A COMPARATIVE STUDY OF

ATM SWITCHING FABRICS

By

WEI YAN

Bachelor of Science

Nankai University

Tianjin, China

1990

Submitted to the Faculty of the Graduate College of the Oklahoma State University in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE May 2000

A COMPARATIVE STUDY OF

ATM SWITCHING FABRICS

Thesis Approved:

Thesis Advisor $^{\circ}$ h >amadzad M. H. Samadzad Wayne B. Pourll Dean of the Graduate College

PREFACE

Asynchronous Transfer Mode (ATM) is often described as the future computer networking paradigm that makes Broadband Integrated Service Digital Network (B-ISDN) possible. This new network will have a huge data rate compared to all existing networks and services, and will make it possible to offer a large variety of new services such as video, live television, full motion multimedia electronic mail, and compact disk quality music. When a frame is transmitted in the ATM network, it will be segmented into ATM cells and then routed through ATM switches. The switching fabric is important in reducing the possibility of blocking or increasing the throughput.

The thesis work is a comparative study of different ATM switching fabrics, including Starlite, Moonshine, Sunshine Switching, and other currently posted research works. The thesis also analyzes the performances of certain ATM switching fabrics under different traffics and internal and external interconnecting structures. Through the comparative study, we can see that in order to determine a suitable ATM switching fabric, we must consider the following aspects: scalability, congestion free, efficient use of bandwidth, reliability, traffic pattern and buffer management.

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my major advisor, Dr. H. K. Dai for his intelligent supervision, constructive guidance, inspiration and friendship. My sincere appreciation extends to my other committee members, Dr. J. P. Chandler, and Dr. M. H. Samadzadeh, whose guidance, assistance, encouragement, and friendship are also invaluable.

I would like to give my special appreciation to my family for their support and encouragement.

Finally, I would like to thank the Oklahoma State University Computer Science Department for supporting me during my two years of graduate study.

TABLE OF CONTENTS

ĩ

Chapter Page
I. INTRODUCTION
II. ATM CONCEPTS AND ARCHITECTURES
2.1 Concepts
2.2 Architectures
III. ATM SWITCHES
3.1 Attributes
3.2 Classification
3.2.1 Time-Division Switch
3.2.2 Space-Division Switch
IV. REPRESENTATIVE ARCHITECHTURES OF ATM SWITCHING FABRICS 12
4.1 Knockout Switch
4.2 Multistage Switch
V. ATM SWITCHING FABRIC DESIGNS AND GOALS
5.1 Starlite Switch Structure
5.2 Moonshine Switch Structure
5.3 Sunshine Switch Structure 19
VI. RESEARCH AND DEVELOPMENT
6.1 Tandem-Crosspoint Switch Architecture
6.2 Scalable Shared Memory ATM Switch
VIL PERFORMANCE ANALYSIS OF FAT-BANYAN SWITCH 29

7.1 Independent Uniform Traffic Pattern	29
7.2 Poisson Distribution Traffic Pattern	32
7.3 Analysis of Fat-Banyan Switch	34
VIII. CONCLUSIONS, DISCUSSION AND FUTURE WORK	57
REFERENCES	10
APPENDIX A: GLOSSARY	12
APPENDIX B: PROGRAM LISTINGS	14

LIST OF FIGURES

Figure Page
Figure 1. An ATM Cell
Figure 2. ATM Reference Model [13]
Figure 3. A Generic ATM Switch [13]
Figure 4. The Knockout Switch [13] 12
Figure 5. The Banyan Network (without blocking) [12] 14
Figure 6. The Banyan Network (with blocking) [12] 14
Figure 7. The Batcher-Banyan Network [9] 15
Figure 8. Starlite Switch Structure [7] 17
Figure 9. Moonshine Switch Structure [7]
Figure 10. Sunshine Switch Structure [12]
Figure 11. Tandem-Crosspoint Switch Structure [11]
Figure 12. The Fat-Banyan Topology [1]
Figure 13. The Truncated Fat-Banyan Topology [1]
Figure 14. Fat Switching Element
Figure 15. Maximum Throughput for the Fat-Banyan Switch Under Independent Uniform Traffic Pattern
Figure 16. Reasonable Arrival Rate for the Fat-Banyan Switch Under Poisson Traffic Pattern
Figure 17. Maximum Throughput for the Fat-Banyan Switch Under Poisson Traffic Pattern

CHAPTER I

INTRODUCTION

Asynchronous Transfer Mode (ATM) is a connection-oriented, high-speed switching and multiplexing networking technology. It has been accepted in the telecommunication and data communication fields. One advantage of this new technology is its ability to handle different types of traffic, like voice, video, multimedia, and LAN traffic in a single ATM network instead of in separate and incompatible networks.

When the cells are routing through ATM switches, a problem that occurs in all ATM switches is how to handle the situations if the cells arriving at two and more input lines want to go to the same output port simultaneously. The switch fabric is responsible for establishing paths between input and output ports as well as implementing a mechanism which will avoid or minimize internal blocking, output blocking and at the same time increase the throughput.

This thesis work is a comparative study of different ATM switching fabrics, including Starlite, Moonshine, Sunshine Switching, and other currently posted research works. The thesis also analyzes the performances of certain ATM switching fabric under different traffics, and internal and external structures. Through the comparative study, we can see that in order to determine a suitable ATM switching fabric, we must consider the T

•

CHAPTER II

ATM CONCEPTS AND ARCHITECTURES

2.1 Concepts

ATM is based on a fixed-size virtual circuit-oriented cell-switching methodology. An ATM cell has a 5-byte header field and a 48-byte information field (Figure 1). The header field contains control information for the cell (such as identification, cell loss priority, routing, and switching information). ATM breaks all traffic into these 53-byte fixed-size cells.



Figure 1. An ATM Cell

The reasons why ATM chooses cell-switching technology are:

(1) Flexibility. As the traffic rate for data is variable and the traffic rate for audio or video is constant, cell-switching methods make it possible to handle both of them, while the traditional circuit-switching methods can not.

- (2) Medium. It is easier to use fiber optics at the very high speed for cell-switching methods compared to the traditional multiplexing methods.
- (3) Broadcasting. Cell-switching methods can provide broadcasting, while traditional circuit-switching methods cannot.

The virtual path identifier (VPI) and the virtual channel identifier (VCI) are two important subfields in the header field. The VPI is used to establish virtual paths semipermanently between networking ports; the VCI is used to establish virtual links over a certain virtual path connection. The combination of these two identifiers is a unique label to determine which cells belong to any given connection due to the connection-oriented feature of ATM.

The VPI (8 bits) allows up to 256 (2^8) virtual paths. Each virtual path contains a number of virtual channels. The VCI (16 bits) allows up to 65536 (2^{16}) virtual channels in one virtual path. The VPI and VCI values are used by the routing protocol to determine the path(s) and channel(s) that a cell will route. As a result, cells belonging to the same virtual channel have the same VPI and VCI, while cells belonging to the same virtual path but different virtual channels have the same VPI and different VCIs. Two networking endpoints use a certain virtual path to make a connection and use different virtual channels to multiplex many individual application streams. VCI is used as a unique identifier to distinguish between these streams.

ATM can efficiently utilize the network bandwidth by dynamically distributing among a variety of user applications. Users may specify their peak and average data rates, and maximum burst size in advance. By using this information, ATM can select virtual channel path according to the expected traffic requests and allocate the bandwidth.

2.2 Architectures

ATM reference model consists of three layers, the physical layer, ATM layer, and ATM adaptation layer [13], plus whatever the users want to put on top of that (Figure 2).



Figure 2. ATM Reference Model [13]

The physical layer deals with the physical medium: voltages, bit timing, and various other issues. It provides the network interfaces and puts the bits on the electrical or optical mediums. This layer consists of two sublayers, the Physical Medium Dependent (PMD) sublayer and the Transmission Convergence (TC) sublayer. The PMD sublayer is the lower of the two sublayers. Its functions depend on the physical medium like bit timing and line coding. The TC sublayer converts bit stream passed from PMD into cell stream and sends them to the ATM layer. Its functions are independent of the physical medium.

The ATM layer takes the payload sent by the ATM adaptation layer and adds fivebyte header information to form a cell. The header check-sum field is used to protect the header field from transmission errors and ensures that the cells will be sent on the right connection. Congestion control is also located here to assign different cell loss priorities to different types of traffics.

The ATM adaptation layer provides a bridge between the services required by higher network layers and the generic ATM cells used by the ATM layer. It segments all data types which are larger than a cell into 48-byte units, then transmits them to the ATM layer, and reassembles them at the other end when cells are passed from the ATM layer. It consists of two sublayers, the Segmentation And Reassembly (SAR) sublayer and the Convergence Sublayer (CS) sublayer. The CS sublayer provides the standard interface, the SAR sublayer provides the segmentation and reassembly. The ATM adaptation layer also plays an important role in the internetworking of different networks and services.

CHAPTER III

ATM SWITCHES

3.1 Attributes

An ATM switch consists of a set of N input and N output ports, a switch fabric, and a management and control processor (MCP).

A switching element is the basic unit of switch fabric (Figure 3). It consists of an interconnection network, an input controller (IPC) for each incoming line, and an output controller (OPC) for each outgoing line.



Figure 3. A Generic ATM Switch [13]

The switch fabric is responsible for establishing paths between input and output ports, routing the incoming cells from input ports and then passing them to the correct output ports as well as preventing excessive cell loss caused by internal or output blocking. It uses hardware and software as tools to handle the following issues:

(1) Establishing a connection between input port and output ports.

(2) Implementing buffer management.

(3) Handling the internal blocking.

(4) Supporting multiple-path connection between input and output ports.

The function for the MCP is to communicate with IPC and OPC, and make the switch operation, administration, and management easily [12].

3.2 Classification

According to the different attributes, switching architectures have several classifications. For example, one classification is strictly based on the internal structure of the switching fabric; but another classification may focus on the applications of ATM as the corporate backbone. In general, the goal to transport traffic between input and output ports within the switch fabric can be implemented by using time-division or space-division methodology.

3.2.1 Time-Division Switch

In a time-division methodology, the information and time slots are multiplexed among several input-output connections, based on discrete time slots. For example, a memory module is the repository of cells supplied by input ports that are removed by output ports. The shared memory switch provides a common memory to store the cells.

Efficient implementation of shared memory strategy requires a multiported memory with up to 2N ports for N ports. Due to this linear behavior, time-division fabrics are competitive when increasing the number of ports. As a multicasting cell is stored in one shared memory, it is easy to implement multicasting connections with little additional complexity.

There are several limitations with this design. One is that shared memory is very difficult to support a large number of high-delay WAN connections that are expected to be mostly or fully loaded. The second is that shared memory is not scalability due to the limitation in the speed of memory.

3.2.2 Space-Division Switch

In a space-division methodology, the switch fabric can support single or multiple paths at a given point between input and output ports. The connections are based on the available non-conflicting physical paths within the fabric. The arrangement of simple switching elements in a crossbar (Figure 4) is the primary example of the space-division implementation [12]. Space-division architecture supports multicasting through either faster interfaces or multiple stages. Supporters of space-division fabrics concentrate on suitability and scalability.

There are self-routing and label routing with the space-division architecture. In the self-routing design, the information in the header field of the cell can be interpreted by the switch fabric and be routed to the output port. This design can provide a high degree

of parallelism. The most common example of this design is Banyan network. In the label routing design, the VCI field in the cell header is used for routing decisions.

Problem are shown if a high-speed input link is routing to a combination of low-speed and high-speed outputs. Congestion at the low-speed output may lead to starvation at the input, the other high-speed outputs also suffer the starvation at the same time. As a result, internal speedup mechanisms must be implemented to minimize contention. Another problem for all ATM switches is how to handle the internal and output blocking. Under this situation, more than one cell will passed through the same internal switch element or arrive at the same output port at the same time slot.

The above problems can be avoided by sorting cells according to their destinations before putting them into the switch fabric. One well-known sorting network is the Batcher network (see 4.2).

We can also put buffers into places where congestion may occur. One of the most basic issues in good buffer design is to avoid HOL blocking problems associated with FIFO buffers. This is a situation where cells destined to multiple outputs may be held up by a cell destined to a congested port. A method to avoid this is to establish a window mechanism where cells behind the first in line are examined for destination.

How to design a non-blocking ATM switch, avoid HOL blocking, and increase the throughput has been a research topic over the past few years.

Whether a single-stage or multistage design is more scalable, is still under research. Although the multistage design has been the preferred option for the scalability beyond 20-60 Gbps in the past years, recent developments have allow us to implement singlestage switches easily. Multistage design may further complicate design and introduce delay when supporting a large number of mid-bandwidth interfaces, while at same time a high-performance single-stage architecture is enough to implement. Due to these reasons, the future large ATM switch may be based on single-stage designs.

For the future ATM switch design, one possible solution is to implement timedivision technologies at the interfaces with the space-division structures as the core of the switch. The recent posted design shown in 6.2 is under this idea.

CHAPTER IV

REPRESENTATIVE ARCHITECHTURES OF ATM SWITCHING FABRICS

This chapter illustrates the representative architectures of the ATM switching fabrics, like Knockout switch, and multistage switch. We can see how these switching fabrics handle the blocking problems and how the Batcher-Banyan network works.

4.1 Knockout Switch

The Knockout switch is a matrix architecture consisting of N (N = 8) broadcasting buses (Figure 4) [13]. The bus interfaced at input port *i* can transmit to output ports





numbered 0 through N-1. It allows N input-output pairs to be connected simultaneously. The switch has a concentrator at each output port and N filters. When multiple cells go to the same output ports or one cell goes to several output ports, the filters filter the desired packets off the buses that are destined for the current output port, while the concentrator with R buffers (R < N) implements a selection algorithm for selecting R cells out of the possible maximum N arriving at that port.

Although the Knockout switch is a non-blocking, single-path architecture, the problem with the Knockout switch is that it is basically a crossbar switch which consists of N^2 cross-points and the size of such switch is limited. This self-routing architecture is not scalable.

4.2 Multistage Switch

Batcher-Banyan switch is the solution to reduce the number of cross-points based on the multistage interconnection networks.

In Figure 5, we have an 8 X 8 three-stage Banyan switch. It can be constructed by interconnecting stages using a mechanism known as Perfect-Shuffle. In order to route one cell from input port to the output port, a routing tag containing the address of the destination is used to route cells in each stage of the switching fabric. The Banyan switch interprets the routing tag from left to right, so stage 1 examines the leftmost bit, stage 2 examines the middle bit, and stage 3 examines the rightmost bit.

Internal and output blocking may occur when two incoming cells want to exit a switching element via the same port at the same time (Figure 6). Depending on the input,



Figure 5. The Banyan Network (without blocking) [12]



Figure 6. The Banyan Network (with blocking) [12]

the Banyan switch can avoid collision, or can not avoid collision. This phenomenon gives us the hint that we can put another switch in front of the Banyan switch to permute the cells so that the Banyan switch can avoid collision.

14

K.E. Batcher (1968) invented the Batcher switch to sort the incoming cells. The Batcher switch is put in front of the Banyan switch to permute the cells so that the Banyan switch can handle the traffic without blocking, this new switch design is called Batcher-Banyan switch (Figure 7). In the example in Figure 7, cells are coming on input lines 2, 3, 4, and 5, going for output lines 6, 5, 1, and 4, respectively. At first, cells for 5 and 6 go to the same switching element. As cell 6 has a higher address than cell 5, so it goes in the direction of the arrow; cell 5 goes the other way, no exchange occurs. For cells 1 and 4, cell 4 goes from the bottom but exits the switch element from the top, an exchange occurs in this case. Internal blocking will not happen in this switch.

In general, the Batcher-Banyan switch is a good ATM switch, but this switch will not eliminate output blocking and handle multicasting, so we have to add additional switch to eliminate the output blocking. By inserting a trap network between the Batcher switch and the Banyan switch, cells with the same address are detected by the trap network and all but one of the duplicated cells will go back to the input sorting networks.



Figure 7. The Batcher-Banyan Network [9]

CHAPTER V

ATM SWITCHING FABRIC DESIGNS AND GOALS

Based on the Batcher-Banyan network, there are Starlite switch, Moonshine switch, and Sunshine switch. They differ primarily in the trap network and how they handle multicast and blocking.

5.1 Starlite Switch Structure

The goal for the Starlite switch structure [6] is to eliminate the output port blocking. It adds the recirculating network to remove output blocking from a Batcher-Banyan architecture. The recirculating network can redistribute cells that lost contention in a given cycle. The recirculated cells will contend for transmission in the next cycle.

In Starlite switch structure (Figure 8), a trap network is inserted between the Batcher and the Banyan network. The trap network examines the output of the sorting network and detects all cells with the same destination address. The duplicated-destination cells will reenter the sorting network to contend in the next cycle. Cells that have to reenter the sorting network will be assigned a higher priority to maintain cell sequence. At the same time, cells entering the Banyan network can be routed to the output port without any internal blocking. Starlite network solves the output blocking problem, but it has the following limitations:

(1) Reentering repeatedly may cause the cells lost or out-of sequence after delivery.

(2) Adding extra trap networks will need more hardware designs.

Due to these drawbacks, we need to design a simpler and more flexible switch structure, the Moonshine switch structure is an alternative approach.



Figure 8. Starlite Switch Structure [7]

5.2 Moonshine Switch Structure

The goal for the Moonshine switch structure [7] is to solve the output blocking problems through sorting and acknowledgements of the winning port before actual cell delivery. This structure uses a three-phase architecture (Figure 9) that includes a contention-resolution phase to eliminate output blocking. In the first phase, input ports send a transmission request to the sorting network. The sorting network sorts the requests based on the destination address. If one of the requests



PHASE I: SEND AND RESOLVE REQUEST

- SEND SOURCE-DESTINATION PAIR THROUGH SORTING NETWORK
- SORT DESTINATION IN NON-DECREASING ORDER
- PURGE ADJACENT REQUESTS WITH SAME DESTINATION



PHASE II: ACKNOWLEDGE WINNING PORT

- SEND ACK WITH DESTINATION TO PORT WINNING CONTENTION
- ROUTE ACK THROUGH BATCHER-BANYAN NETWORK



PHASE III: SEND PACKET

ACKNOWLEDGED PORT SEND PACKET THROUGH BATCHER-BANYAN

Figure 9. Moonshine Switch Structure [7]

competes for a given output port is selected, the others of the duplicated-destination requests are ignored.

In the second phase, the sorting network sends back an acknowledgement to the input port to inform that its request for transmission has won the contention and the cells can be sent without output blocking.

In the third phase, the winning port transmits its cells to the output port.

The three-phase algorithm can avoid data loss, as the data will not be sent until they win the contention and receive the acknowledgement. But this algorithm suffers from Head of Line (HOL) blocking which decreases the overall throughput of the switch [7].

There are several modifications for Moonshine switch structure to reduce the HOL blocking. The first approach is to keep a copy of the sending packets in the input port. Compared to the three-phase algorithm, the packets will be sent to the output port before receiving the acknowledgements. If a certain packet receives the acknowledgement, the copy for this packet will be abandoned. The second approach is to add another paralleling plane of three-phase network. If the HOL blocking occurs in one plane, the packets can try the other planes [7]. The Sunshine switch structure is under the idea of paralleling planes.

5.3 Sunshine Switch Structure

The goal of the Sunshine switch structure [12] (Figure 10) is to remove the output port blocking and increase the number of paths from input to output ports to increase the throughput. The core idea of this switch is to use several planes of Banyan network in parallel.

This structure combines a Batcher sorting network with K parallel Banyan networks for routing. At any cycle, at most K cells routed for the same output port can be delivered. If the number of cells for the same destination is more than K in any one cycle, the excessive cells are stored in a recirculating buffer and transmitted back to the certain input ports for delivering again in the next cycle. The output from the Batcher network consists of cells sorted according to the destination address. The trap network selects at most K cells per destination address to be routed. The remaining cells are separated by the concentrator and forwarded to the recirculation buffer. The selector submits the cells to the Banyan network to be routed. These designs require more complicated IPC and OPC.

The Sunshine switch architecture provides a recirculating queue and output queues in one architecture. This queuing approach is viable and efficient. It is suitable for a wide range of services in the future broadband network. This combined queuing approach reduces the internal buffer requirements, simplifies the interconnection



Figure 10. Sunshine Switch Structure [12]

network, and improves the efficiency by using either one of queuing techniques. The drawbacks for the Sunshine switching structure are that the possibilities for cell loss due to an excessive number of reentering and the out-of-sequenced arriving cells will increase.

Overall, a Banyan network can avoid internal blocking if the inputs are sorted according to the destination requests. In order to sort the incoming cells, a Batcher sorting network based on the sorting algorithm is used [8]. But the Batcher sorting network does not increase the throughput when adding multiple networks like a trap network and a concentrator network to filter out the excessive cells and retransmit them for routing in the next cycle. So the hardware overhead is large in order to achieve the required performance.

CHAPTER VI

RESEARCH AND DEVELOPMENT

The design of ATM switches has evolved in a number of ways over the past few years, but several key issues still need to be resolved. One major issue is scalability and modularity, because many solutions which are efficient for small switches are not scalable to large switches [4].

Current research and development focus on two switch designs – crossbar (spacedivision) and shared-memory (time-division and space-division). Both have advantages and disadvantages, depending upon their intended approaches. We will introduce two examples: tandem-crosspoint switch architecture and scalable shared memory ATM switch architecture. The first approach consists of multiple crossbar switch planes with input and output buffering. The second approach uses shared memory as interfaces and Banyan network as the internal structure fabric, so it is a combination design of spacedivision and time-division.

6.1 Tandem-Crosspoint Switch Architecture

Space-division architectures support multiple connections based on non-conflicting paths within the switch, and may provide a single path or multiple paths between input and output ports.

Space-division switches evolved from Banyan and Batcher-Banyan to more effective high-bandwidth interconnection fabrics with output buffers. Although Banyan based switches were considered attractive because of their low complexity of design, they suffer from poor performance under high traffic loads, burst traffic or unbalanced traffic.

On the other hand, the crossbar switch has the following advantages [11]:

- Modularity and scalability. One can expand existing switch to larger one simply by adding more cross-point switches.
- (2) Less delay. The cell transmission delay in crossbar switches is smaller than in Banyan-based switches due to the smallest number of connecting points between input/output pairs.

Crossbar-based switch consists of input buffering switch and output buffering switch. Due to the deployment of buffering strategy, the HOL blocking problem is the main issue to the design. For example, HOL in the input buffering switch limits the maximum throughput. In order to solve this problem, there are two approaches:

 Increasing the internal line speed of the switch: This approach is not cost-effective for much larger throughputs due to the limitation of current hardware technologies.

23

(2) Employing a parallel switch architecture: The parallel switch consists of K identical switch planes. Each switch plane has its own input buffer but shares output buffers with other planes. This approach requires time stamps which must be placed in each cell header, and cell sequence will be rebuilt at the output buffers. On the other hand, as we combine several parallel planes, the hardware resources needed to implement this approach will be K times that the single plane needs. So this solution is still not cost-effective for much larger switches.

Tandem-crosspoint switch (Figure 11) is a new input and output buffering crossbar switch architecture. This switch architecture avoids to increase the internal line speed and rebuild cell sequences at the output buffers. Also, it uses high-density devices instead of using ultra-high-speed devices because high-density devices relaxes the limitation of hardware technologies.



Figure 11. Tandem-Crosspoint Switch Structure [11]

The structure of the tandem-crosspoint switch is depicted in Figure 11, the number of crossbar switch planes is K in general. These K switch planes can transmit up to K cells to each output port within one time slot. The internal speed in each plane is the same as the input/output line speed. Each switch plane can transmit only one cell to each output port within one time slot. If more than one cell goes to the same output port on the same switch plane, unsuccessful cells that can not be transmitted to the output port are stored in the tandem-crosspoint. If we use multiple output buffers instead of single output buffer and provide priority controls, tandem-crosspoint switch can also allow multiple quality of services.

There are several implementation advantages to the tandem-crosspoint switch:

- (1) Relaxing the limitation of hardware technologies. The switch uses high-density device to avoid increasing the internal line speed and still eliminates the HOL blocking at the same time.
- (2) Employing a simple linear cell algorithm at the input buffer. This switch does not need to add time stamps and rebuild cell sequence at output.
- (3) Using hardware resources more cost-effectively. As it uses high-density device technologies instead of using high-speed device technologies, it does not suffer the limitation of the hardware technologies.

6.2 Scalable Shared Memory ATM Switches

As opposed to almost all other architectures, the shared memory switching allows all inputs and outputs to share a common buffer. Congestion problem at interfaces can be solved by taking advantage of any available buffer at a given cycle. Shared memory switching provides optimal delayed-throughput performance for a given cell-loss rate, and offers a very practical solution for small scale ATM switches. Because of the limitations of the speed of the centralized memory controller and memory access time, pure shared-memory can not provide a scalable solution to ATM switching. This problem can be partially solved by combining shared memory switching with space division switching [1].

The standard Banyan network is known to have a severely blocking problem. We need to modify standard Banyan and combine the shared memory switch to solve this problem. The Truncated Fat-Banyan (TFAB) switching design uses shared memory switch as interface and Banyan switch as a core switch fabric.

The Fat-Banyan (FAB) switching architecture shown in Figure 12 uses incremental bandwidth (dilation) between stages of switching elements. In this switch, each port of a switching element has multiple input and output links. The number of input and output



Figure 12. The Fat-Banyan Topology [1]

links per port may not be equal, so it is called a Fat-Banyan switching element. The goal of this architecture is to expand the internal link bandwidth in order to solve the cell-loss problem caused principally by the insufficient bandwidth in standard Banyan switch.

The TFAB switch shown in Figure 13 is derived from the Fat-Banyan switch by removing one or more switching stages starting from the last (or output) stage and replacing the output buffers by shared-memory switches. The routing in the truncated fabric is similar to the normal self-routing algorithm in Banyan network except that only those routing bits required for routing through the Banyan network are used and the rest of the bits are used by the shared memory modules for buffering and final routing. The TFAB switch has the following advantages:

 Low cost: As TFAB uses shared memory among the output port and employs the existing Banyan network, so it is cost-effective.



Figure 13. The Truncated Fat-Banyan Topology [1]

- (2) Quality-of-Service separation: As TFAB has proper link dilations, the amount of loss affected by streams from other input ports is very small. This quality-ofservice separation is very important to the various streams through the TFAB.
- (3) Multicasting capability: Due to the shared memory among the output ports, it is easy for TFAB to implement the multicasting capability.

CHAPTER VII

PERFORMANCE ANALYSIS OF FAT-BANYAN SWITCH

The performance of an ATM switch depends on the traffic pattern according to which switch cells arrive at its inputs as well as the distribution of the output requests.

7.1 Independent Uniform Traffic Pattern

A performance analysis of the FAB switch model is based on a simple traffic pattern [1], the independent uniform traffic pattern. In this traffic pattern, the arrival process is a Bernoulli process with parameter p at input ports, the arrivals at input ports are independent of each other, and the distribution of the destination requests of the arriving cells is uniform over all the output ports.

A FAB can be characterized by these parameters M, N, and l, where M denotes the total number of input links to each FAB stage, N is the number of output ports, and l is the number of links per output port. Figure 14 depicts Fat-Banyan switching element with the three parameters indicted. For the purpose of the performance analysis, the parametric value of the total number of input links per FAB is sufficient without the need to distinguish between the number of input ports and the number of links per input port.



Figure 14. Fat Switching Element

Under the independent uniform pattern, we define the output rates (per link and per port) as follows (see [1]):

(1) Let P(j | i) be the probability that j cells are destined to a particular output port of a FAB, given that there are i cell arrivals. For independent uniform traffic, each cell arrival has equal probability of being destined to one of N output ports. Then

$$P(j \mid i) = \binom{i}{j} (1/N)^{j} (1 - 1/N)^{i-j}, \text{ where } 0 \le j \le i \text{ and } 0 \le i \le N.$$

(2) For the j cells destined to any output port (with l links), only min(j, l) cells are accepted. The expected number of cells at one output port, given i cell arrivals, is

$$\sum_{j=0}^{i} P(j|i) \min(j,l)$$
. Let $E(i)$ be the expected number of cells that FAB passes

successfully to its outputs per clock cycle, given that *i* cells arrived at the beginning of that cycle. Since there are N output ports, we have E(i) =

$$N\sum_{j=0}^{l} P(j|i) \min(j,l), \text{ where } 0 \leq j \leq i \text{ and } l > 0.$$

(3) Let E be expected number of cells at the output of the FAB per clock cycle. Noting that the cell arrivals follow Bernoulli process with parameter p, $E = \sum_{i=0}^{M} E(i)q(i)$, where q(i) is the probability of *i* arrivals in one clock cycle, that is,

$$q(i) = \binom{M}{i} (p)^{i} (1-p)^{M-i} 0 \le p \le 1, 0 \le i \le M, \text{ and } M > 0$$

(4) The output rate per output link O_l is O_l = E/(N*l), and the output rate per output port is O_p = l*O_l = E/N, note that O_l of current stage output will become the traffic load to the next input stage. We implement the computation to obtain O_p of the final stage.

We performed a simulation (Figure 15) based on the independent uniform traffic model. The parameter p was chosen as 1.0, which means every arrival cell can arrive at



Figure 15. Maximum Throughput for the Fat-Banyan Switch Under Independent Uniform Traffic Pattern

the input port. At the same time, the initial number of input ports M, number of output port N, and the initial dilation factor l were chosen as 2, 2, and 2 respectively. This simulation consists of nine stages. After each stage, M will be increased and M is equal to N * l, then l is incremented by one. In this figure, we can see that the maximum throughput after stage 3 will remain constant even though the number of links still increases. So in the FAB, it is not necessary to keep increasing the degree of dilation to the last stage under the independent uniform traffic pattern.

The pure Dilated Banyan employs fixed dilation of the inputs and outputs of a switching element. Although dilation improves performance of the Banyan network by increasing the internal bandwidth, the internal bandwidth of the switching fabric is not optimally utilized. The FAB switch keeps the dilation variable and achieves a low order of complexity than the pure dilated Banyan [1].

7.2 Poisson Distribution Traffic Pattern

The Fat-Banyan switch shows good performance under independent uniform traffic model, we investigate its throughput performance under Poisson distribution traffic pattern.

Consider the traffic load to the input stage of the switch follows Poisson distribution with parameter λ , where λ is the arrival rate at the input. The possibility of *i* cell arrivals

in one clock cycle is
$$q(i) = \frac{e^{-\lambda}\lambda^i}{i!}$$
.

In order to have every input port receive an incoming cell, we should choose a suitable λ . We plot the Figure 16 to get the relation between λ and the probability of *i* cell

arrivals in one clock cycle, given *i* as 2. From Figure 16, we can see the probability of every input port receiving incoming cell is nearly 1 if the arrival rate λ is large enough. In this case, we chose suitable λ as 20 for our performance analysis under Poisson distribution traffic pattern in Figure 17.



Figure 16. Reasonable Arrival Rate for the Fat-Banyan Switch Under Poisson Traffic Pattern

We plot the maximum throughput map (Figure 17) again by using Poisson distribution as the input traffic pattern instead of independent uniform traffic pattern. The parameter λ was chose as 20. In this simulation, we still chose the initial number of input ports *M*, number of output port *N*, and the initial dilation factor *l* as 2, 2, and 2 respectively with nine stages, which are the same as those of the simulation under independence uniform traffic pattern. After each stage, *M* will still be increased and *M* is equal to N * l, then *l* is incremented by one. The only difference is the traffic pattern in this simulation. From Figure 17, we can see that the curve trend under Poisson

distribution is nearly the same as that under independent uniform traffic pattern. After stage 3, the maximum throughput of each stage becomes stable no matter you keep the dilation degree. That means the FAB switching fabric is still useful under Poisson distribution traffic pattern.



Figure 17. Maximum Throughput for the Fat-Banyan Switch Under Poisson Traffic Pattern

7.3 Analysis of Fat-Banyan Switch

Figures 15 and 17 are based on the simulation for M (number of input ports) = N (number of output ports) = 2. We study the empirical relations between some combinations of (M, N, l) and the maximum throughput.

We plot the Figure 18 to find out the relation between the number of ports and the maximum throughput when the initial dilation factor l is 2. In this simulation, we still chose the independent uniform traffic pattern as our input traffic pattern, but chose different number of initial M and N through several stages. From this figure, it can be

seen that the maximum throughput decreases when the number of ports increases. To increase the number of ports is to increase the number of incoming cells in a certain clock cycle. That means we need more internal bandwidth when we have more incoming cells from the input ports. So we have to find out if there is relation between the dilation factor and the maximum throughput under a certain number of ports.



Figure 18. The Relation Between the Number of Port and the Maximum Throughput (initial dilation factor l = 2)

From Figure 18, we can assume that the maximum throughput will be very lower if the number of ports is 10. So we chose M and N as 10 to illustrate the improvement (Figure 19) if we increase the initial dilation l to get more internal bandwidth. In this simulation, we still chose the independent uniform traffic pattern.



Figure 19. The Relation Between the Dilation and the Maximum Throughput (number of input ports M = number of output ports N = 10)

From Figure 19, we can see that if we increase the initial l, the maximum throughput will improve a lot. That means the dilation factor l is very important in the FAB switching design under a certain number of input ports and output ports.

In Figures 18 and 19, the curve trends are almost the same. That means FAB is a good strategy to make the dilation variable after the maximum throughput becomes stable.

Overall, for a small number of ports, we can choose a small dilation l and get good maximum throughput; if there are larger number of ports, we can choose the larger initial dilation l so that the throughput will be well maximized.

CHAPTER VIII

CONCLUSIONS, DISCUSSION AND FUTURE WORK

According to the comparative study reported in this thesis, we can see that employing small shared-memory switching has several advantages. First, by taking advantage of commercially available memory technologies, we can design ATM switches by determining suitable shared-memory module size, and specifying a proper interconnection among the modules. Due to this modularity feature, switch architectures can be reusable and able to evolve with advanced technologies. Second, shared memory increases the utilization of buffer space and allows the implementation of flexible and fair buffer-allocation algorithms for multiple applications. The limitation in the speed of memory leads to a large number of high-delay WAN connections and affects the scalability of shared memory design.

Space-division switches evolved from Banyan and Batcher-Banyan types of architectures to more effective high-bandwidth interconnection fabrics with output buffers. Space-division architectures support multiple connections based on nonconflicting paths within the switch, and can provide a single path or multiple paths between input and output ports. In order to avoid the HOL blocking, one can effectively

37

use buffering strategies and increase the utilization, which is still the main design issue in space-division switching.

Buffer strategy is another important issue related to the complexity and cost of an ATM switch, especially when the switch needs to corporate with the buffer management to avoid the congestion and HOL blocking. This involves several key design considerations:

- (1) Minimizing the number of buffer stages to reduce cell delay.
- (2) Utilizing the buffer sharing as much as possible in order to reduce total buffer amount.
- (3) Applying efficient control algorithm for the shared buffer management.
- (4) Maintaining the self-routing interconnection fabrics and non-blocking feature under different traffics, like unbalanced traffic.

Traffic pattern is still a key issue to evaluate the design of ATM switches. Recent work based on actual LAN traffic measurements indicates that traditional traffic models, like Poisson traffic and uniform traffic may be overly optimistic and the LAN traffic measured is self-similar, which means that the traffic has similar properties regardless of the time scale on which it is observed [4]. The feature of the self-similar traffic is that the burst length ranges from few milliseconds to minutes or hours, so there is no natural length in this traffic. This is in sharp contrast to the Poisson models, where the traffic tends to become smoother and more predictable when considering longer and longer time average.

The Ethernet local area network traffic is statistically self-similar, that none of the commonly used traffic models is able to capture this fractal behavior. As the LAN interconnection services have potentially big market, understanding the nature of selfsimilar behavior will be useful for the design, control, and analysis of high-speed, cellbased networks. We used conventional traffic models to analyze the throughput performance in the previous chapters. But this management may perform less satisfactorily in a self-similar traffic environment. So the future work will focus on the performance analysis under self-similarity traffic pattern, especially for the LAN.

Overall, in order to determine suitable ATM switching fabric, we must consider the following aspects: scalability, congestion free, efficient use of bandwidth, reliability, traffic pattern and buffer management.

REFERENCES

- Alimuddin, M., and Alnuweiri, H. M., "Design and Evaluation of Scalable Shared-Memory ATM Switches," *IEICE Transactions on Communications*, Vol. E81-B, No. 2, pp. 224-236, Feb. 1998.
- [2] Eckberg, A.E, Doshi, B.T., and Zoccolillo, R., "Controlling Congestion in B-ISDN/ATM: Issues and Trategies," *IEEE Commun. Magazine*, Vol. 29, No. 9, pp. 64-70, Sept. 1991.
- [3] Eckberg, A.E., "B-ISDN/ATM Traffic and Congestion Control," IEEE Network Magazine, Vol. 6, No. 5, pp. 28-37, Sept./Oct. 1992.
- [4] Ginsburg, D., ATM Solutions for Enterprise Internetworking, 2nd ed., Addison-Wesley, Reading, MA, 1999.
- [5] Handel, R., Huber, M.N., and Schroder, S., ATM Concepts, Protocols, and Applications, 2nd ed., Addison-Wesley, Reading, MA, 1994.
- [6] Huang, A. and Knaur, S., "Starlite: A Wideband Digital Switch," Proc. GlobeCom 84, IEEE Press, Piscataway, N.J., pp. 121-125, 1984.
- [7] Hui, J. and Arthurs, E., " A Broadband Packet Switch for Integrated Transport," IEEE J. Selected Areas Comm., Vol. SAC5, No. 8, pp. 1264-1273 Oct. 1987.
- [8] Hong, D., and Suda, T., "Congestion Control and Prevention in ATM Networks," IEEE Network Magazine, Vol. 5, No. 4, pp. 1016, July/Aug. 1991.
- [9] Giacopelli, J. et al., "Sunshine: a High-performance Self-Routing Broadband Packet Switch Architecture," *IEEE J. Selected Areas Comm.*, Vol. 9, No. 8, pp.1289-1298, Oct. 1991.
- [10] Kim, B.G. and Wang, P., "ATM Network: Goals and Challenges," Communications of the ACM, Vol. 38, No. 2, pp. 39-44, Feb. 1995.
- [11] Oki, E., and Yamanaka, N., "A High-Speed Tandem-Crosspoint ATM Switch Architecture with Input and Output Buffers," *IEICE Transactions on Communications*, Vol. E81-B, No. 2, pp. 215-223, Feb. 1998.

- [12] Rooholamini, Reza and Cherkassky, Vladimir, "Finding the Right ATM Switch for the Market," IEEE Computer, Vol. 25, No. 4, pp. 16-28, Apr. 1994.
- [13] Tanenbaum, Andrew S., Computer Networks, Prentice-Hall International, Inc., Upper Saddle River, New Jersey, 1996.
- [14] Vetter, Ronald J., "ATM Concepts, Architectures, and Protocols," Communications of the ACM, Vol. 38, No. 2, pp. 30-38, Feb. 1995.
- [15] Yeh, Y. S. and Hluchyj, M.G., "The Knockout Switch: A Simple, Modular Architecture for High-performance Packet Switching," *IEEE Journal on Selected Areas in Commun.*, Vol. SAC5, No. 8, pp. 1274-1283, Oct. 1987.

APPENDIX A: GLOSSARY

- ATM: Asynchronous Transfer Mode. This mode can be contrasted with the synchronous T1 carrier whose frame rate is governed by a master clock.
- B-ISDN: Broadband Integrated Services Digital Network. It will offer video on demand, live television from many sources, full motion multimedia electronic mail, CD-quality music, LAN interconnection, high-speed data transport for science and industry and many other services that have not yet been thought of, all over the telephone line.
- CD: Compact Disk. Music stored in CD has higher quality.
- CS: Convergence Sublayer. The CS sublayer provides the standard interface (convergence).
- FAB: Fat Banyan network. In this switch, each port of a switching element has multiple input and output links. The number of input and output links per port may not be equal.
- FIFO: First-In-First-Out. This is a situation where cell first enters the input buffer and will first leave the buffer.
- Gbps: Gigabit Per Second. It is a transmission speed rate in network
- HOL Blocking Head of Line Blocking. The problem with input queuing is that when a cell has to be held up, it blocks the progress of any cells behind it, even if they could otherwise be switched. This effect is called HOL blocking.
- IPC: Input Port Controller. It manages the input ports. IPC terminates the transmission links and formats the ATM cells for transport with the switch by inserting a control header in front of each cell. The control header identifies the destination port on the switch and the service priority for the cell being transported.
- LAN: Local Area Networks. They are privately owned networks within a single building or campus of up to a few kilometers in size.

MCP:	Management and Control Processor. The MCP's function is to communicate with port controllers and facilitate switch operation, administration, and management.
OPC:	Output Port Controller. It manages the output ports. Each OPC connects to each Banyan network at the same Banyan port address and contains the output queue. The OPC services these buffers in order of priority and formats cells for transmission.
PMD:	Physical Medium Dependent. The PMD sublayer interfaces to the actual cable. It moves the bits on and off and handles the bit timing.
SAR:	Segmentation And Reassembly. The SAR sublayer is the lower sublayer of the ATM Adaptation Layer. It breaks packets up into cells on the transmission side and puts them back together again at the destination.
TC:	Transmission Convergence. The TC sublayer handles all the issues related to telling where cells begin and end in the bit stream.
TFAB:	The Truncated Fat-Banyan (TFAB) topology is derived from the Fat-Banyan topology by removing one or more switching stages starting from the last (or output) stage and replacing the output buffers by shared-memory switches.
VCI:	Virtual Channel Identifier. VCI is used to establish virtual links over a given virtual path connection.
VPI:	Virtual Path Identifier. VPI is used to establish virtual paths on a semipermanent basis between network endpoints.
WAN:	Wide Area Network. It spans a large geographical area, often a country or continent.

APPENDIX B: PROGRAM LISTINGS

```
// This program is used to get maximum throughput for
// each stage inside Banyan network under independent
// uniform traffic and Poisson traffic
#include <cmath>
#include <iostream>
#include <iomanip>
#define e 2.7181
double factorial ( int );
int min (int, int);
double P (int, int, int);
double Q_Poisson (int, int, double);
double Q_Uniform (int, int, double);
double E (int, int, int);
double ME_Poisson (int, int, int, double);
double ME_Uniform (int, int, int, double);
int main () {
      // Uniform Traffic
      int stage = 1; // stage in Banyan network
      int M = 2; // number of input links
      int N = 2; // number of output ports
      int 1 = 2; // dilation factor
      double p = 1; // load factor
      cout<<"Under uniform traffic: "<<endl;</pre>
      while ( stage < 10 ) {
          cout<<endl;
          cout<<"stage"<<stage<<endl;
          cout<<"E: "<<ME_Uniform(N,l,M,p)<<endl;</pre>
          cout<<"0: "<<ME_Uniform(N, 1, M, p) /N<<endl;</pre>
          p = ME_Uniform(N, l, M, p) / (N*l);
          cout<<"p: "<<p<<endl;
          M = N*1;
          1++;
          stage++; // next stage
      }
      // Poisson Traffic
      stage = 1;
      M = 2;
      N = 2;
      1 = 2;
      double u = 20.0; // arrival rate
      cout << endl;
      cout << "Under Poisson traffic: " << endl;
      while ( stage < 10 ) {
          cout << endl;
          cout<<"stage"<<stage<<endl;
          cout << " ----- " << endl;
          cout<<"E: "<<ME_Poisson(N, 1, M, u)<<endl;</pre>
          cout<<"O: "<<ME_Poisson(N,l,M,u)/N<<endl;
```

```
M = N*1;
          1++;
          stage++;
      }
)
// calculate the factorial of the number
double factorial ( int a ) {
      if (a <= 1)
           return 1;
      else
           return a*factorial(a-1);
}
// calculate the minimum of two numbers
int min ( int a, int b) {
      if (a \le b)
          return a;
      else
          return b;
}
// P is the probability that j cells are destined to a particular
// output port of a FSE, given that there were i cells arrivals.
double P (int i, int j, int N) (
      return (factorial(i)/factorial(i-j)/factorial(j))
             *pow((float)1/N, j)
             *pow(((float)1-(float)1/N),(i-j));
}
// The expected number of cells that FSE passes successfully to its
// outputs per clock cycle given that i cells arrived at the beginning
// of that cycle.
double E (int i, int l, int N) {
      double sum = 0;
      for (int j = 0; j \le i; j++) {
            sum = sum + P(i, j, N) * min(j, 1);
      }
      return sum*N;
}
// The mean number of cells at the output under Uniform traffic
double ME_Uniform (int N, int 1, int M, double p) (
      double sum = 0;
      for ( int i = 0; i <= M; i++ ) {
            sum = sum + E(i,l,N)*Q_Uniform(i, M, p);
      3
      return sum;
}
// The mean number of cells at the output under Poisson traffic
double ME_Poisson (int N, int l, int M, double u) {
        double sum = 0;
        for ( int i = 0; i <= M; i++ ) {
                sum = sum + E(i,l,N)*Q_Poisson(i, M,u);
        }
        return sum;
}
// Q_Poisson is the probability of i arrivals in one clock cycle
```

T

```
// under the Poisson traffic pattern
double Q_Poisson (int i, int M, double u) {
    return pow(e,-u)*pow(u,i)/factorial(i);
}
// Q_Uniform is the probability of i arrivals in one clock cycle
// under the uniform traffic
double Q_Uniform (int i, int M, double p) {
    return (factorial(M)/factorial(M-i)/factorial(i))*pow(p, i)
                         *pow(((float)1 - p),(M-i));
```

}

46

```
// This program is used to get the relation between the number
// of ports and the maximum throughput for a certain dilation
// under independent uniform traffic
#include <cmath>
#include <iostream>
#include <iomanip>
#define e 2.7181
double factorial ( int );
int min (int, int);
double P (int, int, int);
double Q_Uniform (int, int, double);
double E (int, int, int);
double ME_Uniform (int, int, int, double);
int main () {
      // Uniform Traffic
      int stage = 1; // stage in Banyan network
      int M = 2; // number of input links
      int N = 2; // number of output ports
      int 1 = 2; // dilation factor
      double p = 1; // load factor
      // choose different number of ports
      for (int i = 2; i <= 5; i++) {
            N = i;
            M = i;
            cout<<"M = N = "<<i<<endl;
            while ( stage < 10 ) {
                  cout << endl;
                  cout<<"stage"<<stage<<endl;
                  cout<<"E: "<<ME_Uniform(N, l, M, p)<<endl;</pre>
                  cout<<"0: "<<ME_Uniform(N,l,M,p)/N<<endl;</pre>
                  p = ME_Uniform(N, 1, M, p)/(N*1);
                  M = N*1; // dilate the input links
                           // dilation factor incremented by one
                  1++;
                  stage++; // next stage
            }
      }
}
// calculate the factorial of the number
double factorial ( int a ) {
      if ( a <= 1 )
           return 1;
      else
           return a*factorial(a-1);
}
// calculate the minimum of two numbers
int min ( int a, int b) {
      if ( a <= b )
          return a;
      else
          return b:
}
// P is the probability that j cells are destined to a particular
// output port of a FSE, given that there were i cells arrivals.
double P ( int i, int j, int N ) {
      return (factorial(i)/factorial(i-j)/factorial(j))
```

```
*pow((float)1/N, j)
             *pow(((float)1-(float)1/N),(i-j));
}
// The expected number of cells that FSE passes successfully to its
// outputs per clock cycle given that i cells arrived at the beginning
// of that cycle.
double E (int i, int l, int N) {
      double sum = 0;
      for (int j = 0; j <= i; j++) {
            sum = sum + P(i, j, N) * min(j, 1);
      }
      return sum*N;
)
// The mean number of cells at the output under Uniform traffic
double ME_Uniform (int N, int 1, int M, double p) {
      double sum = 0;
      for ( int i = 0; i <= M; i++ ) {
            sum = sum + E(i,l,N)*Q_Uniform(i, M, p);
      }
      return sum;
}
// Q_Uniform is the probability of i arrivals in one clock cycle
// under the uniform traffic
double Q_Uniform (int i, int M, double p) {
         return (factorial(M)/factorial(M-i)/factorial(i))*pow(p, i)
                *pow(((float)1 - p),(M-i));
}
```

```
// This program is used to get the relation between the dilation
// and the maximum throughput for a certain number of ports
// under independent uniform traffic
#include <cmath>
#include <iostream>
#include <iomanip>
#define e 2.7181
double factorial ( int );
int min (int, int);
double P (int, int, int);
double O_Uniform (int, int, double);
double E (int, int, int);
double ME_Uniform (int, int, int, double);
int main () {
      // Uniform Traffic
      int stage = 1; // stage in Banyan network
      int M = 2; // number of input links
      int N = 2; // number of output ports
      int l = 2; // dilation factor
      double p = 1; // load factor
      // choose different number of ports
      for (int i = 2; i <= 5; i++) {
            1 = i;
            cout<<"l = "<<i<<endl;
            while ( stage < 10 ) {
                  cout << endl;
                  cout<<"stage"<<stage<<endl;</pre>
                  cout<<"E: "<<ME_Uniform(N,l,M,p)<<endl;</pre>
                  cout<<"0: "<<ME_Uniform(N,1,M,p)/N<<endl;</pre>
                  p = ME_Uniform(N, 1, M, p) / (N*1);
                  M = N*1; // dilate the input links
                  1++;
                           // dilation factor incremented by one
                  stage++; // next stage
            }
      }
}
// calculate the factorial of the number
double factorial ( int a ) (
      if ( a <= 1 )
           return 1;
      else
           return a*factorial(a-1);
1
// calculate the minimum of two numbers
int min ( int a, int b) {
      if(a <= b)
          return a;
      else
          return b;
}
// P is the probability that j cells are destined to a particular
// output port of a FSE, given that there were i cells arrivals.
double P ( int i, int j, int N ) {
      return (factorial(i)/factorial(i-j)/factorial(j))
             *pow((float)1/N, j)
```

```
*pow(((float)1-(float)1/N),(i-j));
}
// The expected number of cells that FSE passes successfully to its
// outputs per clock cycle given that i cells arrived at the beginning
// of that cycle.
double E (int i, int 1, int N) {
      double sum = 0;
      for (int j = 0; j <= i; j++) {
            sum = sum + P(i, j, N) * min(j, 1);
      }
      return sum*N;
}
// The mean number of cells at the output under Uniform traffic
double ME_Uniform (int N, int l, int M, double p) {
      double sum = 0;
      for ( int i = 0; i <= M; i++ ) {
            sum = sum + E(i,l,N)*Q_Uniform(i, M, p);
      }
      return sum;
}
// O Uniform is the probability of i arrivals in one clock cycle
// under the uniform traffic
double Q_Uniform (int i, int M, double p) {
         return (factorial(M)/factorial(M-i)/factorial(i))*pow(p, i)
                *pow(((float)1 - p),(M-i));
}
```

VITA

Wei Yan

Candidate for the Degree of

Master of Science

Thesis: A COMPARATIVE STUDY OF ATM SWITCHING FABRICS

Major Field: Computer Science

Biographical:

- Personal Data: Born in Tianjin, China, on September 7, 1968, son of Qichang Yan and Xiuhong Zhang.
- Education: Graduated from No. 20 High School, Tianjin, China in June 1986; received Bachelor of Science degree in Microbiology from Nankai University, Tianjin, China in July 1990. Completed the requirements for Master of Science degree in Computer Science at the Computer Science Department at Oklahoma State University in May 2000
- Experience: Teaching Assistant at the Computer Science Department at Oklahoma State University, January 1998 to May 1999; Research Assistant at the Bioinformatics Laboratory at Oklahoma State University, April 1999 to August 1999.