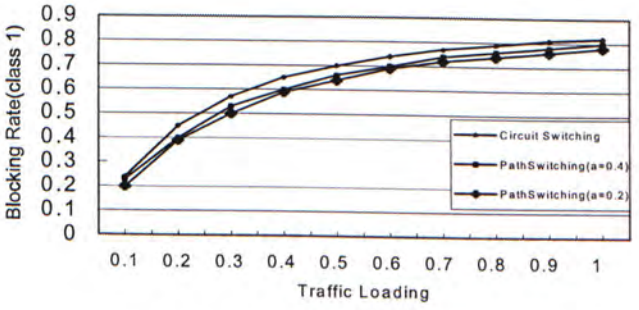

Figure 4.14 Four Cross-Connected Switches

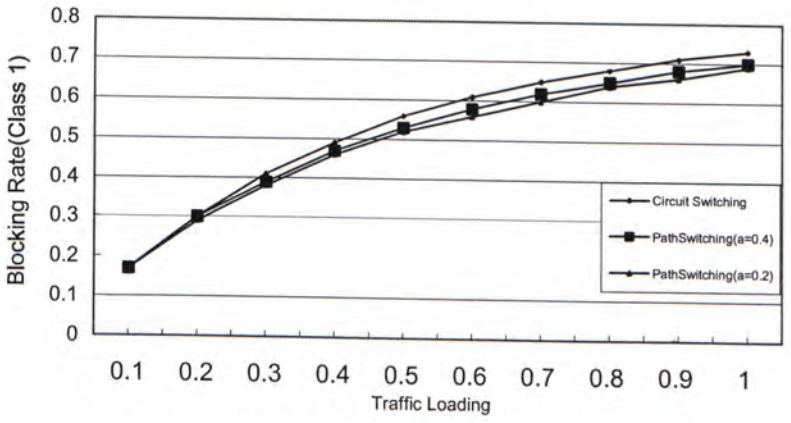

Figure 4.15 Six Cross-Connected Switches

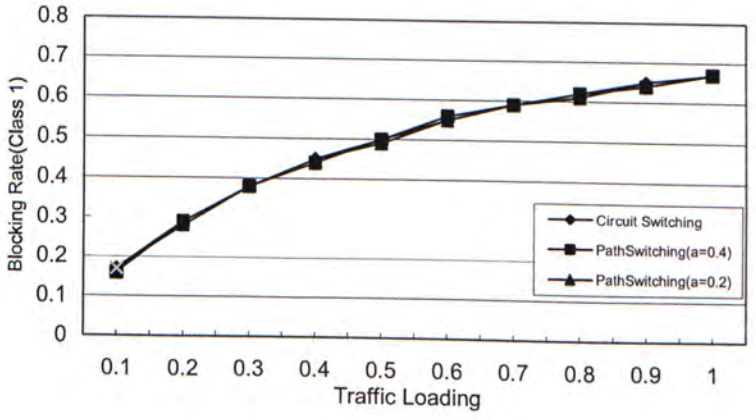

Figure 4.16 Eight Cross-Connected Switches

From the above simulation results, we can see that the blocking probability of path switching is 3 or 4 percent lower than that of the circuit switching in the case of 4 or 6 central

cross-connected switches, this is because path switching scheme requires less bandwidth in the reverse link. However, in the case of 2 or 8 central switches, the blocking probability is almost the same in two schemes. How could this happen?

In the case of 2 central switches, circuit switching can provide up to 2 connections for each FCS. On the other hand, path switching can not support more than 2 connection for each FCS simultaneously either. In Fig 4.17, suppose three connections could be provided by path switching, since ***a*** is larger than zero the column sum or the row sum of CBR matrix ($\mathbf{A}+a\mathbf{A}^T$) is larger than 2, which violates the call admission condition. Thus, the blocking probability is almost the same in the two schemes

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} + a \begin{bmatrix} 0 & 0 & 1 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Figure 4.17 Two Central Switches

For each FCS has 16 ports, in the case of 8 cross-connected switches there are only 8 ports left for the I/O devices, hence the circuit switching can support at most 8 connections from each FCS simultaneously. Although path switching can support more than 8 connections in this case, there are at most 8 connection requests from each FCS at the same time, the blocking probability of path switching will be almost the same as that of circuit switching.

Note that the bandwidth of reverse link is reduced, however,

we don't want to degrade the QOS (Quality of Service) of the traffic. Therefore, it is a great challenge to keep the QOS of the backward traffic in the path switching scheme. Thanks to the early study, it has been show that the more uniform token distribution over the time slots the less delay jitter encountered by each from. Hence, we need a route assignment to perform uniform token distribution. Liew has proposed an algorithm in [12] to implement stop-and-go queuing strategy in path-switched Clos network, which assures the token distribution as uniform as possible. In this thesis, this algorithm is used to convert the capacity matrix into the connection pattern in cross-connected switches in the simulation. Attention is put on the practicality and feasibility on the implementation of this algorithm.

4.2 Measurement Based Algorithm

In Chapter 3, capacity assignment in path switching has been introduced, the purpose of which is to find a matrix whose row sum and column sum is equal to m from CBR matrix, where m is the number of central cross-connected switches. Since the row sum and the column of the CBR matrix is less or equal to m after call admission of the connection-oriented traffic, the capacity assignment of path switching can fully utilize to capacity of central cross-connected switches.

Traditionally, we have to solve an objective function, in which the modified simplex method is often used [9]. However, in [14] it has been proved that the computation complexity of the

modified simplex method is exponential. In the implementation of cross-connected SAN, CBR matrix will be reconstructed if the total CBR traffic in any FCS is changed, and then the capacity assignment will be performed again. Hence, such high computation complexity should be avoided. Moreover, the C_{ij} in the resulting capacity matrix C is not integer in general, a sufficiently large *f*, repetition rate, must be found so that fXC_{ij} are all integers. Otherwise there will be roundoff error, which is inversely proportional to *f*. But the *f* can not be too large, which leads a lot of memory in the local memory in FCS [9]. Therefore, *f* is not easy to be determined in the old way.

In this thesis, a new algorithm, called measurement based algorithm, is proposed to reduce the computation complexity in capacity assignment. The working principle is that: instead of to find the optimal C_{ij}, the C_{ij} can be found to be strictly large $C_{ij} \geq \lambda_{ij}$ subject to $\sum_i C_{ij} = m$ **and** $\sum_j C_{ij} = m$. Since the requirement is lowered, the computation complexity of the measurement based algorithm is reduced to be polynomial, which will be proved later. Moreover, the repetition rate *f* is easy to be found, since all C_{ij} in C has a common denominator.

There are two steps in the measurement based algorithm. Firstly, the left capacity in each row is distributed evenly to form an intermediate matrix (Fig 4.18), in which all the values in the diaognal line are zero. This is because all the traffic from input module *j* to output module *j* will be handled by the switch *j* itself, there is no need for such traffic to pass through the central switches. After the first step, we can get a intermediate matrix C, in which the row sum is equal to *m*. Then the columns in C can

be divided into three groups according to their column sum: L (Column sum is larger than *m*), S (Column sum is smaller than *m*), E (Column sum is equal to *m*). Furthermore, the measurement unit, *U*, should be decided: $U = \frac{1}{h(k-1)}$, where *h* is a positive integer and *k* is the number of the FCS.

$$\lambda = \begin{bmatrix} 0 & \lambda_{0,1} & \cdots & \lambda_{0,k-1} \\ \lambda_{1,0} & 0 & \cdots & \lambda_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{k-1,0} & \lambda_{k-1,1} & \cdots & 0 \end{bmatrix} \Rightarrow C = \begin{bmatrix} 0 & \lambda_{0,1}+c_0 & \cdots & \lambda_{0,k-1}+c_0 \\ \lambda_{1,0}+c_1 & 0 & \cdots & \lambda_{1,k-1}+c_1 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{k-1,0}+c_{k-1} & \lambda_{k-1,1}+c_{k-1} & \cdots & 0 \end{bmatrix}$$

$$c_i = \frac{m - \sum_j \lambda_{ij}}{k-1}$$

Figure 4.18 The First Step in Measurement

After the intermediate matrix is found, the measurement can be performed to build the capacity matrix. The detail procedure of measurement is shown in Fig 4.19

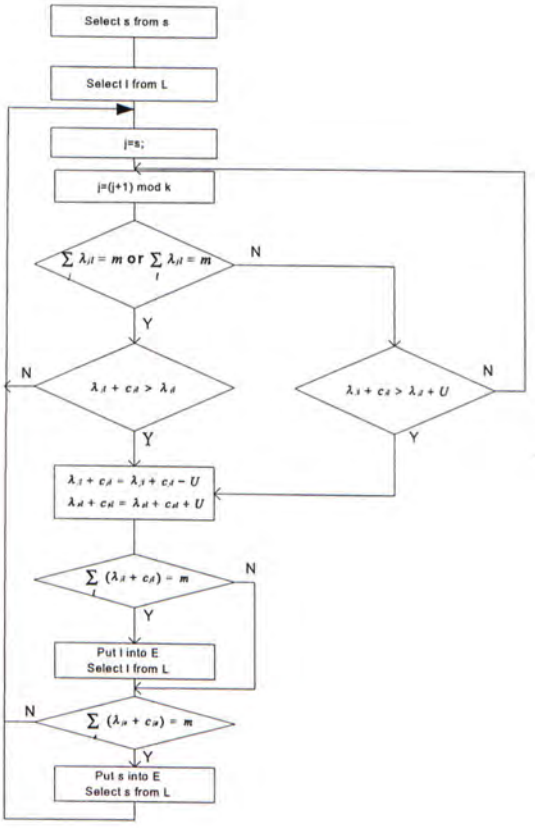

Figure 4.19 The Third Step in Measurement

Remark: from the above algorithm, we can see that:

(1) The row sum in the matrix C is still *m*. During the measurement, the row sum has never been changed. Therefore, all the row sum of matrix C is still *m*: $\sum_{l} C_{il} = m$

(2) The column sum in matrix is *m* after measurement, which is the goal of the measurement. From Fig 4.18, it is known that the total sum of all the elements in matrix C is *km*. Hence, if there is one column in the matrix whose sum is

larger than *m*, there must be another column whose sum is smaller than *m*. The whole measurement will proceed until all the column sum is equal to *m*: $\sum_i C_{il} = m$

(3) Suppose $\sum_l \lambda_{il} = m$, from Fig 4.18, we can know that : $\lambda_{ij} = C_{il}$. Thus, during the measurement the C_{ij} will not be changed. At the end of measurement, $\lambda_{ij} = C_{ij}$. Suppose $\sum_i \lambda_{il} = m$, all the C_{il} that is larger than λ_{il} must be adjusted to be equal to λ_{il}. Otherwise the column sum of I will be larger m, resulting the measurement going on. At the end of measurement, $\lambda_{il} = C_{il}$.

(4) Suppose $\sum_l \lambda_{il} < m$ **and** $\sum_i \lambda_{il} < m$, from the above measurement, it can be seen that at the end of measurement, $C_{ij} > \lambda_{ij}$ **or** $C_{ij} \geq \lambda_{ij} + U$.

(5) The measurement can proceed until all the column sum is equal to *m*. Given $\sum_j \lambda_{jl} < m$ **(1)** and $\sum_j C_{jl} > m$ **(2)**, we should prove that there is at least one row j in column I that is larger than $\lambda_{jl} + U$ **(3)**, then the measurement can proceed.

Proof:

We prove it by contradiction:

Assuming all $C_{jl} \leq \lambda_{jl} + U$, since C_{ll}=0;

$$\sum_j C_{jl} \leq \sum_j \lambda_{jl} + (k\text{-}1)U \quad \textbf{(4)}$$

From the definition of *U*, it is known that $U = \frac{1}{h(k-1)}$, where h is an positive integer. From (4),

$$\sum_j C_{jl} \le \sum_j \lambda_{jl} + (k\text{-}1) \bullet U \le \sum_j \lambda_{jl} + (k-1)\frac{1}{h(k-1)} \le \sum_j \lambda_{jl} + \frac{1}{h} \le m$$

which contradicts (2). Therefore, there is at least one row j in column I that is larger than $\lambda_{jl} + U$, which means the measurement can proceed.

(6) The time complexity of this algorithm is polynomial: $O(hmk^2)$, which can be known from the simulation programming. In the measurement simulation program, there are four loops: one is from 0 to *h*, one is from 0 to *m*, and the other two is from zero to *k*.

(7) It is easy to see that all in elements in matrix C has a common denominator: $h(k-1)$. Hence, the repetition rate *f* is easy to be determined: *h(k-1)*.

In Fig 4.20, a complete example is shown how to build the capacity matrix through measurement.

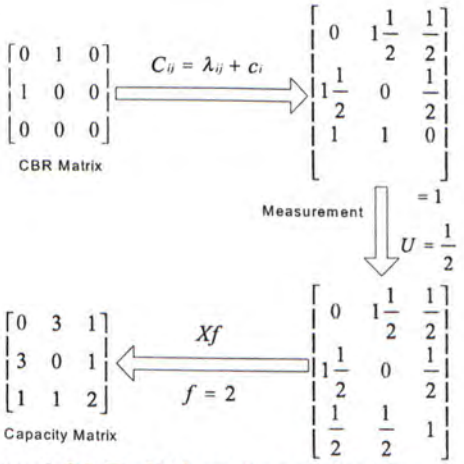

Figure 4.20 Building Capacity Matrix

4.3 Repetition Rate

From above subsection, it has been concluded that the repetition rate of path switching in cross-connected SAN is *h(k-1)*, where *k* is the number of FCS and *h* is positive integer determined by the designer himself.

Note that a critical problem has been ignored in the former discussion: the wasted token. In Chapter 2, it has been pointed out that the only difference between the cross-connected SAN and Clos network is that the traffic from input module j to output module j will be handled by the FCS itself, it is internal-switch traffic (Fig 4.21, Fig 4.22). Hence, while constructing the capacity matrix in path switching, the value of the elements at the diagonal line of the capacity matrix should be equal to zero. If there are some elements at the diagonal line in the capacity matrix after measurement that is larger than zero, they are called the wasted token (Fig 4.23). In our path switching scheme, the number of wasted token should be as small as possible.

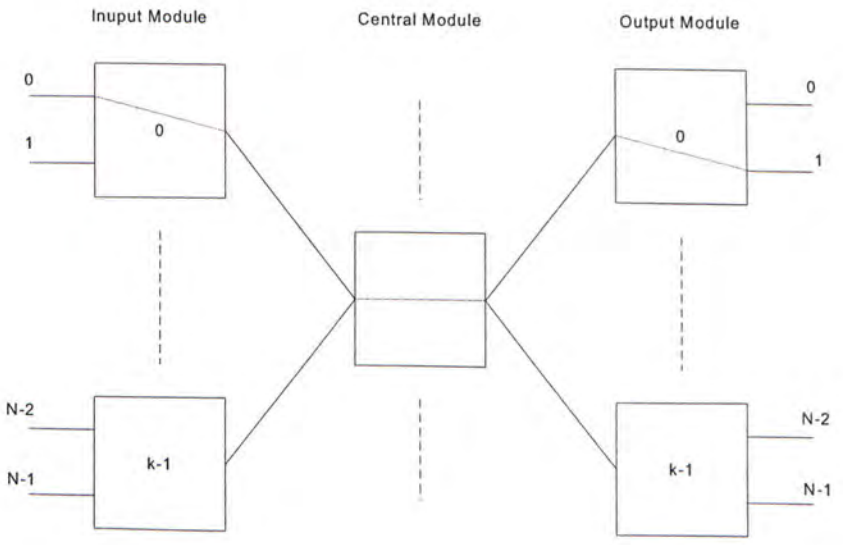

Figure 4.21 Connection in Clos network

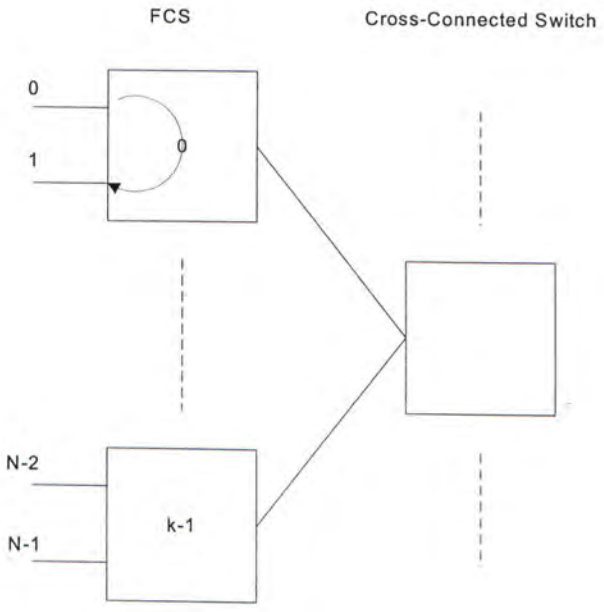

Figure 4.22 Connection in Cross-Connected SAN

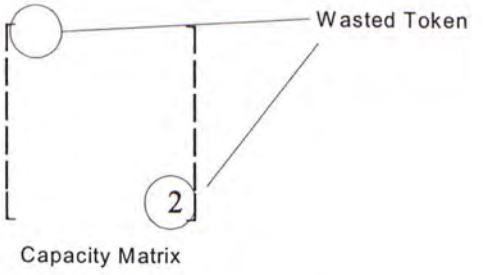

Figure 4.23 Wasted Token in Capacity Matrix

While building the CBR matrix, all the elements at the diagonal line are equal to zero. However, after the measurement, some elements at the diagonal line may be changed to be larger than zero, which is shown from the example in Fig 4.20 and causes wasted tokens. From this research, we find that the wasted token is inversely proportional to *h*. Moreover, since the repetition rate *f* is linearly proportional to *h* , we can say the wasted token is inversely proportional to *f*.

We still use the example of Fig 4.20, and do the measurement with different value of *h* in Fig 4.24, Fig 4.25.

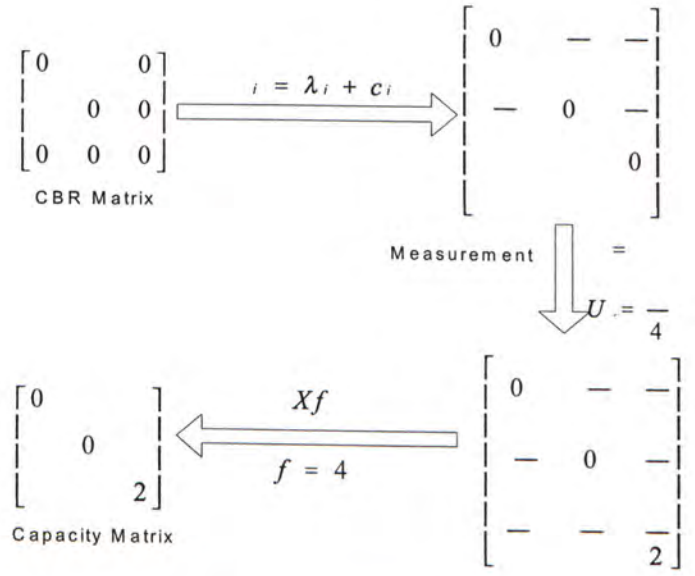

Figure 4.24 Measurement with *f*=4

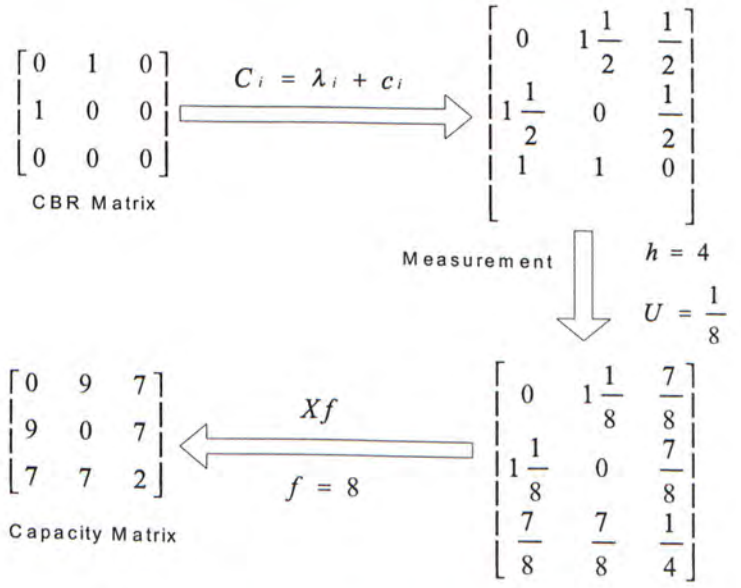

Figure 4.25 Measurement with *f*=8

Among the above two examples and the example in Fig 4.20, all of them have two wasted tokens at the last capacity matrix. But they have different repetition rates, which means two tokens are wasted in two time slots or four time slots or eight time slots. So the percentage of wasted token is the smallest in the example whose repetition rate *f* is eight, in the case that the repetition rate is two, the performance is the worst. The reason is following:

Suppose two columns are chosen for measurement, s and l, whose column is smaller or larger than *m* respectively. From the flow chart of the measurement, this algorithm will search row by row in column s and l but the row s will be visited at the last. Suppose *h* is larger, which implies *f* is larger and *U* is smaller, all the elements other than row s in column l will have more chance to do measurement with their counterpart in column s. So the row s has less probability to be measured, which means the element C_{ss} is smaller, resulting the less token wasted in the final capacity matrix.

In order to reduce the wasted token, the *f*, or the *h*, should be larger. However, it has been proved that the computation complexity of measurement based algorithm is $O(hmk^2)$, *h* is larger, resulting the compuation complexity is more complex. In addition, in path switching, the connection pattern of central croos-connected switches should repeat in the repetition rate *f*, which should be stored in the memory of input modules. The larger the *f*, the more memory required to store the routing information. Hence, the choice of f is a tradoff between the number of token wasted and the memory required.

Chapter 5

Conclusion

In this thesis, we proposed a quasi-static scheme, Path Switching, in cross-connected SAN. Instead of the old circuit switching scheme, this scheme can reduce the routing complexity. In addition, the blocking probability of the connection-oriented traffic can be lowered in some cases. Beyond these advantages, the left capacity for the connectionless traffic can be greatly improved.

A new algorithm, namely measurement based algorithm, is also introduced to find the capacity matrix of path switching in this research. The old measurement to find the capacity matrix has two shortcomings: the computation complexity is exponential, and the repetition rate of path switching is not easy to be determined. By using the measurement based algorithm, the computation complexity is decreased, which is proved to be polynomial. Furthermore, the repetition rate of path switching is fixed.

Another issue investigated in this thesis is the choice of the repetition rate in path switching. It has been shown that the capacity matrix after measurement can have some wasted token, which is inversely proportional to the repetition rate. However, the memory requirement in the FCS is linearly proportional to the repetition rate, since the connection pattern

of the central cross-connected switches within the repetition rate must be stored in the local memory of the FCS. Therefore, the choice of repetition rate is a compromise between the memory requirement and the number of token wasted.

Bibliography:

1. T. Anderson, R. Cornelius, "High-performance switch with Fibre Channel," *Digest of Papers, IEEE COMPCON*, February 1992.
2. L. Cherkasova, V. Kotov, T. Rokicki, "Evolution and design of high-performance fabric topologies," *HPL Report*, HPL-95-69, June 1995.
3. B. Phillips, "Have storage area networks come of age?" in *Computer*, vol. 31, pp. 10-12, July 1998.
4. "Fibre Channel – Arbitrated loop (FC-AL)," ANSI X3T9.3 Working Document, Revision 3.4, October 1993.
5. J.R. Heath and P.J. Yakutis, "High speed storage area networks using a fibre channel arbitrated loop interconnect," in IEEE Network, pp. 51 – 56, March-April 2000.
6. L. Cherkasova, V. Kotov and T. Rokicki, "Designing fibre channel fabrics," in *Proceeding of IEEE ICCD '95*, pp. 346 –351, Oct. 1995.
7. Larry.Olson, "Building SANS to scale", *InforStor,* Jan 1999
8. A. Varma, V. Sahai and R. Bryant, "Performance evaluation of a high-speed switching system based on the fibre channel standard," in *Proceedings of the 2nd International Symposium on High Performance Distributed Computing*, pp. 144 – 151, July 1993
9. T. T. Lee and C. H. Lam, "Path switching-a quasi-static routing scheme for large-scale ATM packet switches," *IEEE J. Select. Areas Commun*, vol. 15, pp. 914-924, June 1997.
10. Charles Clos, "A study of non-blocking switching networks," *The Bell System Technical Journal*, March 1953.

11. M.C. Chan, P.P. To, T.T. Lee, "Per-connection performance guarantees for cross-path switch," in the *Proceeding of IEEE ATM Workshop'99*, p. 469-474, May 1999.
12. S.Y. Liew, T.T. Lee, "Bandwidth assignment with QoS guarantees in scalable ATM switches," in *IEEE Transactions on Communications*, Vol 48,No.3, pp. 377-380, March 2000.
13. L.Cherkasova,V.Kotov,T.Rokicki."The Impact of Message Scheduling on Packet Switching Interconnect Fabric," *Proceedings of the 29th Annual Hawaii International Conference on System Science*-1996.
14. G.L.Nemhauser, A.H.G.Rinnooy Kan, "Optimization," *Elsevier Press,* 1989
15. X3T9.3 Task Group of ANSI: *Fibre Channel Physical and Signaling Interface* (FC-PH), Rev. 4.2 October 8, 1993
16. Fibre Channel Association: *Fibre Channel: Connection to the Future*, 1994, ISBN 1-878707- 19-1
17. Gary Kessler: Changing channels, *LAN Magazine*, December 1993, p69-78
18. Malavalli, K.; Stovhase, B, *Thirty-Seventh IEEE Computer Society International Conference*, Digest of Papers, 1992 , Page(s): 269 –274
19. D.Bertsekas and R.Gallager, *Data Network*, 2nd, ed, Englewood Cliffs, NJ: Printice-Hall,1992
20. A. Kershenbaum, *Telecommunication Network Design Algorithm.* New York: McGraw-Hill,1993
21. F.S.Hillier and G.J.Liberman, *Introduction to Operations Research*, 5th, ed, New York,:McGraw-Hill, 1990.
22. M.Frank and P.Wolfe, "An algorithm for quadratic programming ", *Nav. Res. Logist. Quat. Vol 3.* pp. 95-110, 1956
23. L.S. Lasdon and A.D.Warren, "Generalized reduced gradient software for linearly and nonlinearly constrained problems". *In the Design and*

Implementation of Optimization Software the Netherlands: Sijthoff and Noordhoff.

24. R.J.Wilson, *Introduction to Graph Theory*, New York: Academic, 1972.
25. F.T.Leighton, *Introduction to Parallel Algorithm and Architecture*, Arrays.Trees.hypercubes. Los Altos, CA: Morgan Kaufmann, 1992
26. R.J. McEliece, R.B Ash, and C.Ash *Introduction to Discrete Mathmatics*. New York: McGraw-Hll, 1989.
27. T.T.Lee and S.Y.Liew, "Parallel algorithm for benes networks" in *Proc IEEE Infocom'96*
28. L.Cherkasova, V.Kotov, T.Rokicki, "Simulation Study of Fibre Channel Fabrics with Particular Emphasis on 64-Node Clusters", *HPL Report*, HPL-95-69, June 1995
29. S.C.Liew and T.T.Lee, "Principles of Broadband Switching and Networking", *Lecture Notes*, Department of Information Engineering, Chinese University of HongKong, 1995.