INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Switching and Error Recovery in Terabit ATM Networks

by

Amr Gaber Sabaa
Master of Electrical Engineering, Cairo University, 1993
Bachelor of Electrical Engineering, Cairo University, 1991

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy of Applied Science in Electrical Engineering

in the Department of
Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

Dr. Fayez ElGuibaly, Supervisor (Electrical and Computer Engineering)

Dr. Dale Shpak, Co-supervisor (Electrical and Computer Engineering)

Dr. Ashoka Bhat, Departmental Member (Electrical and Computer Engineering)

Dr. Peter Driessen, Departmental Member (Electrical and Computer Engineering)

Dr. Eric G. Manning, Outside Member (Computer Science)

Dr. Anwarul Hasan, External Examiner (University of Waterloo)

Supervisor: Dr. Fayez ElGuibaly, Co-supervisor: Dr. Dale Shpak

ABSTRACT

This thesis addresses two of the main issues required to build reliable terabit ATM networks. A high-capacity switch and an efficient error recovery protocol are the key elements in building a reliable terabit ATM network. In this thesis, a terabit switch architecture and a reliable end-to-end error recovery protocol for terabit networks are introduced.

The proposed terabit ATM switch architecture is designed to work efficiently in low-capacity and high-capacity environments. The architecture is developed by interconnecting small-capacity switching modules in a scalable fashion. The switching module can be used alone as a small-capacity ATM switch. Multiple the switching modules can be used to achieve any required switching capacity. The proposed interconnecting scheme provides remarkable low cell-delay characteristics with a simple distributed cell scheduler. The proposed architecture has a high reliability: Even when a complete switching module fails the switch will continue to work efficiently.

The switching element which is introduced as the main building block for the terabit switch architecture is a nonblocking input buffer ATM switch. The input buffers are implemented as groups of parallel shift-registers. The parallel nature of the storing buffers overcomes the Head Of Line and low throughput problems of existing input buffer switch architectures. In addition, using the shift registers overcomes the need for serial-to-parallel and parallel-to-serial format conversions.

ATM networks support different types of services having different delay and loss requirements. A priority scheduling scheme is proposed to facilitate the support of different Qualities of Service. The proposed scheme satisfies both real-time and non-real-time service requirements.

Cell loss is not acceptable for some data applications. This thesis proposes an efficient error recovery protocol which guarantees reliable communication with limited overhead. The proposed protocol requires a low number of control packets to achieve reliable com-

munication. It also adapts itself, in order to work efficiently during both congested and non-congested states.

Examiners:

Dr. Fayez ElGuibaly (Electrical and Computer Engineering)

Dr. Dale Shpak (Electrical and Computer Engineering)

Dr. Ashoka Bhat (Electrical and Computer Engineering)

Dr. Peter Driessen (Electrical and Computer Engineering)

Dr. Eric G. Manning (Computer Science)

Dr. Anwarul Hasan (University of Waterloo)

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. F. ElGuibaly, for his continuous support and encouragement and for giving lots of his precious time to supervise this research work and the process of writing this manuscript. Sincere thanks to Dr. D. Shpak for his support, encouragement, and advice.

Thanks to Dr. E. Manning for his participation and his helpful comments.

I would like to thank my colleague, Hani ElGuibaly, for stimulating useful discussions.

I am grateful to my family, especially my parents, for their love, patience, and countless encouraging words.

To
My Family

List of Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACK	Acknowledge
ADPCM	Adaptive Differential Pulse Code Modulation
ANSI	American National Standards Institute
ARQ	Automated Repeat reQuest
ATD	Asynchronous Time Division
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BER	Bit Error Rate
bps	Bits Per Second
BW	Bandwidth
CAC	Connection Admission Control
CAM	Content Addressable Memory
CBR	Constant Bit Rate
CBS	Cell Buffer Status
CDT	Cell Delay Tolerance
CDV	Cell Delay Variation
CER	Cell Error Ratio
CI	Congestion Indicator
CLP	Cell Loss Priority
CLR	Cell Loss Ratio

CM	Communication Module
CN	Copy Network
CRC	Cyclic Redundancy Check
CS	Convergence Sublayer
EFCI	Explicit Forward Congestion Indication
FCFS	First-Come-First-Serve
FDDI	Fiber Distributed Data Interface
FEC	Forward Error Correction
FIFO	First-In-First-Out
FM	Forward Monitoring
FRS	Frame Relay Service
GBN	Go-Back-N
HDLC	High Level Data Link Control
HEC	Header Error Check
HOL	Head of Line
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Standards Organization
ITU	International Telecommunications Union
LAN	Local Area Network
MAC	Medium Access Control
MAN	Metropolitan Area Network
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
NACK	Negative Acknowledge
OA&M	Operations, Administration and Maintenance

OSI	Open Systems Interconnection
PAD	Packet Assembler and Disassembler
PCM	Pulse Code Modulation
PCR	Peak Cell Rate
PMD	Physical Medium Dependent
PDU	Packet Data Unit
PHY	Physical Layer
PSRP	Periodic Selective Repeat Protocol
PT	Payload Type
PTI	Payload Type Identifier
PVC	Permanent Virtual Circuit (ATM)
PVCC	Permanent Virtual Channel Connection (ATM)
PVPC	Permanent Virtual Path Connection (ATM)
QOS/QoS	Quality of Service
RDI	Remote Defect Indication
RM	Resource Management
SAP	Service Access Point (B-ISDN)
SAR	Segmentation and Reassembly
SCCP	Signaling Connection and Control Part
SCM	Switching Controller Module
SDS	Space Division Switching
SDU	Service Data Unit
SE	Switching Element
SM	Switching Module
SS	Schedule Status
SRP	Selective Repeat Protocol

SSCF	Service Specific Coordination Function
SSCOP	Service Specific Connection Oriented Protocol
SSCS	Service Specific Convergence Sublayer (ATM)
SVC	Switched Virtual Circuit (ATM)
SVP	Switched Virtual Path (ATM)
TC	Transmission Convergence
TCP	Transmission Control Protocol (Internet)
TCS	Transmission Convergence Sublayer (ATM)
TDS	Time Division Switching
TM	Traffic Management
TS	Time Slot
UBR	Unspecified Bit Rate (ATM)
UNI	User Network Interface (B-ISDN)
UPC	Usage Parameter Control
VBR	Variable Bit Rate (ATM)
VC	Virtual Channel (Virtual Circuit) (ATM)
VCC	Virtual Channel Connections (ATM)
VCI	Virtual Channel Identifier (ATM)
VP	Virtual Path (ATM)
VP/VC	Virtual Path, Virtual Circuit (ATM)
VPC	Virtual Path Connection (ATM)
VPI	Virtual Path Identifier (ATM)
WAN	Wide Area Network

Chapter 1

Introduction

Most existing telecommunication networks are oriented toward specific applications. However, the broadband Integrated Services Digital Network (B-ISDN) has been introduced as a high-capacity unified network supporting different services, which leads to economical resource utilization as well as unified network management operation and maintenance.

This chapter is organized as follow: Section 1.1 provides a comparison between different existing communication networks. Section 1.2 explains why Asynchronous Transfer Mode (ATM) networks have been introduced. The main features of ATM networks are listed in Section 1.3. Section 1.4 provides an overview of error recovery in communication networks. A survey of ATM switch architectures is provided in Section 1.5. Section 1.6 provides a comparison between different cell queue strategies. The main factors that affect the measuring of the ATM switch performance are listed in Section 1.7. Section 1.8 provides the thesis motivations and goals. Section 1.9 provides the organization of the thesis.

1.1 Networks Today and Tomorrow

Different types of communication networks are in use now. Each of them is optimized for the specific service it supports. Some of these networks are illustrated below:

1.1.1 Circuit switched Networks

Circuit switched networks are used for Constant Bit-Rate (CBR) services (e.g. voice). A circuit switched connection is set up by establishing a physical circuit connection through telecommunication networks. The physical circuit is set up when a call is initiated and the connection is terminated when the call is terminated. The circuit is dedicated to only one call for the duration of the connection, even if no data is being sent. Connection bandwidth can not be dynamically changed after the establishment of a physical connection. Classical switching techniques employed for this type of networks are Space Division switching (SDS), Time Division switching (TDS), and combinations of both techniques.

Circuit switching is inefficient for Variable Bit Rate (VBR) services, since the bandwidth requirements of the latter vary with time. Even in multirate circuit switched systems which allow the allocation of bandwidth in integer multiples of a basic rate, the choice of the basic rate is a difficult design decision. In order to accommodate the relatively low basic ISDN rate, many parallel low-rate channels must be established for high-rate services. This implies extra control overhead to synchronize all channels comprising a connection, and is a source of errors in practice.

1.1.2 Packet Switched Networks

Packet switched networks divide the available bandwidth between different supported traffic types and have a lot of similarities with TDS techniques. The main difference between packet switching and TDS is that packet switching allocates bandwidth dynamically while TDS allocates portions of the bandwidth permanently by assigning a time slot for every connection. The major characteristics of packet switching and TDS are presented on Table 1.1.

Characteristics	TDS	Packet Switching
Network delay	Low and fixed network transit delay	Variable network transit delay
Setup time	Needed to start a call	No setup time
Resources	Pre-allocation of resources is required and is dedicated for a user.	Resources is allocated on demand and can be shared among different users.
Address	Not used	Used
Application	Suitable for voice and video	Suitable for data traffic
Error handling	No error detection or correction mechanisms	Error detection and correction is done on link-by-link basis for most packet switched networks.
Complexity	Simple to design and operate	Relatively complex to design and operate

Table 1.1. TDS vs. packet switching.

VBR services are efficiently supported by packet switching networks. Data is sent in variable-length packets and each packet travels through the switching network independently. Sources transmit their packets when they are available. Each packet has a complete address to its destination so it can be routed inside the network. Different connections may share the same physical link. Since the packets have variable lengths, a complex buffer management scheme is required inside the network. This may lead to a large delay and delay jitter. In addition, a resequencing buffer may be required at the destination point to store out-of-order cells.

Clearly, circuit switching is easily managed and the delay inside the network is limited. However, to support VBR, a bandwidth based on the peak demand should be provided. On the other hand, efficient bandwidth utilization can be achieved by using a packet switching network. Unfortunately, the switching process is relatively slow and the delay inside the network is undetermined because each intermediate switch has to buffer variable size packets and check their destination addresses before sending them out and this makes packet switching unsuitable for CBR traffic.

Circuit and packet switching have been developed to serve completely different needs and each one of them performs well for the services it supports. However, it is uneconomical and inflexible to install and maintain separate networks for CBR and VBR traffic [1].

1.1.3 Fast Packet Switched Networks

Fast packet switching is a concept which can be divided into two different methods of data delivery: Frame relay and cell relay. The frame relay method is very close to packet switching and can be implemented using existing network hardware. Cell relay is a new technique for WAN which requires major changes in hardware.

Frame relay is a high-speed version of packet switching. In frame relay networks, data is forwarded in variable length frames (similar to the packets in packet switching) which are multiplexed to share the same physical transmission links. Each of the frames include addressing information that can be used to route the frame. Frame relay networks operate at higher frequencies than existing packet switching networks (such as X.25) and is well suited for high-speed data applications but not for delay-sensitive applications (e.g voice and video) because of the relatively long delay inside the network. The intelligence of packet switching (error checking, correction, recovery, retransmission, flow control) lies in network nodes while in frame relay it resides at the user or network access hardware.

Cell relay combines the benefits of TDS and packet switching. It operates on the packet switching principle of statistically interleaving packets on a link on an 'as required' basis, rather than on a permanently allocated slots basis. Using the bandwidth 'as required' achieves a high link utilization. The size of the packets that are multiplexed onto the link is fixed and the packets are called cells. Cells from different sources may share the same physical link and no time-slot allocation is required.

1.1.4 ATM Networks

ATM is a connection-oriented packet switching transfer mode based on statistical time-division multiplexing techniques. ATM is a compromise between circuit and packet switching. Routes are selected when the connection is being established. In addition, switch buffers and link time-slots may also be allocated. The physical links are dynamically time-shared between different connections. As a result of the deterministic nature of cell switching in ATM networks, it can be employed for different kinds of traffic with high effectiveness.

The ATM concept has also been appreciated by manufacturers of computer systems and LANs as a flexible switching and multiplexing technique for a wide range of applications. ATM networks provide flexibility in bandwidth allocation and carry diverse services ranging from narrowband to wideband [5]. However the challenge is to build high-capacity fast packet switches and an error recovery protocol that satisfies different traffic requirements.

1.2 Objectives of ATM

Advances in communications and computer technologies have produced a significant and rapid increase in the demand for new communications services. Telecommunication carriers provide a wide range of services to their customers, including transmission of voice, video and various kinds of data [2]. They also offer specialized services such as broadcasting and alarm monitoring.

As the result of the growth of digital technology in society, many services such as voice, data, image and video services have been migrating towards digital technologies for economic and quality reasons. The merging of high-speed packet data multiplexing and switching technologies, coupled with low-error fiber transmission, has offered the best performance in supporting different services simultaneously [3].

B-ISDN is a high-speed digital network architecture that supports many types of services. ATM has been adopted as the preferred transfer technique for B-ISDN [4]. ATM networks are expected to provide a large number of services for various applications. Predominant services will be the support of multimedia traffic (e.g. data, real-time audio and video). The traffic classes supported differ widely, not only in their individual bit-rates but also in their burstiness and Quality of Service (QoS) requirements.

Currently ATM networks are used for LAN interconnections and LAN emulations. These networks require bandwidths of a few hundred Mbit/s. In contrast, in the future it is expected that residential broadband applications will become universally available and this requires high-capacity networks. With the growth of traffic volume in ATM networks, the need for high-capacity ATM switches becomes a necessity. Future B-ISDN control offices will likely require a switching capacity of a terabit per second or more. In addition, a fast end-to-end error recovery protocol will become a must.

1.3 ATM Features

Briefly, ATM networks have the following properties:

- Small and fixed cell size

  An ATM cell consists of 53 bytes. The payload is 48-bytes long and it carries user data. The remaining 5 bytes are header control information, as shown in Figure 1.1 and Figure 1.2. The header carries control information about how a cell should be routed [8]. The Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) addresses are assigned at connection establishment and remain unchanged for the duration of the connection. The VPI and VCI values uniquely identify a connection on a given physical link. The hierarchical relationship between virtual channel, virtual path and physical links is illustrated in Figure 1.3. A physical link is multiplexed to provide a number of VPs. Each VP is a number of VCs.

The small size of the ATM cell simplifies switching node design. It also reduces cell delay and delay jitter, which is a major requirement, espically for delay-sensitive services such as voice and video.

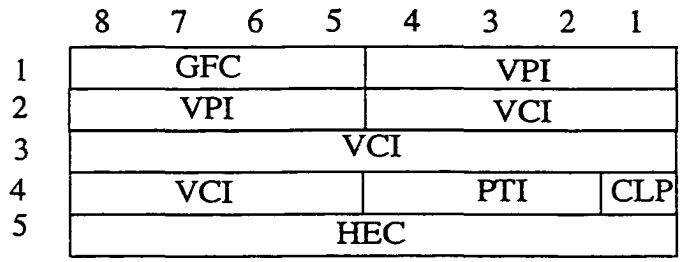

Figure 1.1 ATM header structure for User-Network Interface (UNI).

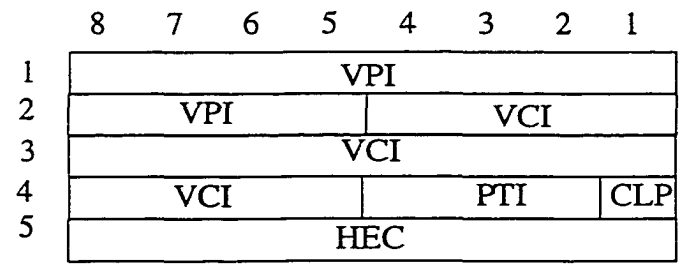

Figure 1.2 ATM header structure for Network-Network Interface (NNI).

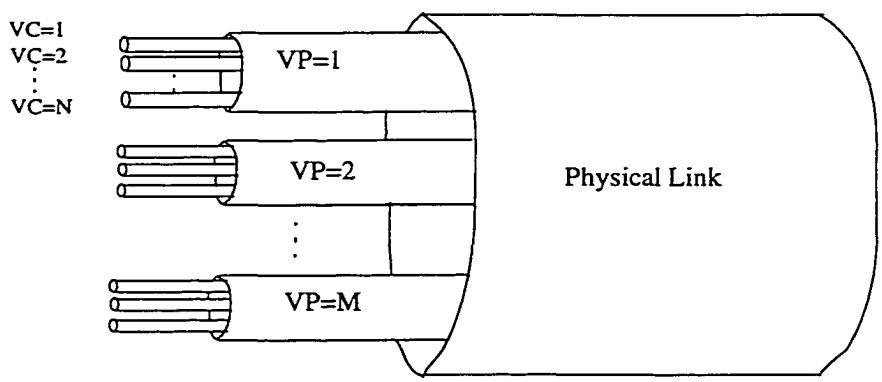

Figure 1.3 Relation between virtual channel, virtual path and physical path.

- No error protection or flow control on a link-by-link basis

  As the optical links in ATM networks have a very low bit-error rate, no error protection or flow control is implemented on a link-by-link basis. Transmission errors are recovered from by relying on end-to-end protocols [9].

- Different quality of service

  Quality of service (QoS) is defined as acceptable cell-loss rate, cell-delay and variance of delay for a supported service. The different services supported by ATM networks require different QoS. ATM networks support high-bandwidth services and services having bursty data-rate requirements. Bandwidth allocation depends on the burstiness of the applications supported. ATM networks must satisfy the QoS for all of the supported services.

- Connection-oriented operation

  During call setup phase, the network decides whether to accept or reject a call. This decision is based on the availability of sufficient resources to support a new connection. After a connection is established, some resources may be allocated at each switching node to maintain the QoS of this call. These resources are freed only when the call is terminated. Although ATM networks are connection-oriented, they also support connectionless services (AAL5 services) [9].

1.3.1 B-ISDN Layer Model

Like the Open System Interface (OSI) communication model, a hierarchical architectured model is used for ATM, to divide the functionality. Figure 1.4 includes the lower three layers of the ATM protocol model: The physical layer, ATM layer and ATM adaptation layer. The ATM protocol reference model uses separate planes for user, control and management functions, as described in International Telecommunications Union (ITU) recommendation I.321 and I.320. The user plane is responsible for transporting user data. The control plane establishes and terminates virtual connections. It also performs the critical functions of addressing and routing. The management plane arranges the co-ordination between user and control planes.

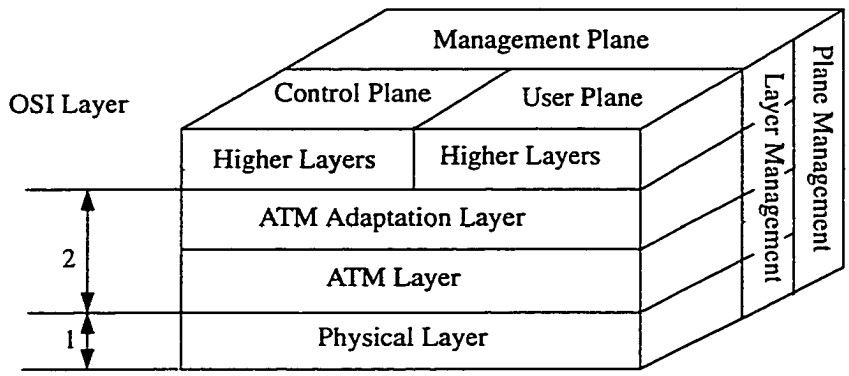

Figure 1.4 ATM protocol reference model.

For each plane, different layers are used, with independence between layers. The physical layer performs the same function as layer 1 in the OSI model. It performs at the bit level. The ATM layer performs as the lower part of layer two of the OSI model. The ATM adaptation layer performs the higher layer protocol of layer two of the OSI model.

Physical Layer

The physical layer comprises two sublayers: The Physical Medium Dependent (PMD) sublayer and the Transmission Convergence (TC) sublayer. The PDM sublayer provides the actual transmission of bits. The TC sublayer transforms the flow of cells into a steady flow of bits; it is needed to make the differences between physical layers standard.

ATM Layer

The ATM layer provides cell transfer capabilities. The characteristics of this layer are independent of the physical medium used. The ATM layer provides cell multiplexing, demultiplexing and routing functions. Furthermore, flow control and cell policing are supported by the ATM layer. No re-transmission of lost or corrupted cells is performed at ATM layer.

ATM Adaptation Layer (AAL)

ITU recommendation I.362 defines the basic principles and classification of AAL functions. There are 4 classes defined based on the timing and connection mode of supported services. The services could be constant bit-rate (CBR) or variable bit-rate (VBR) while the connection mode is connection-oriented or connectionless. The four classes are

Class A: CBR services with end-to-end timing in a connection-oriented mode.

Class B: VBR services with end-to-end timing in a connection-oriented mode.

Class C: VBR service with no end-to-end timing in a connection-oriented mode.

Class D: VBR services with no end-to-end timing in a connectionless mode.

Table 1.2 shows the attributes of the four service classes.

AAL1 through AAL4 are defined to implement the 4 classes. As AAL3 and AAL4 have many common features, they were combined into AAL3/4. As a result of the complexity found in AAL3/4, AAL5 was developed for computer applications. AAL5 has been adopted by the ATM forum, American National Standards Institute (ANSI), and ITU. It has become the predominant AAL in many data communication equipments. AAL5 supports connection-oriented and connectionless VBR services and it has been standardized for supporting Internet Protocol (IP) traffic, frame relay and signaling messages.

AAL5 allows ATM networks to carry commonly used protocols. Two methods are defined for carrying multi-protocol over ATM: Protocol encapsulation and VC multiplexing. Protocol encapsulation allows for multiple protocols to be multiplexed over a single ATM VC. Protocol encapsulation operates by prefixing the Protocol Data Unit (PDU) with IEEE 802.2 logical link control (LLC) which identifies the PDU type. VC multiplexing allows different protocols to be simultaneously carried over one ATM network. Each protocol is carried over a separate ATM VC.

Attribute	Class A	Class B	Class C	Class D
Timing relation between source and destination	Required	Required	Not required	Not required
Bit rate	Constant	Variable	Variable	Variable
Connection mode	Connection-Oriented	Connection-Oriented	Connection-Oriented	Connection-less
AAL(s)	AAL1	AAL2	AAL3/4 or AAL5	AAL3/4 or AAL5
Example(s)	DS1, nx64 kbps emulation	Packet video, audio	Frame Relay	IP

Table 1.2. ATM service classes.

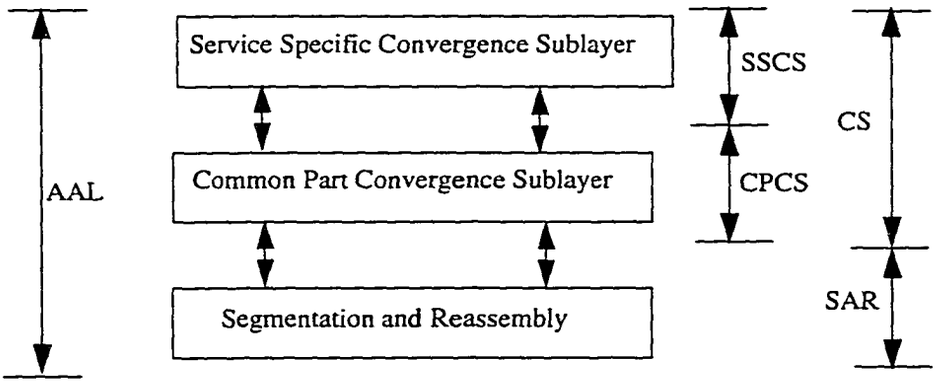

Figure 1.5 Generic AAL protocol sublayer model.

A generic AAL protocol sublayer model is shown in Figure 1.5. It comprises of the Convergence Sublayer (CS) and Segmentation and Reassembly (SAR) sublayers. The CS is divided into the Service Specific Convergence Sublayer (SSCS) and the Common Part Convergence Sublayer (CPCS). SSCS supports connection-oriented and connectionless modes. For AAL5, end-to-end error recovery protocol which is responsible for re-transmitting and re-ordering lost data is located in SSCS [29, 30].

1.4 Error Recovery in Communication Networks

Due to the inevitable presence of noise, communication systems must be capable of handling transmission errors. Traffic characteristics in high-speed networks can be divided into real-time data and non-real-time data. For real-time data where packet-delay is important, Forward Error Correction (FEC) is the appropriate solution to correct erroneous or lost packets because it corrects the errors without introducing retransmission delay. For non-real-time data, retransmission may be used to achieve reliable communication. Retransmission increases the load in the network and reduces the network throughput. Automatic Repeat Request (ARQ) is the best scheme for handling the retransmission of erroneous or lost packets [1].

The basic concept of ARQ is to detect the erroneous or lost packets at the receiver side then send a retransmission request to the sender [2]. There are three basic ARQ protocols: Stop-and-Wait (SW), Go-Back-N (GBN), and Selective Repeat Protocol (SRP). Many modifications on these basic techniques have been proposed to achieve better performance or to customize them for certain environments. SRP has recorded to have the best performance [86, 95]. This protocol is based on positive/negative acknowledgments (ACK/NACK) and time-out. If a packet is correctly received, an ACK is sent from the receiver to the sender. If the receiver discovers that a packet is lost, it sends NACK for this packet. If the sender does not receive an ACK or NACK after a predetermined interval, called the time-out, the packet will be considered lost and will be retransmitted.

The high throughput of SRP is achieved at the cost of out-of-order arrival of packets at the receiver and by using a large number of control packets. To deliver packets in order at the receiver, a resequence buffer is used. This resquence buffer stores packets until the missing packets have been received. The resequencing process introduces a delivery delay which should be kept as small as possible.

In ATM networks, loss-sensitive applications need a fast end-to-end error recovery protocol. The error recovery protocol should deliver lost cells with a minimum number

of control cells. In addition, error recovery delay should be minimal and the protocol should be robust enough to work efficiently in either congested or uncongested network states.

The SSCS Protocol Data Unit (SSCS-PDU) frame in AAL5 contains 24 bits used as a frame sequence number. This 24-bit sequence number allows very high sustained rates to be achieved in a window flow controlled protocol.

1.5 ATM Switch Structures

ATM is a high-speed connection-oriented packet switching technique with minimal functionality in the network, where traffic is segmented into cells for transmission across the network. The small size of ATM cells simplifies switch architecture and reduces delay inside the network. However, the sequence integrity of all cells on a virtual connection must be preserved across the ATM switches. The small, fixed cell size and the fixed address location of VPI and VCI have an important influence on the optimal ATM switching architecture.

The main function of an ATM switch is to physically route a cell from input port I_i to output port O_j, based on its header value. ATM switches change the VPI and VCI values while switching a cell and routing it to the next switch. A conflict may occur when cells from different input ports are destined to the same output port. Therefore, there should be a buffer to store cells which cannot be immediately served.

The rapid pace of VLSI has brought new switching system concepts for meeting these high-performance requirements. Different levels of parallelism and distributed control can be used to achieve the high-speed requirements. In the following sections, we will introduce some of the existing ATM switch architectures.

ATM switches can be classified into blocking and nonblocking switches [43]. For a nonblocking switch, N cells from N different input ports can be simultaneously directed to N different output ports. For a blocking switch, the simultaneous switching for N cells from

N different input ports to N different output ports can not be achieved. ATM switching architectures can also be classified into time-division architectures and space-division architectures.

1.5.1 Time-Division Switches

For pure time-division switches, incoming cells flow across a single communication medium. This communication medium is a shared bus or a shared memory. Since every cell flows through a single shared communication medium, this class of switches easily supports multicasting.

Shared-Medium Architecture

In a shared-medium architecture, shown in Figure 1.6, incoming cells are multiplexed onto a common medium, typically a bus or a ring. A FIFO is required to store incoming cells until they can access the medium. Output contention can not occur in this architecture as two or more cells can not arrive at an output port simultaneously. However, output buffers are required if the arrival rate of cells at a particular output port exceeds the outgoing link bandwidth. A central controller regulates access to the shared medium. The maximum capacity of the shared medium puts an upper limit on the capacity of the switch. Shared-bus switches with capacities up to 10 Gb/s have been designed [10, 11].

The scalability of time-division switches is limited by the bandwidth of the shared medium and the centralized controller requirements. Shared-medium architectures do not scale well and can only support a relatively small number of ports. Alternatively, they can be switching elements for large switches, in which each unit is connected to others according to some particular topology.

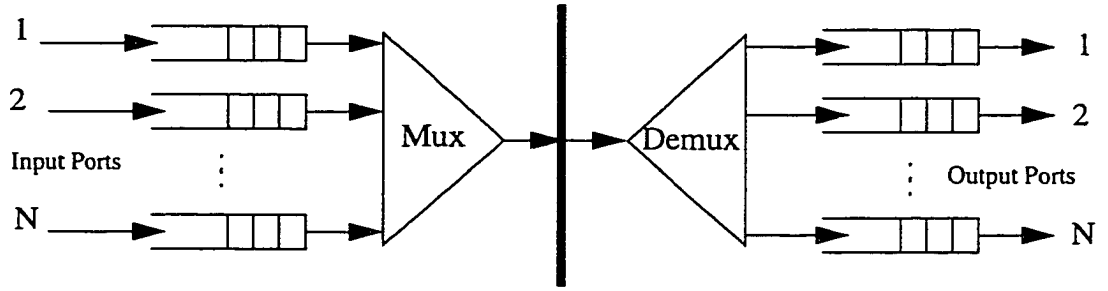

Figure 1.6 Shared-medium ATM switch.

Shared-Memory Architecture

A shared-memory switch consists of a single memory module shared by all input and output ports, as shown in Figure 1.7. Incoming cells are multiplexed into a single stream and written to the shared memory. Cells are read out from the memory, demultiplexed and transmitted to outgoing links.

Shared-memory access time places an upper limit upon the switch capacity. Shared-memory switches have been reported for a capacity up to 10 Gb/s [2].

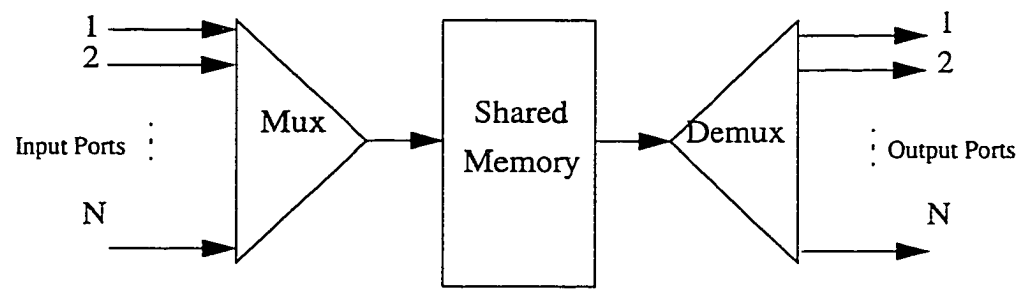

Figure 1.7 Shared-memory ATM switch.

1.5.2 Space-Division Switches

Space-division switch fabrics are divided into two classes: single-path or multi-path. In a single-path switch, only one path exists for any input output pair, while in a multi-path switch, there is more than one path for each input-output pair.

Single-path networks

Single-path interconnection networks have only one path between any input-output pair. Single-path networks always have a limited capacity. In this section, some of these networks are briefly discussed.

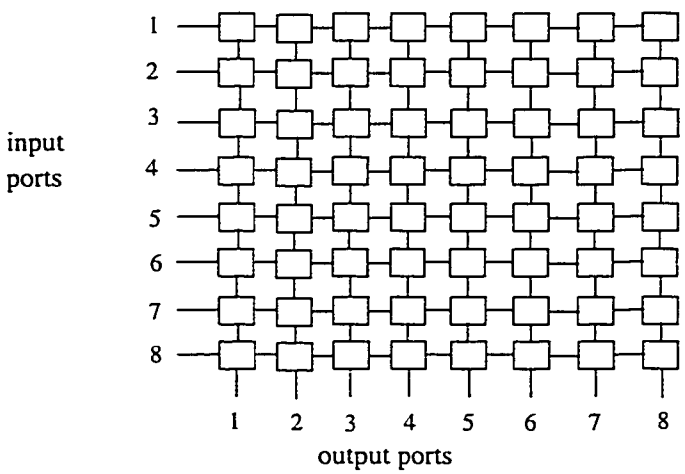

Figure 1.8 An 8x8 crossbar switch.

Crossbar: The crossbar topology is shown in Figure 1.8. The complexity of the crossbar grows as a function of N^2 (N is the number of input/output ports). The crossbar architecture is suitable for nonblocking, self-routing switches. As long as cells at each input port are destined for different output ports, the crossbar switch allows N connections to be simultaneously established, thereby achieving simultaneous delivery of N cells. Knockout switch [67, 68] is an example of a crossbar switch architecture.

Banyan: Banyan switch architecture is a family of 2x2 switching elements with a single path between any input-output pair. An example of the switching process is shown in Figure 1.9. The complexity of Banyan switch architecture is of order $N\log N$. This architecture is blocking and its performance degrades rapidly as its size increases. Switches that are based on this architecture were reported in references [12, 13, 14].

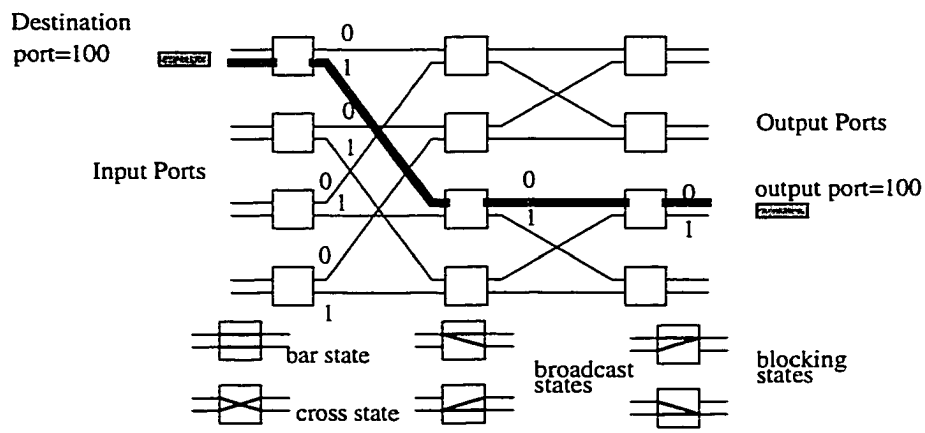

Figure 1.9 Basic structure for Banyan switch architecture.

Batcher-Banyan: The blocking probability in Banyan switches can be reduced by sorting the incoming cells based on their desired output ports. An example of how the Batcher sorter is working is shown in Figure 1.10. An (N*N) Batcher switch can be built using $\log_2 N(\log_2 N+1)/2$ stages each with $N/2$ binary comparators or sorting elements [19]. The Batcher and Banyan switch is non-blocking if no more than one cell is simultaneously destined for the same output port. The complexity of this switching type is given by $N*\log N^2$. Some Batcher-Banyan based switch architectures are found in references [8, 20, 21, 22, 23].

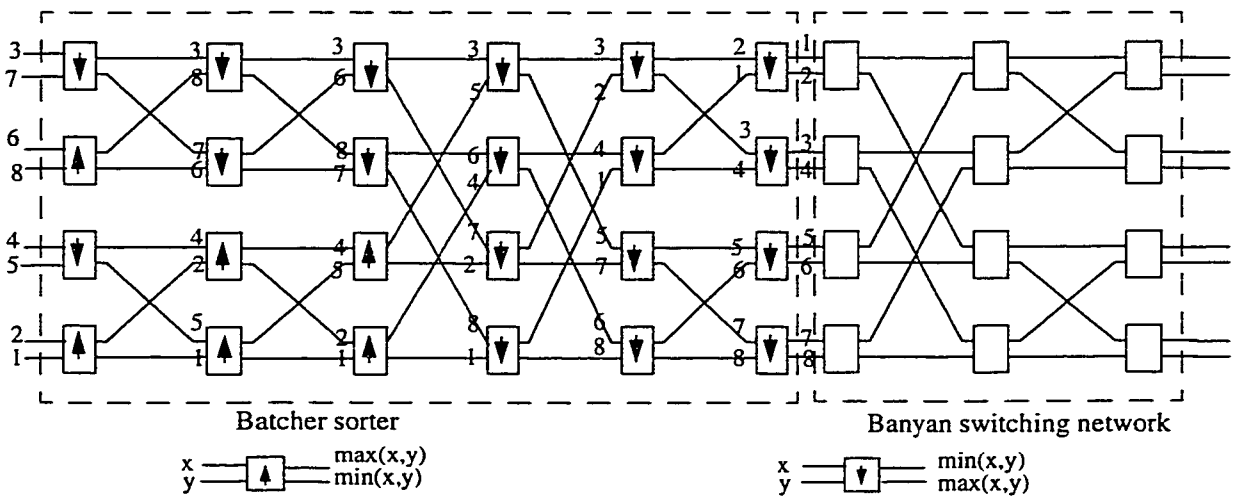

Figure 1.10 Batcher-Banyan switching network.

Multiple-path networks

For space division switches, multiple simultaneous paths can be established between input and output ports. These paths operate concurrently so that many cells can be transmitted across the switching fabric at the same time. The total capacity of the switch is calculated by multiplying the line speed bandwidth by the number of paths that can be simultaneously established. Routing can be centralized or distributed. Centralized routing examines the destination address of incoming cells and sets up required paths accordingly. In this case, the growability of the switch is limited by the capacity of the central controller. In *distributed routing*, also known as *self routing*, each input controller of the switch performs routing for cells coming into a port. As there is more than one path between an input and an output port, the out-of-order problem should be carefully considered when designing multiple-path switches. Multiple-path switches are used for improved performance over that of a single-path switch or to build a large switch from small switching modules.

Augmented Banyan/Benes: Introducing intermediate stages or switching elements to a Banyan switch creates multiple paths between input and output pairs. The Benes

switch is achieved by adding $\log_2 N$-1 stages to a Banyan switch . Switch architectures that use this approach can be found in [24, 25, 26].

Parallel Switch Planes: Multiple planes, shown in Figure 1.11, are connected in parallel to improve switching performance and achieve a higher degree of reliability. Some examples of architecture using this approach are found in [24, 27, 28, 29].

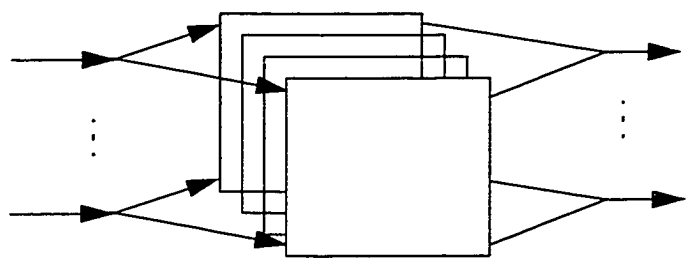

Figure 1.11 Multi-switching planes.

1.6 Information Rate and Queuing Disciplines

As a result of the probabilistic nature of the cell arrivals to an ATM switch, queuing is necessary to resolve contention for output ports. When two or more cells simultaneously arrive on different input ports destined to the same output port, one of them is passed to the output port but the other must be queued.

The most important factors in choosing a queuing strategy are:

- Queue size: The size of the queue depends on the performance requirements, such as cell loss and allowable delay. The main objective is to incur a cell-loss rate no more than the cell-loss rate of optical fiber transmission lines (i.e. 10^{-6}).
- Memory speed: The access time of the queuing memory depends on the memory organization, word length, and queue size.

- Queue arbitration: An arbitration scheme is required to control cell access. Arbitration scheme complexity depends mainly on queue architecture and performance requirements. For instance, the Head-of-Line (HOL) protocol may be used as a simple control logic but for higher performance requirements, a more complicated scheme may be required.

There is a large body of research concerned with memory organization in order to reduce the queuing time and prevent cell loss [80]. Typically, the memory is organized as an input queue, output queue, shared queue or as a combination of these.

1.6.1 Input Queue

For the input queue architecture, illustrated in Figure 1.12, a separate buffer is assigned to each input port. Each input queue buffers cells at the arrival rate of a single port. The internal speed of the input queue buffer is lower than other buffering approaches. Furthermore, the complexity of the buffer controller is less when compared to other approaches that are discussed next [42]. An input queue should perform one writing and one reading operations per time slot, where time slot (T_s) is the time required to receive a complete ATM cell from the physical interface and is defined as $T_s = \frac{53 \times 8}{L}$ *sec*, where L is the I/O link speed.

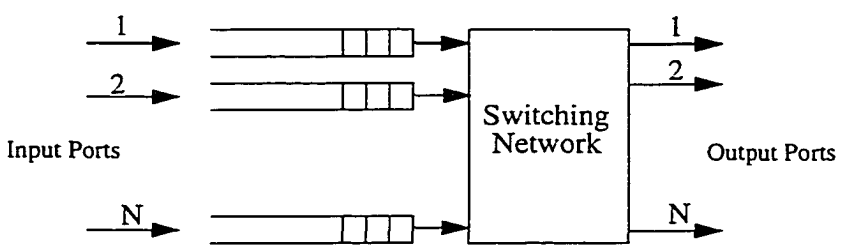

Figure 1.12 Input queue architecture.

The input queue scheme is affected by congestion at output ports. Since any particular input port can receive cells destined for different output ports, one congested output port can result in delaying cells destined for other output ports, because when a cell at

the HOL waits to be sent to a congested output port, successive cells in the queue will also wait. From the performance standpoint, a pure input queue switch which has infinitely large input buffers and infinitely large number of input and output ports and implementing First-In-First-Out (FIFO) scheduling protocol has a maximum throughput value (ζ_m) = 0.586 [43, 44]. The input queue strategy requires the least memory bandwidth because an input queue should be able to perform just one write and one read operation per time slot.

1.6.2 Output Queue

The output queue architecture is shown in Figure 1.13. Each output port has a dedicated queue. An output queue needs a high memory bandwidth [41, 69]. This is because, for an *N*N* switch, the output queue should be able to perform *N* writing operations and one reading operation per time slot. To reduce the memory bandwidth requirements, a limited number of cells may be written per time slot. In [40], it has been shown that using a concentrator relaxes the memory bandwidth requirements but increases the cell-loss probability.

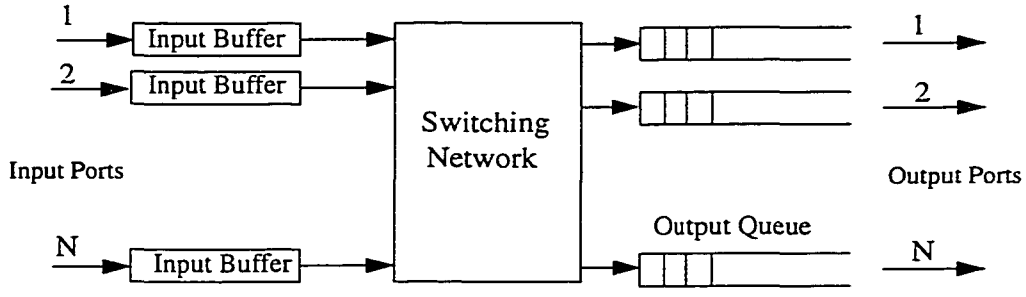

Figure 1.13 Output queue architecture.

1.6.3 Shared Queue

A shared queue architecture, shown in Figure 1.14, uses a shared memory that can be accessed by all input and output ports. All the input and output ports can access the shared buffer simultaneously. Incoming cells are stored in the shared buffer until the switching

network schedule them to destined output ports. A shared buffer architecture achieves better delay-throughput performance than an output queue architecture [62]. Advantages of a shared buffer memory include better buffer space utilization and less memory controller complexity, owing to having less memory to manage [45]. The main disadvantage of shared buffer memory switches is the high memory accessing speed requirement [43]. Another disadvantage is the unfair allocation of memory between different ports [70].

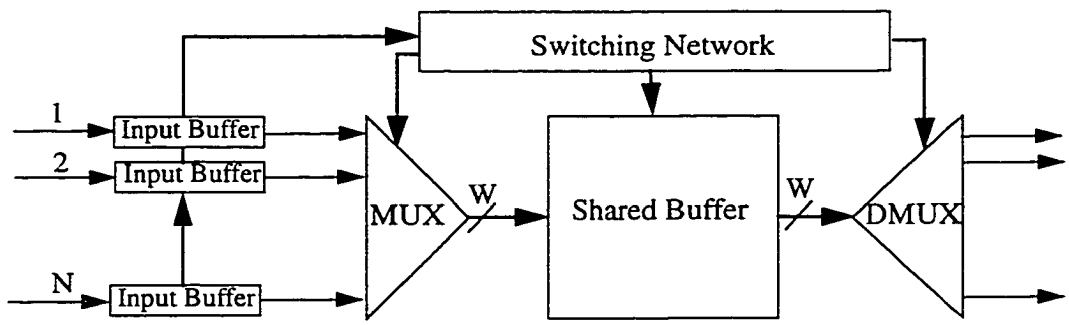

Figure 1.14 Shared buffering architecture.

For an *N*N* switch, the shared buffer has to perform *N* writing and *N* reading operations per time slot. Hence increasing the number of I/O links sharing a buffer necessitates reducing the required memory access time and places increased demand on shared memory speed, pin count and power dissipation.

1.7 Switch Performance Requirements

ATM switches should be able to provide broadcast and multicast functionalities, which are typically required for video conferencing and TV distribution. Switch performance is measured by throughput, connection blocking probability, switching delay, and cell-loss probability [45]. Typical values for cell-loss probability in ATM switches range between 10^{-8} and 10^{-11}. Typical values mentioned for delay in ATM switches range between 10 and 1000 μsec [31].

1.8 Research Motivations and Goals

The main research goals of this thesis are to address two issues that are crucial to establishing a reliable high-speed high-capacity ATM network. Achieving this type of network requires a fast high-capacity ATM switch that supports different QoS and a reliable end-to-end error recovery protocol.

A fast, high-capacity ATM switch is required because a great traffic volume will be carried by ATM networks. Low cell-loss, low cell-delay and high reliability should be satisfied by the switch architecture. Currently existing ATM switches support a limited traffic volume and none of them can support traffic in the range of terrabits per second. In addition, none of the existing switches uses a fair algorithm for priority control, or they use a slow software implementation of a fair priority control scheme, which is difficult to implement in hardware. To satisfy the high capacity requirement, ATM switches should have the priority control scheme implemented in hardware.

Current error recovery protocols in ATM networks do not efficiently use the bandwidth. Many control cells are used to achieve reliable communication. These control cells waste potentially profit-generating bandwidth.

In this thesis, we introduce a terabit ATM switch architecture which is built from small switching modules, each of which can be used alone as an ATM switch. The architecture of the switching module is introduced in Chapters 2 and 3. The architecture of the terabit switch is provided in Chapter 4.

Cell loss inside a network can approach an arbitrarily small value, but it can not be zero, as there may be cell loss as a result of congestion or component failures. For loss-sensitive ATM services, cell loss is not acceptable. An end-to-end error recovery protocol is a must to achieve reliable communication. This thesis introduces a novel end-to-end error recovery protocol; a limited number of control cells and limited cell-delay are the main features of the proposed protocol. The description and the analysis of the proposed protocol are provided in Chapter 5.

1.9 Thesis Overview

This thesis is organized as follow: Chapter 2 introduces a shift-register based ATM switch architecture which is used in chapter 4 to build a terabit switch. Chapter 3 provides a new cell scheduling algorithm that is used in the proposed shift-register based ATM switch to provide efficient support for different priorities of services. Chapter 3 also provides an analysis for the behavior of the proposed switch architecture. Chapter 4 introduces a high-capacity ATM switch architecture. This switch is built using small-capacity switching modules which have been introduced in Chapter 2 and Chapter 3. Chapter 5 provides a new end-to-end error recovery protocol. The thesis conclusions and future work suggestions are provided in Chapter 6.

Chapter 2 introduces an input-buffer based ATM switch architecture. It takes advantage of the low-speed requirement of the input buffer but without the Head-of-the-Line (HOL) blocking. The input buffers are implemented as groups of shift registers. The use of shift registers eliminates the need for serial-to-parallel and parallel-to-serial format conversions which improves the switching speed. The proposed architecture will be used as a building block in the terabit switch architecture proposed in Chapter 4.

Chapter 3 is concerned with cell scheduling for the shift-register based switch architecture. Provided that a switch is able to internally switch an incoming cell from an input port to an output port, cell delay inside a switch is introduced by the cell scheduler. The cell scheduler is responsible for satisfying different QoS for supported services. We introduce a new cell scheduling algorithm which satisfies QoS for delay sensitive, delay-insensitive, loss-sensitive and loss-insensitive services. The behavior of the proposed scheme under different traffic characteristics is analyzed.

High-capacity ATM switch architecture is the subject of Chapter 4. Three high-capacity switch architectures are proposed. Although the first and the second architectures can support a great traffic volume, their scalability are limited and their capacity can not be scaled up to support traffic volume in the range of terabits per second. The third

architecture, which is a space-switch based architecture, can be scaled up to support traffic volume in a range of more than a terabit per second. The shift-register based switch architecture which has been proposed in chapter 2 and chapter 3 is used as the main building block in the new terabit switch architecture. The modularity and regularity of the proposed architecture facilitates its scalability to different sizes. Different levels of redundancy are introduced to achieve a reliable switch.

Chapter 5 introduces an efficient end-to-end error recovery protocol which minimizes the number of control frames required to achieve reliable communication. The proposed protocol adapts itself to work efficiently during congestion. A complete analysis and a simulation that presents the efficiency of the proposed protocol are introduced.

The conclusions of the thesis and suggestions for future work are presented in Chapter 6.

Chapter 2

Shift-Register Based ATM Switch Architecture

In this chapter, we introduce a shift-register based ATM switch architecture. The proposed architecture will be used, in Chapter 4, as the main building block for a high-capacity switch architecture that can support traffic volume in the range of a terabit per second. The shift-register based switch takes the best features of the input buffer scheme. Furthermore, buffer access speeds match port speeds and the buffer acts in effect, as a multiport memory. The input buffers are implemented as groups of parallel shift registers. The parallel shift registers overcome the HOL blocking and low throughput problems of classical input buffers. The use of shift register buffers allows operating speeds much higher than what is possible using RAM buffers. The parallel nature of the input queues simplifies the supporting of multicast functions. In addition, the modularity of the proposed architecture facilitates its scalability.

This chapter is organized as follows: Section 2.1 provides a review of contention resolution schemes for input buffer switching. Section 2.2 introduces a novel shift-register based ATM switch architecture which implements input buffers with high throughput. The operation of the switch is explained in Section 2.3. Section 2.4 discusses the performance evaluation of the proposed switch architecture. The conclusions for this chapter are provided in Section 2.5.

2.1 Input Buffer Switches

Space switches with input queuing represent the simplest ATM switch architecture with minimum cost. Various implementations of the space switch have been investigated. Some of these implementations are crossbars, Batcher-Banyan and variations of these types. Regardless of the implementation, throughput degradation caused by contention for finite bandwidth within the switch has been a major problem. If the head-of-the-line cell from different input queues are destined to the same destination, one of them will be forwarded to that destination and the others will be blocked. The throughput of input-queuing switches is limited due to *external blocking*: the presence of cells in multiple input queues that are simultaneously destined for the same output port.

2.1.1 Contention Resolution

Contention resolution schemes can be centralized or distributed. Many contention resolution schemes have been proposed for input buffer ATM switches. Each one of them has its pros and cons, and here we review their advantages and disadvantages.

- Ring reservation [62]: Input ports are interconnected via a ring which is used to request access to the output ports. No centralized arbitration processor is required. No extra switch traffic is generated. A string of one-bit tokens is passed around a ring through all input buffers in pipelined fashion. Each bit represents the status of an output port. If the head-of-line cell is destined for a port corresponding to a free token, the cell is granted admission. If the token was previously taken, the cell is blocked. Note that the ring speed should be N times the speed of the external links, to be able to poll all the N ports. It is hard to include traffic having different priorities with this scheme.
- Sort and arbitrate [65]: All cells are sorted based on their destined output ports. After sorting, cells requesting the same output port appear adjacent to each other. An arbitration mechanism is implemented to select between the sorted cells. The use of sorting network and an arbiter introduces delay, power and area penalties.

- 3-Phase Algorithm [63]: In phase 1, all input ports issue requests for required output ports. In phase 2, requests are arbitrated by the switch fabric hardware and acknowledgments are sent to winners. During phase 3, acknowledged ports forward head-of-line cells. This algorithm requires a significantly large processing overhead and significant traffic which may cause an unacceptable delay. In addition, it suffers from fairness problem since either lower or higher address input ports are given priority in the allocation of the switch bandwidth.
- Scheduling Content Addressable Memory (SCAM) [66]: With the arrival of a cell, a request is sent to an arbitration processor which employs SCAM. The SCAM determines the earliest time at which the requested output port is free. The input port is instructed to send a cell at that time. Central communication is a disadvantage for this algorithm. In addition, priority handling is impractical.

2.1.2 Head-of-Line Blocking Alleviation

Head-of-line (HOL) limits the input buffer switch throughput to 0.586 [43, 44]. Many schemes have been developed to alleviate HOL blocking.

- Output buffering: Creates more than one path between each input-output port pair. More than one cell may be sent to an output port at the same time. This method requires additional output buffers.
- Grouping: Combines the switch output ports into groups. A cell is sent to a group instead of sending it to a specific output port.
- Windowing: Allows scheduling for cells behind the HOL cell. It requires extra contention cycles to schedule different cells.
- 2-Dimension Round Robin: Instead of a single queue per input port, a shared buffer is employed. Separate logical queues per destination are implemented at each input port. Using a 2-dimensional round robin arbiter, N^2 requests from N input ports are issued. A complicated central arbiter and a complicated buffering hardware unit are required.

2.2 Proposed Shift-Register Based Switch

We introduce a new nonblocking ATM switch architecture. The main features of the proposed switch are:

- Shift registers are used as input buffers.
- Each input port is assigned a group of shift registers.
- All shift registers are connected to the switching module via a concentrator. This overcomes the HOL priority and low throughput problems.
- No serial-to-parallel format conversion is required as data will be accepted, stored and dispatched in bit-serial fashion. This reduces communication area and power requirements.
- Memory access time is independent of switch or memory size.
- Multicasting functions are easily supported.
- The modular design of the proposed architecture permits scaling the switch size.
- Virtual output queues have been implemented to prevent HOL blocking and facilitates the support of different QoS.

2.2.1 Switch Architecture

The switch architecture, shown in Figure 2.1, is divided into five modules:

1. Input module

   Each input module (Figure 2.2) has a dedicated input buffer, an input scheduler, an input controller, and a concentrator. The input buffer is composed of shift registers. The input scheduler determines the next available shift register to receive an incoming cell and it routes the incoming cell, which is assumed to be received in a bit-serial fashion, to a storing register. The input controller updates the header of the received cell. The shift-registers are connected to a concentrator which is controlled by the output modules, a feature that eliminates resource contention. In a traditional switch design [80], the concentrators

are controlled by the input modules. This naturally gives rise to contention and arbitration problems.

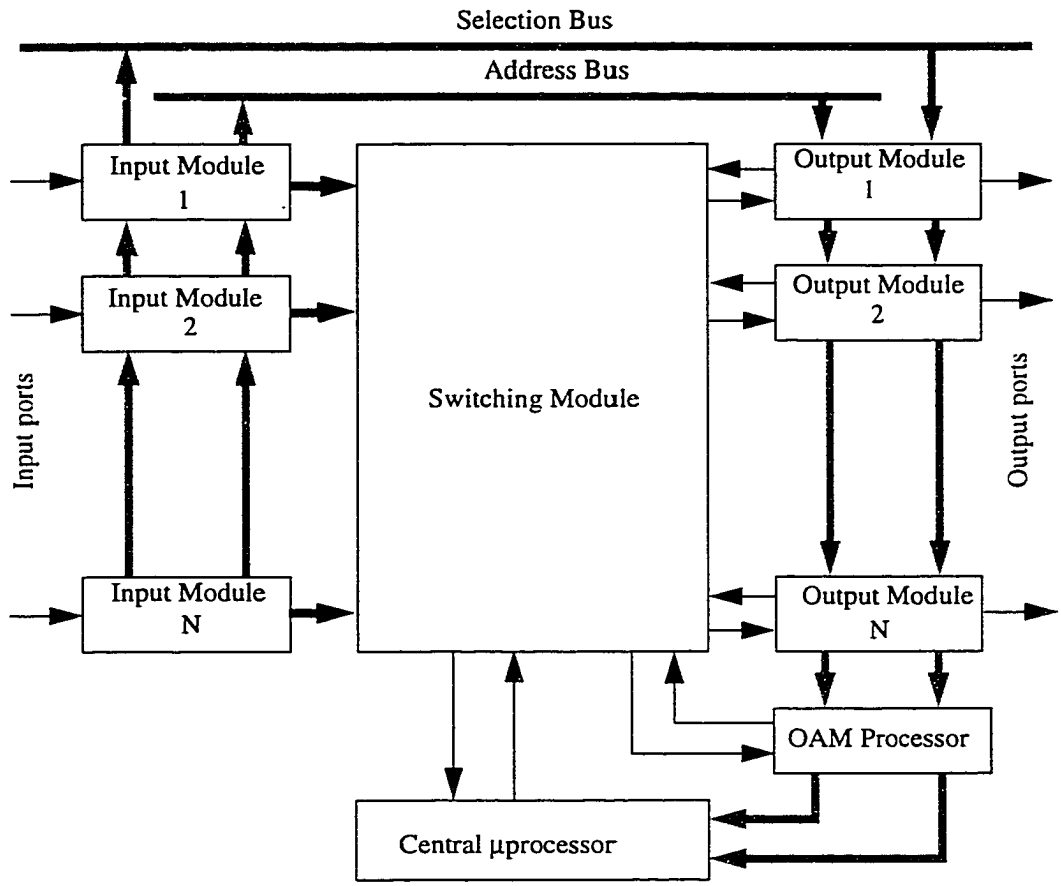

Figure 2.1 Block diagram of the shift-register based switch architecture. Gray lines represent control signal paths.

2. Output module

Each output port (Figure 2.3) has a virtual queue and an output scheduler. The virtual queue contains the addresses of the cells destined to that port. The output scheduler updates the virtual queue every time a new cell is received. It also controls the switching module to transmit a cell from an input buffer to an output port.

3. Switching module

   The switching module is basically a shared bus of N lines equals to the number of output ports. The first line of the bus is connected to the first output module and the second line is connected to the second output module and so on. The first output lines of all the concentrators in the input modules are connected to the first line of the bus through a tri-state buffer. The second output lines of all the concentrators are connected to the second line of the bus through a tri-state buffer, and so on.

4. Operation, Administration, and Maintenance processor

   Operation, Administration, and Maintenance (OA&M) functionality is supported by a central OAM processor. OAM cells are sent to the OAM processor which reacts based on the information carried in these cells.

5. Central micro processor

   A central micro processor is the master arbiter for accepting or rejecting a connection request. It maintains updated information about switch resource allocation. A new connection request will be accepted only if there are enough resources to support it.

There are two buses connecting the input and output modules: an address bus and a selection bus. The address bus carries the address of a particular cell in the input buffer to the virtual queue of the desired output port. The selection bus enables one or more output virtual queues to get address information carried by the address bus.

2.2.1.1 Input Module

Each input port has an input scheduler, a shift register bank, a concentrator, an Usage parameter control (UPC) processor and an input controller, as shown in Figure 2.2. The input scheduler keeps updated information about the occupancy of the input buffers associated with the input port. The input controller determines the desired output port based on

the header value and accordingly updates the virtual queue of the desired output port. In essence, cell routing decisions are performed locally for each input port. To avoid access conflict, there is a separate controller for each input port, rather than a single shared controller. An incoming cell is stored in two separate shift registers, the header buffer and the payload buffer. Reading the header starts as soon as the first five bytes of the cell arrive. Thus cells in all input ports can be processed simultaneously as soon as the header is received, even before the cell payload arrives. This gives the input controllers enough time to perform more complex tasks.

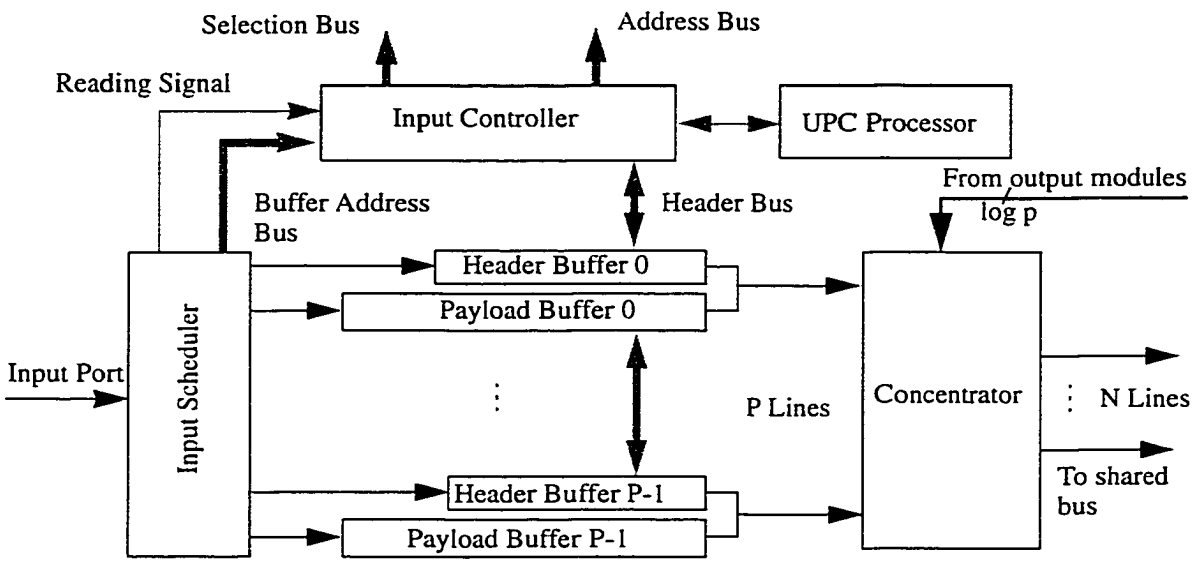

Figure 2.2 Input module block diagram. Gray lines represent control signal paths.

The buffers are connected to a concentrator which is controlled by output modules. The concentrators are connected to output modules using *log P* lines which are shared between all output modules. If there is a cell destined to the first output port, it will be delivered to the first output line of the concentrator which is connected to the first line of the shared bus. The maximum number of cells that can be sent from an input module during a switching cycle is the number of output ports (N). UPC is located in the input module. It ensures

that user traffic complies with the traffic contract. Remedial actions are taken for violating cells.

There are two buses inside the input module: the buffer address bus and the header bus. The buffer address bus carries the address of the input buffer that received the incoming cell. The bi-directional header bus routes the header data from the header buffers to the input controller and carries the updated values of VPI and VCI for unicast traffic or a multicast identifier back to the header buffer. Also, there is a reading signal which is used to inform the input controller that there is a cell that needs to be processed.

2.2.1.2 Output Module

The output module is shown in Figure 2.3. Each output port has an output scheduler, a virtual queue, and a flow controller. The output scheduler controls the concentrators in the input modules to get a stored cell from an input shift-register. Furthermore, it keeps updated information about the occupancy of the virtual queue.

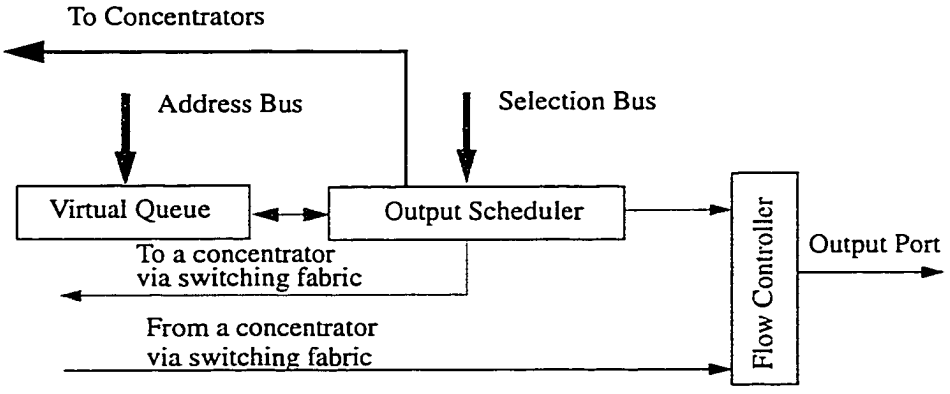

Figure 2.3 Output module block diagram. Gray lines represent control signal paths.

The virtual queue keeps a list of the cell addresses destined for the output ports, effectively turning the input shift registers into output queues. Cells are scheduled out from virtual output queues and this eliminates any possibility of HOL blocking. With virtual queues, the switch acts exactly as if it is an output buffering switch.

The priority scheduler for the output queuing strategy is simpler and more efficient than for input queuing. This is because, input queuing requires priority scheduler at each input port to prioritize cells coming from that input port and heading to the same output port and another priority scheduler at each output port to prioritize cells going out from that port. Using virtual output queues makes it easy to support any priority scheduler designed for output buffer queuing. A new scheme for a priority scheduler that can be implemented as a part of the output scheduler is introduced in Chapter 3.

The flow controller regulates the sending of cells out of an output port. Any VC is unable to send a cell unless it has credits (for credit-based flow control) or has not exceeded the maximum transmission rate (for rate-based flow control). The flow controller supports credit-based and rate-based flow control. Rate-based flow control is supported in a credit format. The flow controller converts the supported rate to a number of credits per second. Only one engine is required to support rate-based and credit-based schemes. For more information about credit-based and rate-based flow control refer to [100, 101, 102].

2.2.1.3 Switching Module

The switching module is simply a shared bus. The concentrator outputs in all the input modules are connected to the shared bus via tri-state buffers. The first output lines of all the concentrators are connected to the first line of the shared bus and the second output lines of all the concentrators are connected to the second line of the shared bus and so on, as seen in Figure 2.4. Each output port is also directly connected to a line on the shared bus. A tri-state buffer is enabled when the concentrator output connected to it is active. Output modules share the using of $log\ N$ lines to control the 3-state buffers.

The ith output module controls an input concentrator to connect the shift-register, which has a cell destined to the ith output port, to the ith line of the shared bus and enables the tri-state buffer connected to that line.

The ith output module controls all tri-state buffers connected to the ith line of the shared bus. To get a cell destined for this output port, the appropriate tri-state buffer is enabled.

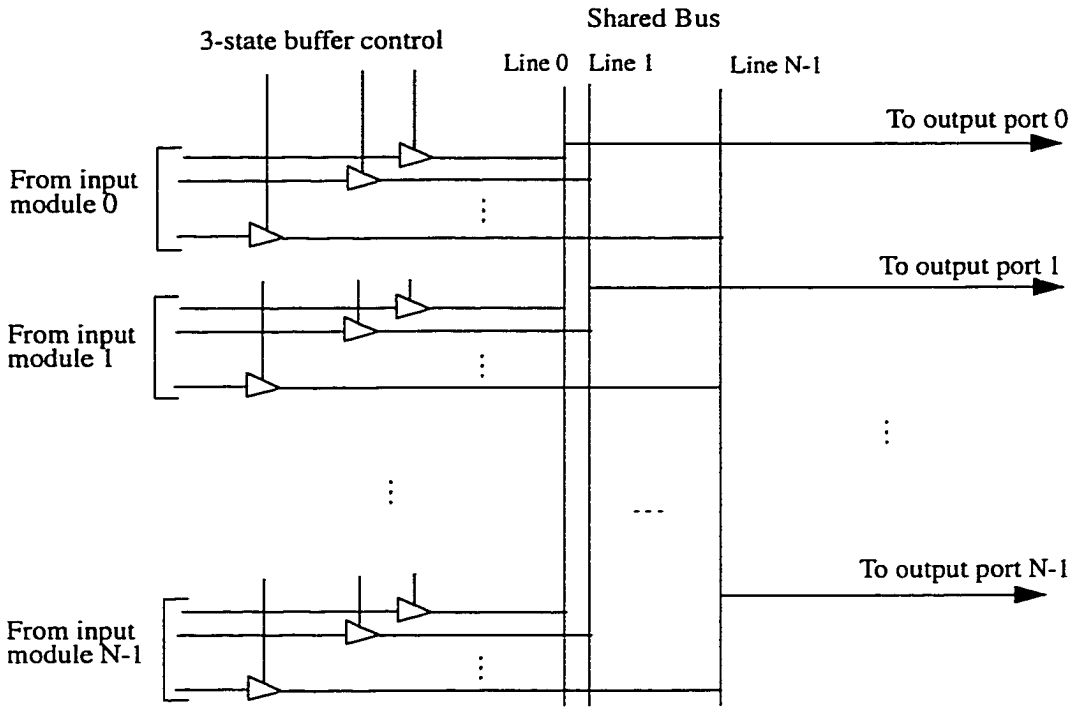

Figure 2.4 Switching module block diagram.

2.2.1.4 OAM processor

The OAM processor handles incoming OAM cells. OAM functions can be divided into 3 groups [96, 97]: Fault management, performance management and activation/deactivation. Fault management discovers or declares a fault in the network. Performance management measures the performance of the network by counting the number of misinserted and lost cells. Activation/deactivation enables or disables specified OAM functionality.

2.2.1.5 Central μprocessor

The connection Admission Control (CAC) process is provided by the central μprocessor. Signaling messages are forwarded to the central μprocessor. The central μprocessor is able to insert a signaling message whenever it is needed.

2.3 Switch Operation

The switch operation can be described by explaining the data and control flows inside the switch for receiving and sending cells.

2.3.1 Control and Data Flow For Receiving a Cell

Figure 2.5 shows the control flow between an input module and an output module. The control flow proceeds in the following fashion.

1. When a cell arrives at an input port, the input scheduler directs it, as a bit-serial stream, to an available shift register.
2. After storing the header, the input scheduler asserts the reading signal, thereby informing the input controller that there is a cell to be processed. It sends the stored cell address to the input controller using the local buffer address bus.
3. The input controller reads, in parallel, the cell header.
4. The input controller sends VPI and VCI values to the UPC which validates the incoming cell. If the cell is violating the traffic contract, a UPC action is taken.

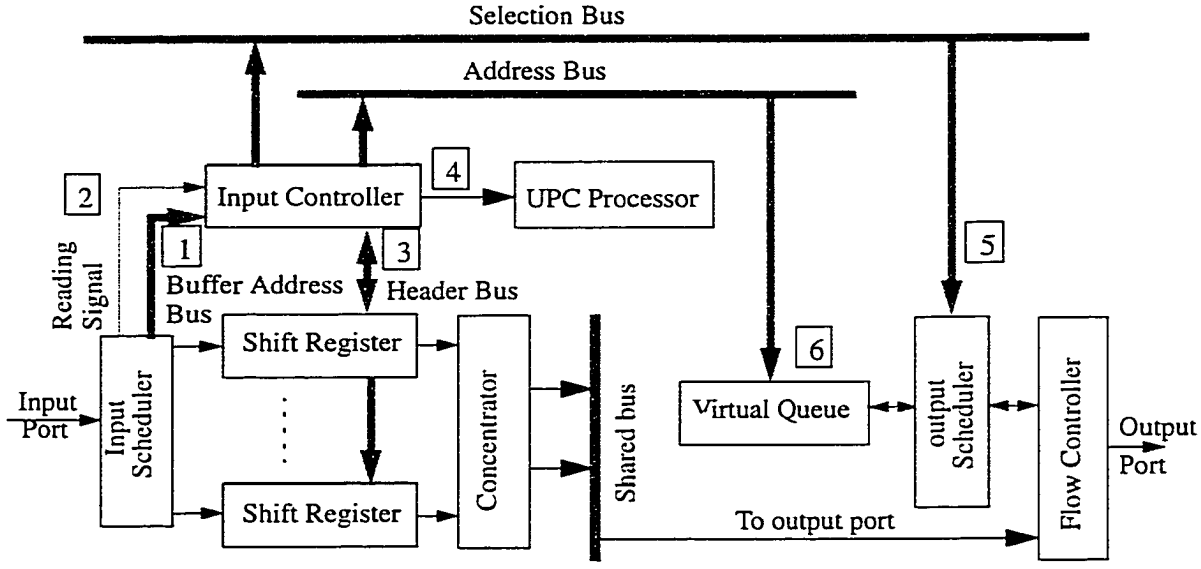

Figure 2.5 Control and data flow for receiving a cell.

5. If the cell is not violating the contract, the input controller determines the destination port and sends the local buffer address of the stored cell with a flag indicating if the cell is a unicast or multicast cell to the output scheduler of the relevant output port. In addition, it updates the stored values of VPI and VCI, using the header bus. The input controllers send information to an output scheduler in a TDM fashion since there is only one address bus.
6. The output scheduler updates its virtual queue accordingly.

If an OAM cell is received, it is forwarded to the OAM processor. If signalling cell is received, it is forwarded to the central μprocessor.

2.3.2 Control and Data Flow For Sending a Cell

Routing a cell from an input port to an output port is shown in Figure 2.6 and it proceeds in the following fashion.

1. Virtual queuing is done per Virtual Channel (VC) and each virtual queue is implemented as a FIFO.
2. The output scheduler in the output module selects a VC to be dequeued. It also reads a buffer address from the selected VC virtual queue.

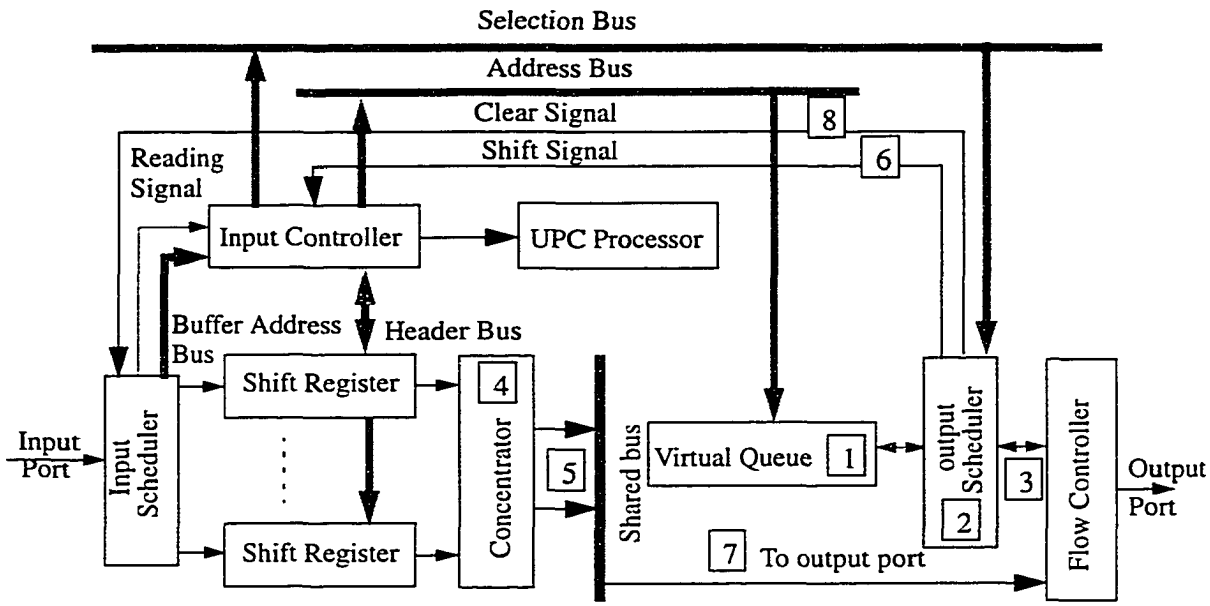

Figure 2.6 Control and data flows for sending a cell.

3. Messages are exchanged between the output scheduler and flow controller, which validate the credit balance of the selected cell. If no credit is available for that VC, another VC is selected.

4. If credit is available, the concentrators and the tri-state bus buffers are configured, based on the dequeued cell address, to connect between the desired buffer to the line of the shared bus that is connected to the desired output port. Each output port controls a concentrator in turn in a TDM fashion.

5. Access conflict is removed in the proposed scheme since the concentrators are driven by the outputs. The commonly-used input-driven fabric is subject to access conflict.

6. A shift signal is issued from the output scheduler to shift out the data stored in the shift register.

7. The cell is sent from a shift register to the desired output port through the configured path.

8. After sending the cell out, a clear signal is issued from the output scheduler to the input scheduler to free the reserved shift register.

2.3.3 Multicast and Broadcast Functions

One of the basic requirements in ATM switches is the support of multicast traffic. Many schemes have been developed for multicast [43, 31, 85]. The choice of multicast strategy depends on the switch architectural details. A conventional output buffer ATM switch needs only a simple multicast controller to support multicast, as shown in Figure 2.7, where a multicast cell is copied to all destination output buffers. Additional memory is needed as the multicast cell is copied many times.

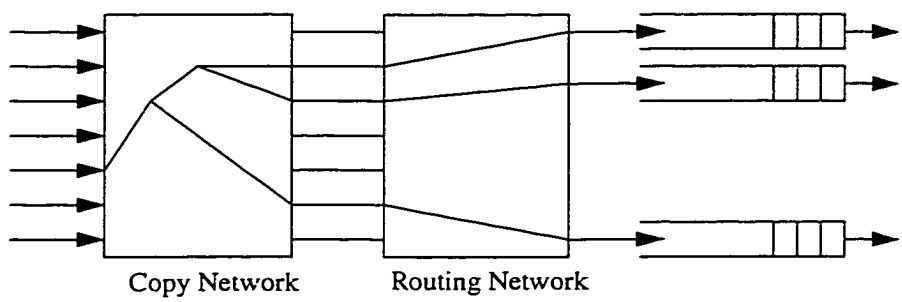

Figure 2.7 Multicast output buffer switch architecture.

For a shared memory switch a complex controller is needed since the multicast controller makes copies of the multicast cells before storing them in the shared memory [88, 84].

Multicast functionality can be easily supported in the proposed switch architecture. In the proposed multicast scheme, one copy of the cell is used to support the multicast functions. No moving or copying of a cell is required. When a multicast cell is received, it is stored in a shift register, as usual. The destination of the cell is checked by the input controller. All virtual queues of the desired output ports will be updated with the address of the multicast cell. The VCI and VPI fields of the stored cell will be replaced by a multicast identifier. This identifier will be mapped by the output module to the destined VCI and VPI. Any of the output schedulers can access the stored cell by controlling the concentrator

connected to the shift-register where the cell is stored. More than one output port can simultaneously access the cell by establishing a connection between a particular input of a concentrator and the bus lines connected to the desired output ports. The cell could be simultaneously sent to multiple output ports. Based on the multicast identifier which is stored with the cell, the output module prepares the outgoing VCI and VPI and sends them directly to the output port. For multicast cells, the shift register is used as a ring shift register. While a cell is shifted out, it is fed back and shifted in. The input controller counts the number of clear signals sent to the shift-register and frees it after forwarding the multicast cell to all destined output ports.

2.3.3.1 Memory Implementation for Shift Registers

Figure 2.8 shows a memory implementation for a group of parallel shift registers. The size of a shift register equals the size of the ATM cell. The memory depth equals the number of shift registers and the memory width equals to the ATM cell size. In a writing process, up to n cells can be written in parallel, assuming that the starting bits of all n cells are synchronized. At each system clock tick, up to n bits, from n cells, will be received by the memory write router. If the n bits received are the first bits of n cells, the memory write router selects n unused words and assigns each one of them to a cell. All the bits received from a cell will be stored in the memory word assigned to that cell. The memory address for writing the mth bit of a cell is defined by increasing the row address of first bit of that cell by m.

Routing a cell out is done by Memory Read Router which has a similar functionality to the Memory Write Router. It addresses the first bit of the cell and increases this address while reading.

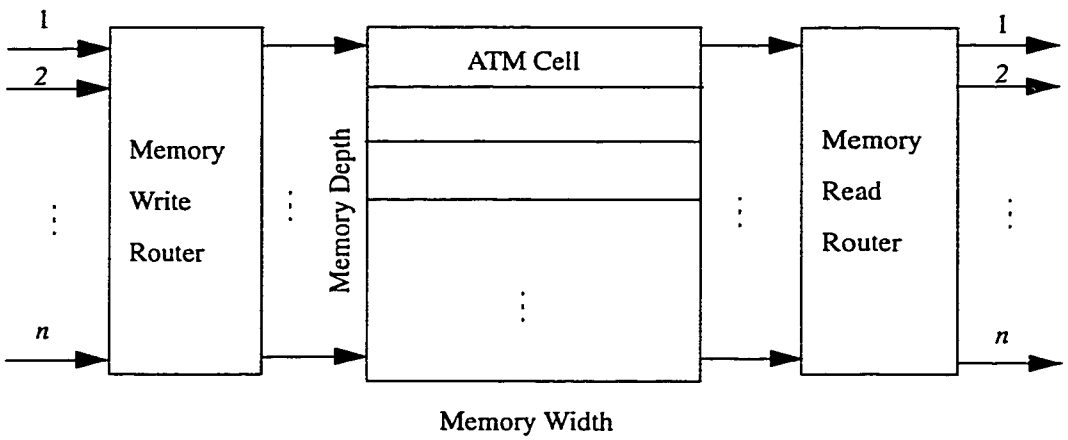

Figure 2.8 Memory implementation for shift registers.

2.3.4 Fabric Scalability and Switch Size

ATM switches will be used in different environments, including small and large LANs and WANs. Different switch sizes are required to work efficiently in these different environments. For cost efficiency, an ATM switch should be scalable and should efficiently satisfy both the low and high capacity requirements. The proposed architecture permits scalable from a small number of ports to a relatively large number. Increasing the number of supported ports will increase the capacity of the switch. The modular design of the proposed switch makes it easy to support a flexible number of ports. Updating the virtual queues of the output modules needs time multiplexing between the different input modules using the address and selection busses. The updating process involves writing the address of the stored cell into an output virtual queue. The width of the switching module shared bus increases with the increase of the number of ports. The complexity of the switching module and the virtual queue updating process put an upper limit on the number of ports.

2.3.5 Plane Function Requirements

ATM switches should be able to support four types of information flow: user, control (signalling), management, and traffic control flows. Each type has different functional requirements within the switch.

2.3.5.1 User Plane Requirements

The main function of an ATM switch is to relay user data cells from the input ports to the output ports. The VPI/VCI information derived from the cell header is used for that. The payload of the cell is carried transparently.

user cell flow

When a cell arrives at an input module, the input scheduler stores it in a shift register and informs the input controller. The input controller checks the header of the cell and sends the VPI and VCI values to the UPC processor which checks the validity of the received cell. For a valid cell, the input controller sends the stored cell address to the output scheduler of the destined port which updates the virtual queue. To drive a cell out, the output scheduler selects a VC based on the priority. The flow controller checks credit availability for the selected VC. If credits are available, the output scheduler gets the HOL cell address from the virtual queue. Based on the shift register address stored in the virtual queue, a concentrator is configured to connect the storing shift register to the destined output port via the shared bus.

2.3.5.2 Control Plane Requirements

The control plane supports AAL signaling above the ATM layer. Connection request and connection admission are carried by signaling cells which are identified by their VPI/VCI fields. Unlike user cells, information in the signaling payload is not transparent to the switch. Signaling information must be separated from user cells and processed by the switch.

Signaling flow

Signaling cells are identified by special VCI and VPI values. The input controller identifies signaling cells and forwards them to the central μProcessor. Signaling cells from various input ports are forwarded to the central μProcessor. Sending a cell to the μprocessor is similar to sending a cell to an output module. Based on resource allocations, the μProcessor accepts or rejects a connection request. When the μprocessor inserts a cell, it prepares

the cell and stores it in a local memory and sends its address to an output virtual queue. The process of scheduling a cell out from the μprocessor is the same as for scheduling a cell out from an input module.

2.3.5.3 Management plane requirements

The management plane monitors and controls the ATM network to ensure efficient operation. The Operation, Administration and Management (OAM) requirements for ATM networks are defined in ITU-T I.610. The payloads of OAM cells are processed by ATM switches. OAM cells may be generated by ATM switches and mixed with the outgoing user data.

OAM Flow

OAM cells are mixed with incoming traffic. OAM cells are identified by their VCI and Payload Type (PT) values. OAM traffic, similar to the user traffic, should obey the traffic contract. After storing an OAM cell, the UPC processor checks the validity of the cell. If the cell is valid, it is forwarded to the OAM processor. The OAM functionality is implemented by the OAM processor. If an OAM cell flow is terminated at this switch, the OAM cell is extracted and processed. The process of sending OAM cells to the OAM processor is equivalent to the process of sending a user cell to an output port. For the OAM processor to insert an OAM cell, it prepares and stores the cell in a local memory and updates the virtual queue of the destined port. The process of sending a cell out from the OAM processor is similar to sending a cell out from an input module.

2.3.5.4 Traffic Control Plane Requirements

Traffic control monitors and regulates traffic to prevent and control congestion. The main function of the traffic controller is usage/network parameter control (UPC/NPC) and congestion control. UPC/NPC forces traffic sources to follow traffic contracts. Congestion control prevents/recovers from congestion. Resource management cells transfer information about the network status between switches.

Resource management cell flow

Resource Management (RM) cells are identified by their headers. They are switched, just like user cells, from the input module to the desired output module. Unlike user cells, resource management cells are not sent directly to an output port. Instead, they are extracted by the flow controller. The flow controller analyzes RM cells and updates the credit balance based on information carried on them. When the flow controller wants to insert a RM cell, it prepares the cell, stores it in a local memory, and sends an insertion request to the output scheduler. The output scheduler decides when the cell should be sent.

2.4 Performance Evalution

Before implementing a switch, the performance of the switch should be evaluated. Three main methods can be used to evaluate the performance of a switch:

- Computer simulation
- Mathematical modeling
- Hardware prototype

Each of these methods has its pros and cons. Computer simulations can be done easily but it does not measure the performance accurately. Mathematical models are more accurate than computer simulations but it is hard to be driven and to simplify the derivation process some assumptions need to be done and this reduces the accuracy of the models. Hardware prototype gives better performance evaluation than the other two methods but it requires high implementation cost.

In order to get conference in shift-register based switch architecture, Dr. Fayez ElGuibly's group has built mathematical models and done computer simulation for the shift-register based switch [103, 104, 105]. The performance evaluations got from the two methods are very close and this gives confidence of the proposed architecture.

In order to show the superiority of the proposed switch, simulations have been done to compare the performance of the proposed switch with the performance of the three main types of the switch architectures proposed in the literature: input buffer, output buffer, and

shared buffer. Cell loss of the proposed switch is much less than input or output buffer switch for the sane number of buffers per port. The performance from the proposed switch is slightly worth than the complete shared buffer switch. However the performance of the switch can be improved by increasing the buffer size [103, 104, 105].

2.5 Conclusions

In this chapter we introduced a shift-register based ATM switch architecture. The architecture uses shift registers as input queues and this overcomes the need for serial-to-parallel and parallel-to-serial format conversions. In addition, the switching speed is improved over output buffer based switches. This is because, after the cell is stored it is not moved from the storing shift register until it is sent out. In addition, the parallel storage structure avoids the HOL blocking problem which happens with conventional input queues. The access time of the storage memory becomes independent of memory size. Switch scalability and multicast functions are supported in this architecture. Using virtual output queues makes it feasible to use any kind of priority scheduling techniques without having to move the cells from their input registers.

In the next chapter we will introduce a priority scheduling scheme which satisfies the different QoS requirements. The proposed scheme works with any output buffer ATM switch architecture.

Chapter 3

A Window-Based Cell Scheduler

The major potential of high-speed ATM networks is the efficient and flexible carriage of different kinds of traffic and the offering of diverse services. ATM networks have to handle a wide range of traffic characteristics and performance requirements. This can only be realized when an effective hardware switch with a good priority scheduler is available. Prioritized cells should be handled in an appropriate manner during cell discarding and scheduling. The QoS of an ATM connection depends on the cell loss, the cell delay and the delay variation incurred by the cells belonging to that connection.

For nonblocking switches, such as the shift-register based ATM switch architecture proposed in Chapter 2, cell delay is completely controlled by the priority scheduler. In this chapter, we propose a cell selection scheme for multiple priority services. The proposed scheme satisfies multiple QoS requirements. Cells from different service classes are stored in different priority queues. In order to maintain the QoS of each class, the higher-priority queues can send a predefined number of cells before the lower priority queues are served. When a particular priority queue consumes its maximum allowable number of cells, other queues with lesser priority are allowed to send. The proposed scheduler spreads out the delay variations across the queues depending on their priorities.

This chapter is organized as follow. ATM traffic types are explained in Section 3.1. The different models that can be used to represent ATM traffic are introduced in Section 3.2. The problem of scheduling different priority services is stated in Section 3.3. The proposed window-based scheduling algorithm is introduced in Section 3.4. Simulation

results that show the effect of the scheduling protocol on the performance metrics are provided in Section 3.5. Comparisons between this scheme and an alternative are also discussed in Section 3.5. The conclusions for this chapter are provided in Section 3.6.

3.1 ATM Traffic Expectation

Four classes of data have been defined by ITU-T. Class 1 is time-sensitive, constant bit rate connection-oriented data. Class 2 is time-sensitive variable bit rate, connection-oriented data. Class 3 is time-insensitive, variable bit rate, connection-oriented data. Class 4 is variable bit rate connectionless data.

A typical representative for the first class is digital telephony or voice. The principal requirement of voice services is bounded delay. Voice packets that are not delivered to their destinations within a certain period have to be discarded. The maximum tolerable end-to-end total delay is in the range of 100-600 ms [91]. Although voice traffic may have a bursty nature, coding techniques such as Pulse Code Modulation (PCM) and Adaptive Differential Pulse Code Modulation (ADPCM) result in a constant bit rate of 64 and 32 K-bits/s respectively [91].

Class 2 services have been created mainly for classifying variable bit rate video. The end-to-end delay requirements for video services depend on the type of the service; for one-way television applications such as broadcast TV, delay is irrelevant when compared to that of two-way applications like video-conferencing. On the other hand, video as an information carrier is complementary to sound, and must stay synchronized with it. The acceptable cell-loss rate depends on the coding algorithm and the amount of motion in the scene. The bursty nature of video traffic does not seem to impose any particular problem. When many video sources are combined in one virtual path, the total rate exhibits less burstiness.

Typical examples of Class 3 services are interactive communications. Interactive data traffic is highly bursty and has moderate end-to-end delay requirements. In addition,

although packet loss may be tolerable, it is better to be avoided. Interactive data can have moderate priority.

Class 4 corresponds to connectionless data services. This type of service is similar to those of user datagrams in conventional packet switched networks. Connectionless services are expected to generate highly bursty traffic with no guaranteed maximum bandwidth. Therefore the most reasonable choice would be to treat connectionless services as mass data transfers, i.e. assign them low priority.

All classes of traffic can be transported by ATM networks. It is important to acquire knowledge of this traffic to determine network resource allocation for each service call.

3.2 Traffic Models

Traffic characteristics supported by ATM networks differ from one service to the other. The burst length of data is service dependent [92]. Hence, the Poisson model is not a valid model to simulate cell arrival processes. In this section, we will introduce some ATM traffic models which will be used later in this chapter to simulate the behaviour of the proposed window-based scheduling algorithm.

A sporadic source (burst-silence) is a realistic ATM traffic model for interactive data communication. To model this ON/OFF periodic process, not only generation of cells during the ON period should be examined carefully but also traffic during the OFF period has to be explored.

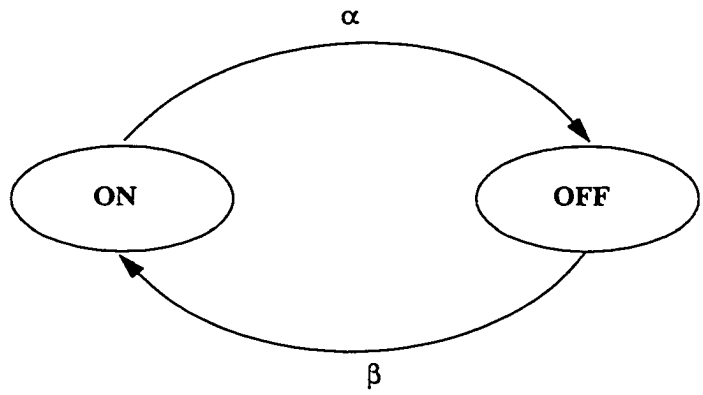

Figure 3.1 Two-state Markov process.

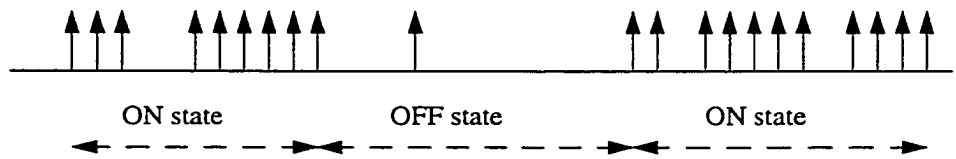

Figure 3.2 Interactive data source model.

Figure 3.1 shows a two-state Markov process model. The two states represent the ON and OFF states of a source. During the ON state, a source periodically sends a cell with probability P_{on}. A source can send cell in the OFF with a rate much less than the sending rate in the ON state. The probability of sending a cell in OFF state is P_{off}. Figure 3.2 illustrates the sending sequence in both ON and OFF states. The source stays in the ON or OFF states for a duration approximated by a Poisson distribution. The mean time of the ON and OFF states are $1/\alpha$ and $1/\beta$ respectively. Hence, the transition rates from ON to OFF state and from OFF to ON state are α and β, respectively as shown in Figure 3.1.

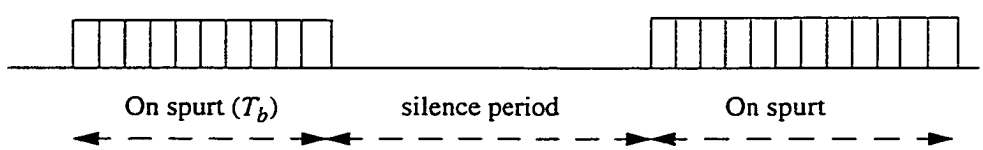

Figure 3.3 Voice source model.

This ON/OFF model can be used for modeling a voice source. The two-state Markov process is used to represent the talk spurt and silent states. The source in the burst state emits cells with a constant cell interval T_b. No cells are emitted in the silent state. T_b is between 5 ms and 15 ms according to the encoding rate [93]. Figure 3.3 shows the voice source model as described in the literature [93].

For N multiplexed voice sources, an N state Markov process is a good approximation. In this model, the data rate changes between N rates. The rate increment A and the transition rates α and β are chosen to match the mean, variance, and autocovariance function of the multiplexed sources. Figure 3.4 illustrates an N state Markov process that represents N multiplexed voice sources.

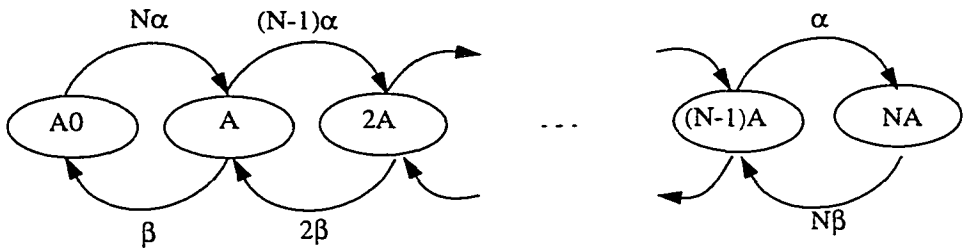

Figure 3.4 N-state Markov model for voice traffic.

A two-dimensional Markov process can be used to model encoded video traffic, such as, broadcast television, video conferencing and video telephone. An $N_1 \times N_2$ Markov model is shown in Figure 3.5. The data rate of the multiplexed sources may change among different fixed-rate levels. Basically, there are two data rates that are the main building components of the model: a high-rate depicted as A_h, and a low-rate depicted as A_l. Different combinations of these two rates produce a wide range of discrete data rates. The maximum data rate that can be reached is $N_1 \times A_l + N_2 \times A_h$.

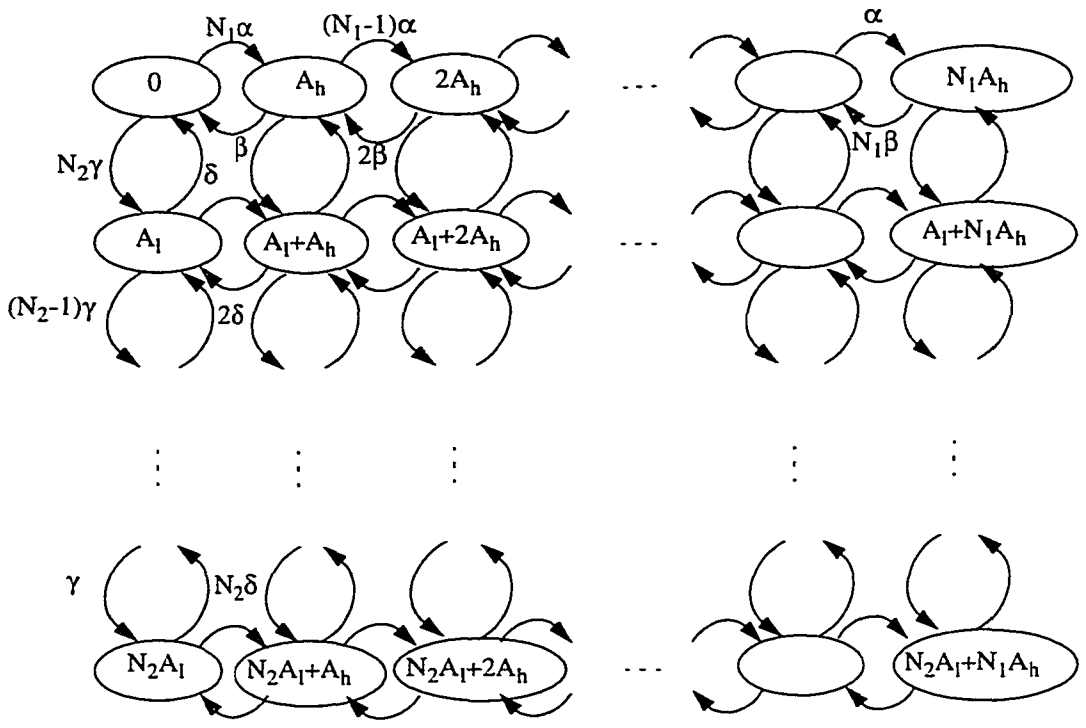

Figure 3.5 N_1 x N_2 Markov model for video sources.

3.3 Priority Control Problem Statement

The scheduling policy influences the amount of buffer size needed to guarantee a given loss probability bound. The necessary buffer size can be computed from the values of the local delay bound [94]. Thus, as long as we can compute the delay bound for a given scheduling policy, the cell-loss probability can otherwise be ignored.

The scheduling policy must not starve (i.e., ignore forever) any cell present inside the switch at any time. Further, in a real-time environment, the scheduling policy must satisfy the requirements of real-time and non-real-time services.

Static priority schemes usually lead to starvation in the switch. In other words, higher priority classes tend to monopolize the output links, severely degrading the service of lower priority classes. The proposed window-based scheduling approach reduces latency on the

oppressed lowest priority queues by relaxing the stringent delay requirements on the higher priority queues. It is also anticipated that there will be some improvement in the memory requirements for the lowest-priority classes.

3.4 Proposed Window Scheduling Algorithm

The selection of a particular scheduling discipline for an ATM cell scheduler involves a trade-off between the need to service a large number of connections with diverse delay requirements and the need for simplicity in the scheduling operations. We consider a few basic scheduling disciplines and propose some alterations to suite different QoS requirements.

A First-Come-First-Serve (FCFS) scheduler transmits cells in the order of their arrival. Since the maximum delay in a FCFS scheduler is the same for all connections, all connections have an identical delay [24]. There are no priorities supported in this protocol. A cell priority is defined by the cell arrival time. This protocol is not suitable for ATM networks where many services with different priorities are supported. It is unacceptable to assign the priorities solely based on cell arrival times.

In a Head-Of-the-Line (HOL) scheduler, each connection is assigned a priority, where a lower-priority index indicates a higher priority. A HOL scheduler maintains one FCFS queue for each priority level, always selecting the first packet in the highest priority FCFS queue for transmission. If no cells in the highest priority queue are present the second higher priority queue will be served [39]. This protocol may lead to service starvation for lower priority services, because it keeps on serving the high priority services for as long as it has cells. The HOL protocol may provide very high QoS for the high priority services but the protocol will not be able to satisfy the QoS required for lower priority services.

The idea of round-robin scheduling, in general, is that a server process circularly and repeatedly visits a number of clients and performs one job for each of them. Visiting the VCs in a round-robin fashion and forwarding one cell from each ready VC is, in princi-

ple, fairer than FCFS muitiplexing. This protocol assume having equal priorities for all supported services. However, this mechanism should not treat all VCs equally, but rather giving a weight factor for each VC based on its priority [88].

We introduce a scheduling scheme that satisfies the requirements of ATM traffic. The scheme chooses cells for dispatching from the priority queues in an ascending order. Cells from the higher-priority queues are selected before the lower-priority queues up to a certain maximum number of cells and within a certain period of time (window of time slots). When a priority queue consumes its maximum allowable cells, other priority queues with lower priority are serviced. No priority queue can exceed its maximum allowable number of cells within a predefined window of time slots unless all other queues are empty or the lower priority queues have exceeded their maximum allowable number of cells. The scheduling algorithm is described as follows:

- Divide output traffic into *max* priority queues.
- Define n_i $(0 \leq i < max)$ as the maximum number of cells that can be sent from each queue. Where $i=0$ is the index of the highest priority queue.
- Define T_i as the window frame for priority queue Q_i within which it can send at most n_i cells. T_i can be written as

$T_i = T_c n_i$ where T_c is the time required to transmit a cell.

- Define the time required to service all the supported priority queues as the window frame time (T)

$$T = \sum_{i=0}^{max} T_i$$

- Define the number of cells that have already been sent during a window frame (T) from the ith priority queue as x_i.
- The operation of this schedule is described in the following section.

3.4.1 Window-Based Scheduler Flow Chart

The Flow chart for the window based scheduler is shown in Figure 3.6. The D_i flag indicates if the ith queue has cells or not. The D_i flag equals *1* if one cell or more is queued. The D_i register carries the number of cells that is stored in the ith queue. The W_i flag indicates if the ith queue has already sent its allowable number of cells or not. The W_i flag equals *1* if the ith queue has not yet sent a number of cells equal to its allowable number. The W_i register carries the number of cells that the ith priority queue can send during a window time frame (T). *MAX* is the maximum number of priority queues supported.

When a cell with priority x arrives, it is stored in the xth queue, the D_x register is incremented queue and the D_x flag of this queue is set. The process of scheduling a cell out starts by checking the highest priority queue. If the highest priority queue did not use all its allowable number of cells or none of the lower priority queues can send, a cell will be scheduled from the highest priority queue. If the highest priority queue can not send the check is done for the second highest priority queue and so on. When a cell is scheduled from the ith queue, The W_i register is decremented to indicate that this queue has used one of its allowable number of cells. If W_i becomes zero, the W_i flag is cleared to indicate that this queue has finished its allowable number of cells. The D_i register is also decremented. If the D_i register is zero, the D_i flag is cleared to indicate that there are no more cells to be sent from this queue. The W registers and flags are initialized every window frame time.

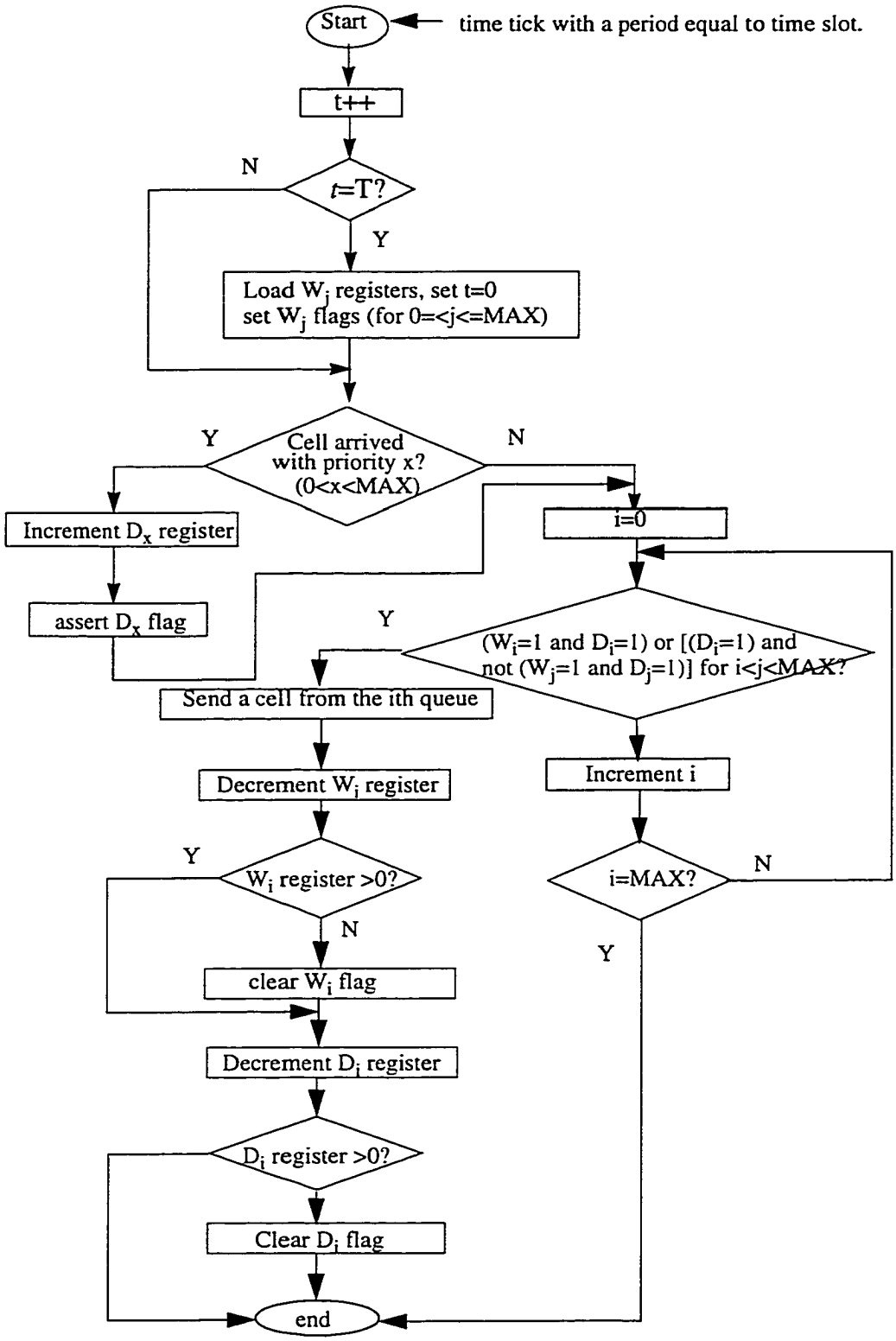

Figure 3.6 Flow-chart for the scheduling algorithm.

3.4.2 Hardware Design Implementation

The hardware configuration of the proposed scheduler for one output port is shown in Figure 3.7. There is a controller for each priority queue. This controller is responsible for updating the D and W flags which are used by the main port control. The flow-chart for updating the flag is shown in Figure 3.6. A set D flag indicates thst a queue contains cells. A cleared W flag indicates that a queue has already sent its allowable number of cells. At the beginning of each time slot, the main port controller indicates which queue will send. It takes the D and W flags from different queues and allows only one of these queues to be active via control signals (A_i), with A_0 controlling the highest priority queue.

For four priority queues, the equations that govern this controller can be written as follows.

$$
\begin{aligned}
A_0 &= D_0W_0 + D_0D'_1D'_2D'_3 + D_0D'_1W'_2W'_3 \\
&\quad + D_0W'_1W'_2D'_3 + D_0W'_1D'_2W'_3 + D_0W'_1W'_2W'_3 \\
&\quad + D_0D'_1W'_2D'_3 + D_0D'_1D'_2W'_3 + D_0W'_1D'_2D'_3 \\
A_1 &= D'_0D_1W_1 + D'_0D_1W'_2W'_3 + D'_0D_1D'_2D'_3 \\
&\quad + D'_0D_1W'_2D'_3 + D'_0D_1D'_2W'_3 + W'_0D_1W_1 \\
A_2 &= D'_0D'_1D_2W_2 + W'_0D'_1D_2W_2 + D'_0W'_1D_2W_2 \\
&\quad + W'_0W'_1D_2W_2 + D'_0D'_1D_2D'_3 + D'_0D'_1D_2W'_3 \\
A_3 &= D'_0D'_1W'_2D_3W_3 + D'_0W'_1D'_2D_3W_3 + D'_0D'_1D'_2D_3 \\
&\quad + D'_0W'_1W'_2D_3W_3 + W'_0D'_1W'_2D_3W_3 + W'_0D'_1D'_2D_3W_3 \\
&\quad + W'_0W'_1D'_2D_3W_3 + W'_0W'_1W'_2D_3W_3
\end{aligned}
$$

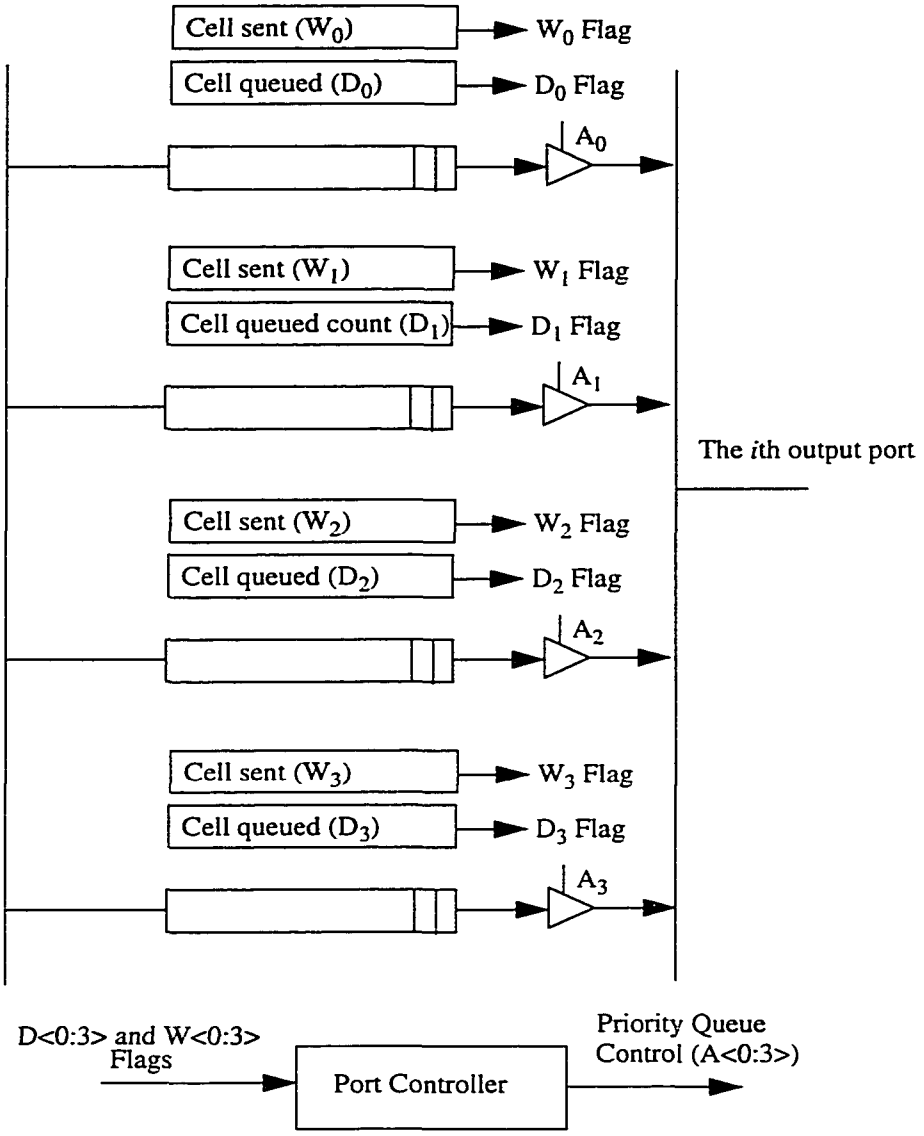

Figure 3.7 Hardware block diagram of the priority scheduler.

3.4.3 Window-Based Algorithm Versus Weighted Round-Robin

The weighted round-robin (WRR) algorithm has been developed to achieve fair allocation of bandwidth between different queues based on their priorities. Many variations of the WRR algorithm have been developed [88, 98, 99]. The idea of the WRR algorithm, in general, is that a server process circularly and repeatedly visits a number of different pri-

ority queues and schedules N_i cells for the ith queue. N_i is a weighting factor that differentiates between different queue priorities. The flow chart for the weighted round-robin algorithm is shown in Figure 3.8.

The main difference between a window-based algorithm and the WRR is how the scheduler behaves when the number of cells available for the ith queue is less than its allowed value (N_i). For WRR, the server moves to the $(i+1)$th priority queue and will return to the ith queue only after serving all other priority queues. In other words, the bandwidth that was allocated to the ith queue will be partially lost. This scheme may introduce a delay for cells from a high-priority queue as they may arrive just after the server process is moved to the next lower priority queue and they will not be served until all other lower priority queues have been served.

For the window-based algorithm, if the number of cells available for the ith queue is less than its allowed value, the server process will move to the $(i+1)$th queue and serve only one cell from that queue. After that it will return again to the ith queue. If there is a cell available, the scheduler will serve it, otherwise it will continue serving the $(i+1)$th queue. This scheme assures low delay for the high priority queue even if the arrival of its cells is distributed.

As the shaded box in Figure 3.6 shows, the window based scheduler always starts the scheduling process from the highest priority queue. The round robin algorithm starts from the queue it was serving in the last time slot. This feature in the window based scheduler facilitates easy support for real time services regardless of their cell-arrival distribution and the number of priority queues supported.

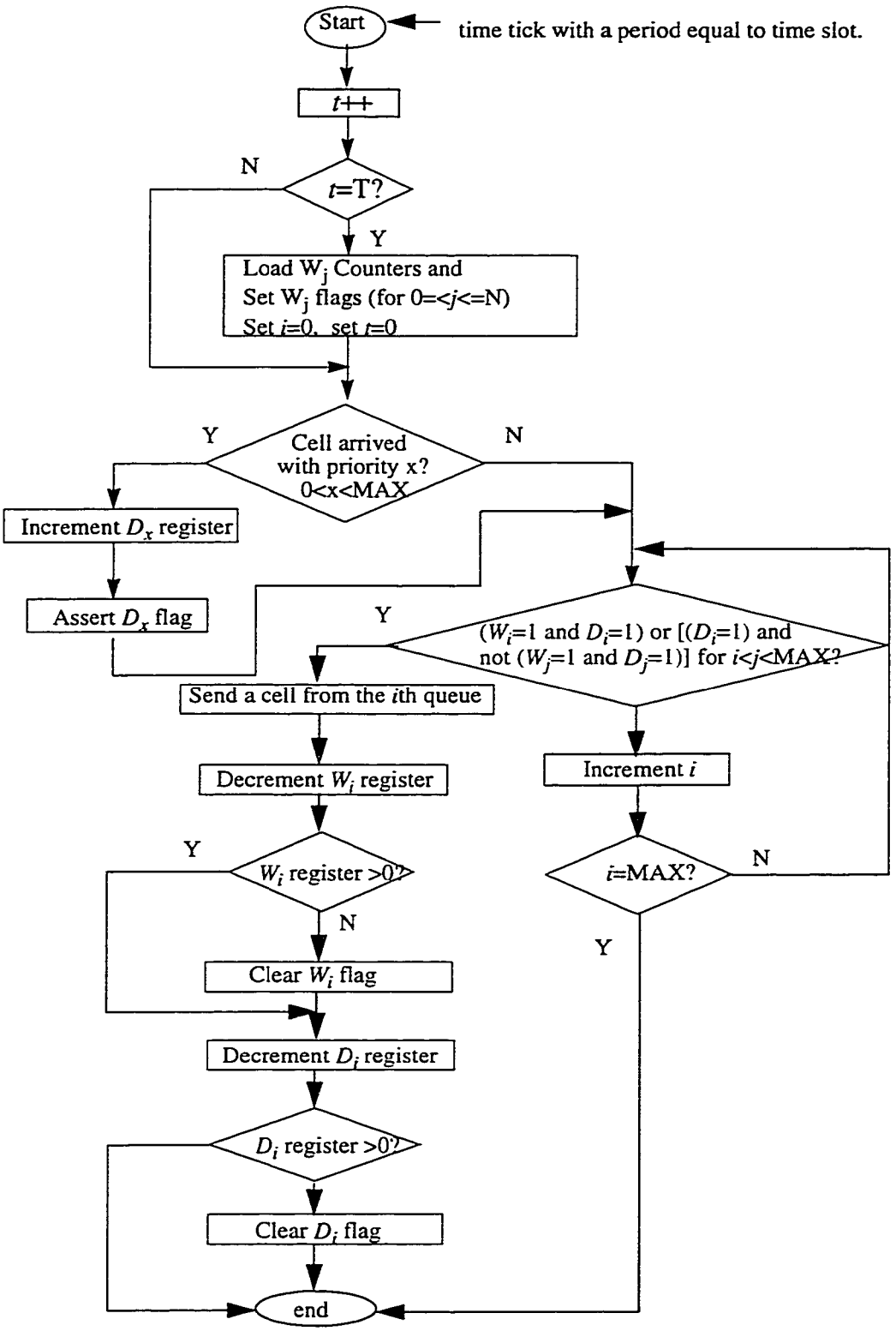

Figure 3.8 Flow-chart for the weighted round robin scheduling algorithm.

3.5 Simulation Analysis and Results

Simulation is useful to show the effectiveness of an architectural detail for an ATM switch and remains the most flexible method for examining switch operation. An ATM switch implementing the window scheduler is simulated under realistic traffic combination of voice, video, and data. We are particularly interested in switch buffer requirements and cell-delay performance. HOL provides the best performance for the high-priority queue when compared to the other existing priority scheduling schemes. A comparison between the proposed window based scheme and the static priority (HOL) is pursued to illustrate that the proposed scheme achieves a better result for lower-priority queues with a negligible degradation for the higher priority queues. Graphs are plotted to emphasize the importance of certain tuning parameters to the scheme such as window size, weight assignment per priority class, and load.

3.5.1 Simulation Assumptions

The traffic for each port is assumed to be a uniform multiplexing of data, voice, and video traffic. Four main traffic sources are used for this simulation. Each source belongs to a priority class.

- Serious-priority class is modeled by an $N_1 \times N_2$ Markov model as a representation of bursty video sources. This can be mapped to class B of ATM traffic.
- High-priority class is represented by an N state Markov process which captures the behavior of multiplexed voice sources. This can be mapped to class A of ATM traffic.
- Both medium and low-priority are modeled by two-state Markov processes, each of them having its own burst rate β, and mean emission rate μ. This can be mapped to class C and D of ATM traffic.

The traffic from the four sources are multiplexed together so that the ratio between any two traffic sources is 1:1.

We simulated a switch of size 16x16. The switch is internally non-blocking. Cells coming to an input port are sent to output ports based on a uniform distribution. There are

four priority queues per output port, each belonging to a priority class. The switch load can be changed by changing the load at the input ports.

The time window frame is selected to be 10 and the number of allowed cells for each priority is given as 4, 3, 2, and 1 for serious, high, medium, and low priority queues respectively.

The simulation parameters are given as:

Voice active mean (α) = 352
Voice idle mean (β) =652
Voice rate (A) =12
Video active mean high (α) = 352
Video idle mean high (β) = 600
Video active mean low (γ) = 352
Video idle mean low (δ) = 600
Video high rate (A_h) = 3000
Video low rate (A_l) = 825
Data active mean (α) for first ON-OFF model = 110
Data idle man (β) for first ON-OFF model = 12
Data active mean (α) for second ON-OFF model = 100
Data idle man (β) for second ON-OFF model = 10

Please refer to the simulator models presented in the section 3.2 for the meaning of these parameters. The simulation program is written in C and the simulation was done in a UNIX environment. The HOL protocol is simulated as the flow chart shown in Figure 3.9. The same traffic parameters that are used for the window-based protocol are used for the HOL scheme.

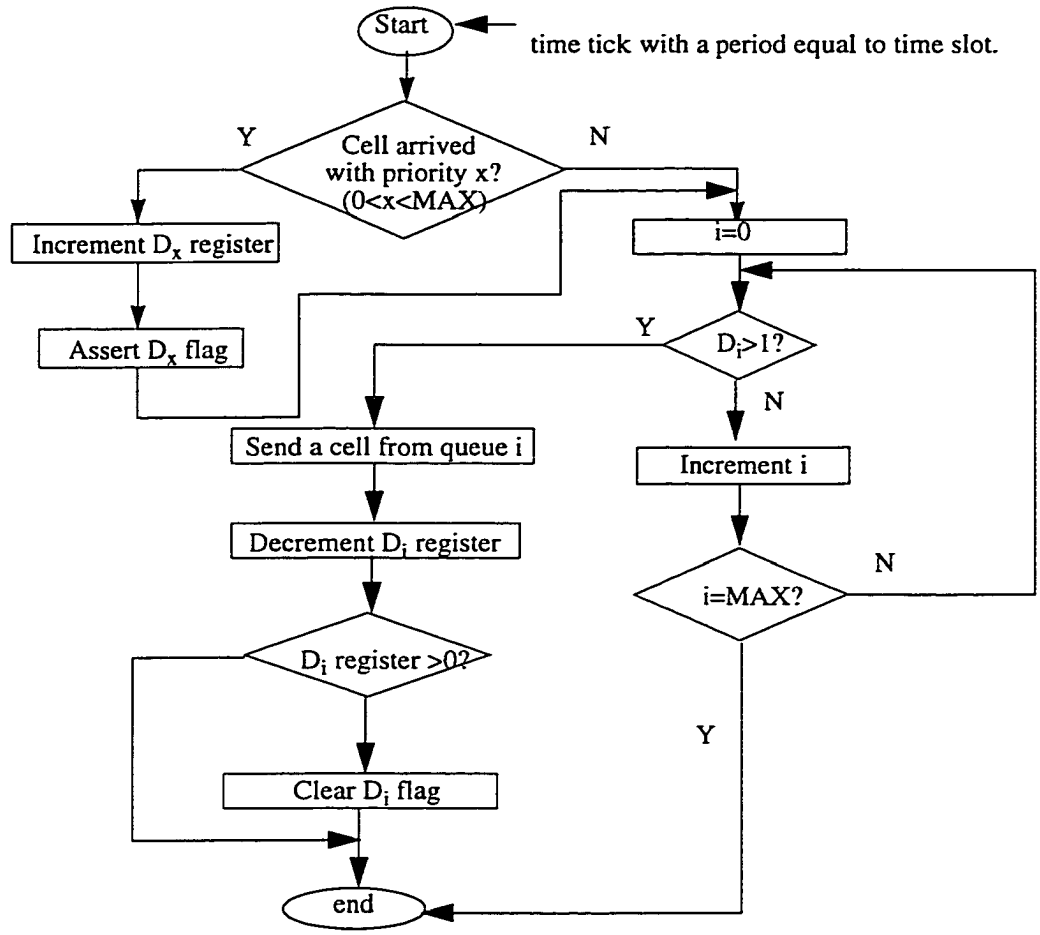

Figure 3.9 Flow-chart for the HOL algorithm.

In the following subsections, we analyze the results obtained for cell-delay and present the effect of window size.

3.5.2 Load and Latency

Figure 3.10 shows the resulting cell delay versus load for the window-based scheduling scheme, where load is defined as average number of cells that are received by the switch in a time-slot divided by the maximum number of cells that can be received in a time-slot. It is obvious that the higher priority cells exhibit less delay than the lower priority cells. The delay is fairly distributed between the different queues with a factor proportional to

the priority of each one of them. The serious-priority queue still has the smallest cell-delay to maintain its QoS. Cell-delay for the low-priority queue is also small enough to achieve a good performance.

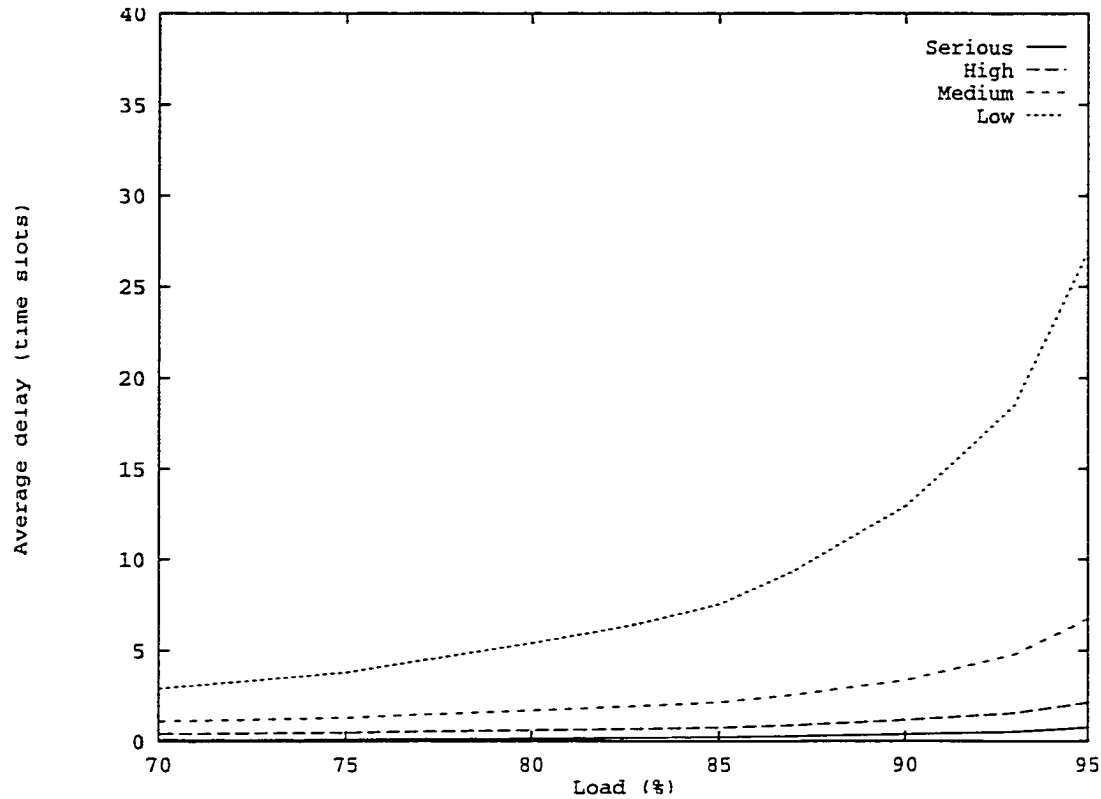

Figure 3.10 Window scheduler performance for four classes.

Figure 3.11 shows a plot for delay versus load for the serious priority class for both HOL and Window schemes. Comparing the Window scheduling scheme with the HOL scheme, we notice a small increase in the delay for the higher priority class as the offered load increases. HOL has an almost fixed delay for the serious class, while the Window scheme slightly increases the delay as the offered load increases for the same class. It is worth noticing that the average increase for the serious priority queue in the window-based scheme is less than one cell-delay, which is almost negligible.

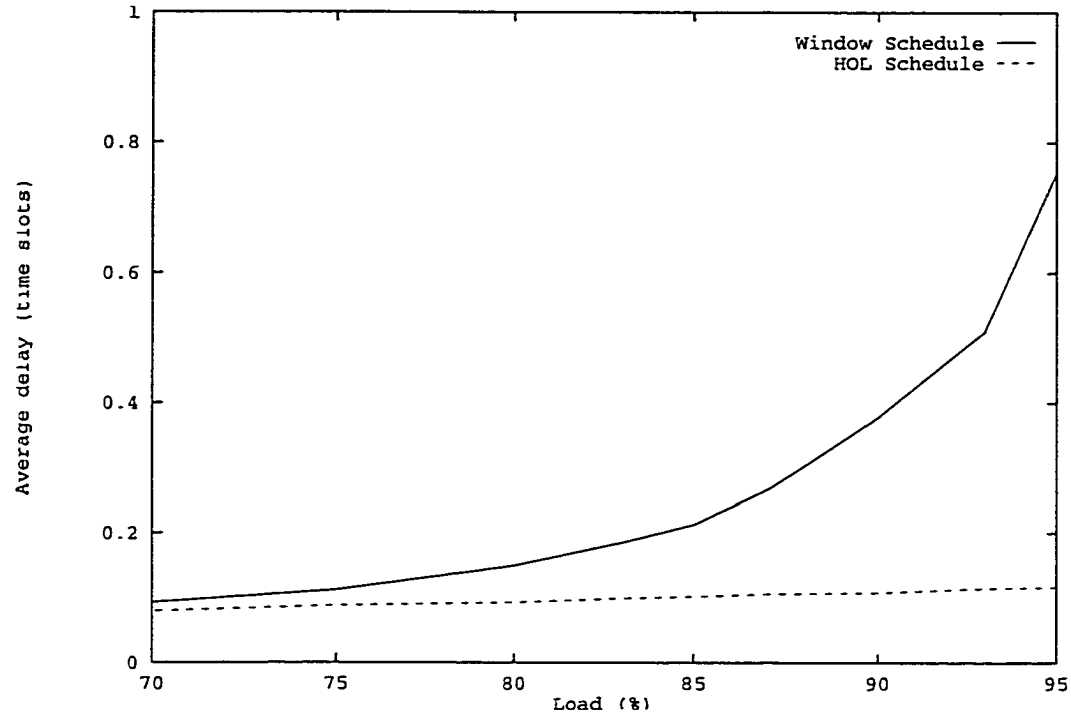

Figure 3.11 Serious class: Window vs. HOL (delay-load).

Figure 3.12 shows a plot for the low priority class for both schedulers. HOL has poorer cell-delay performance as the offered load increases. The window scheduler improves the cell-delay performance for this class. The improvement is remarkable for high load. This improvement comes at the cost of less than one cell-time delay for the serious class.

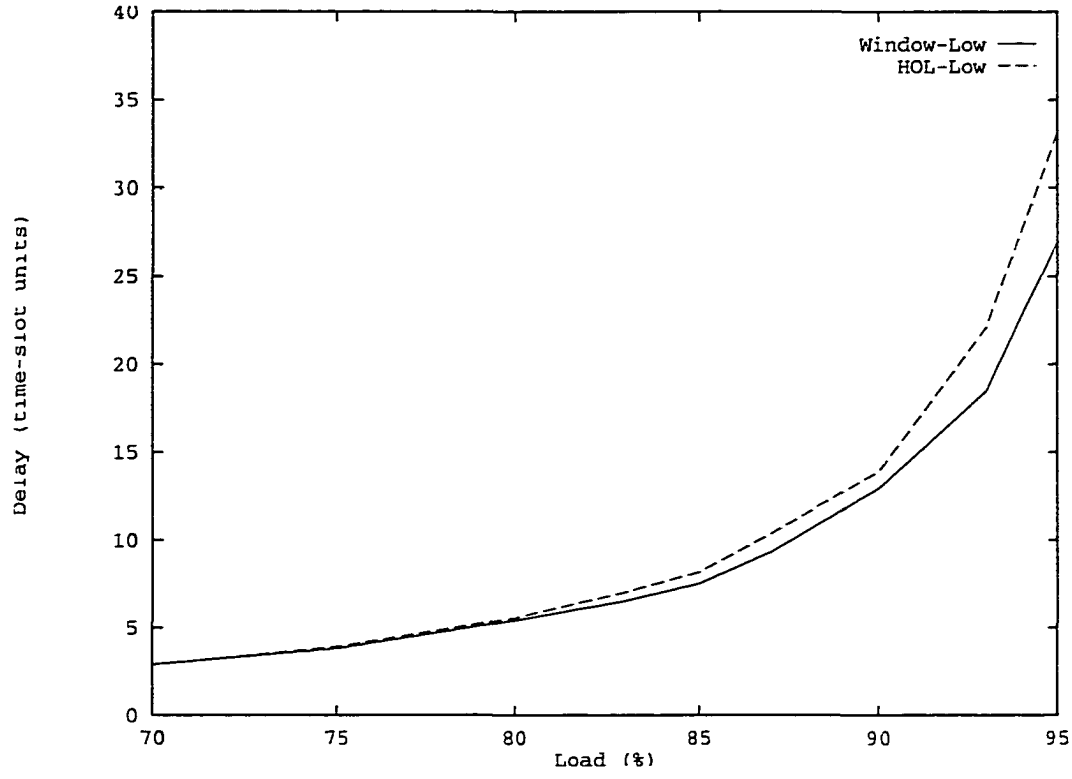

Figure 3.12 Low class: Window vs. HOL (delay-load).

3.5.3 Effect of Window Size on Cell Delay

The Window scheduling scheme ensures that a relatively fair service has been offered to all priority queues. To monitor the effect of the window size (in time-slots), simulations were run for different window sizes and the delay behavior was recorded.

Figure 3.13 illustrates the effect of window size on delay for the proposed scheme for an offered load of 95%. Serious, High, and Medium classes exhibit an increasingly lowered cell-delay. Concurrently, the low-priority class exhibits a belated increase in cell-delay as the window size increases. In fact, as the window size increases, the scheduling scheme approaches the performance level of the HOL scheme of severe delay for the Low priority class and very little delay for the higher classes. For a very large window size the performance of the Window-based scheme is very close to the HOL scheme as high-priority cells will be always served before the low-priority cells.

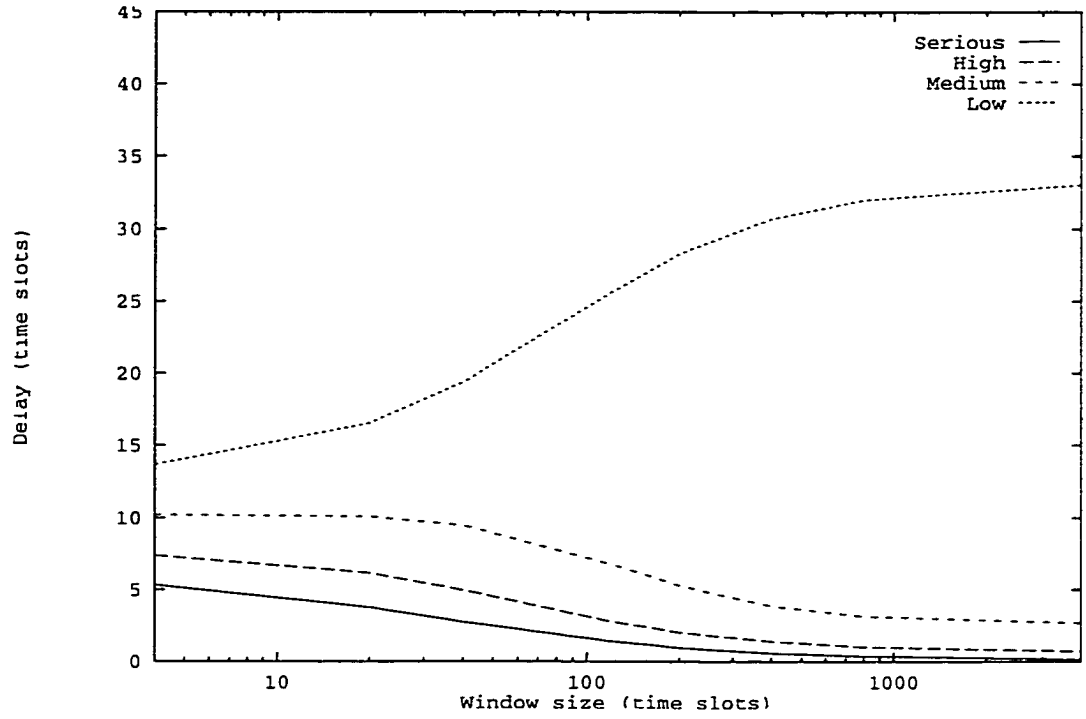

Figure 3.13 Effect of window size on the delay (load = 95%).

Figure 3.14 shows the effect of the window size for Low-priority class queues. Window size versus cell-delay has been plotted for input load of 75%, 85%, and 95%. It is intuitive that as the load on the switch increases, cell delay increases. However, as we reduce the offered load on the switch, cell-delay tends to stabilize irrespective of the window size increase.

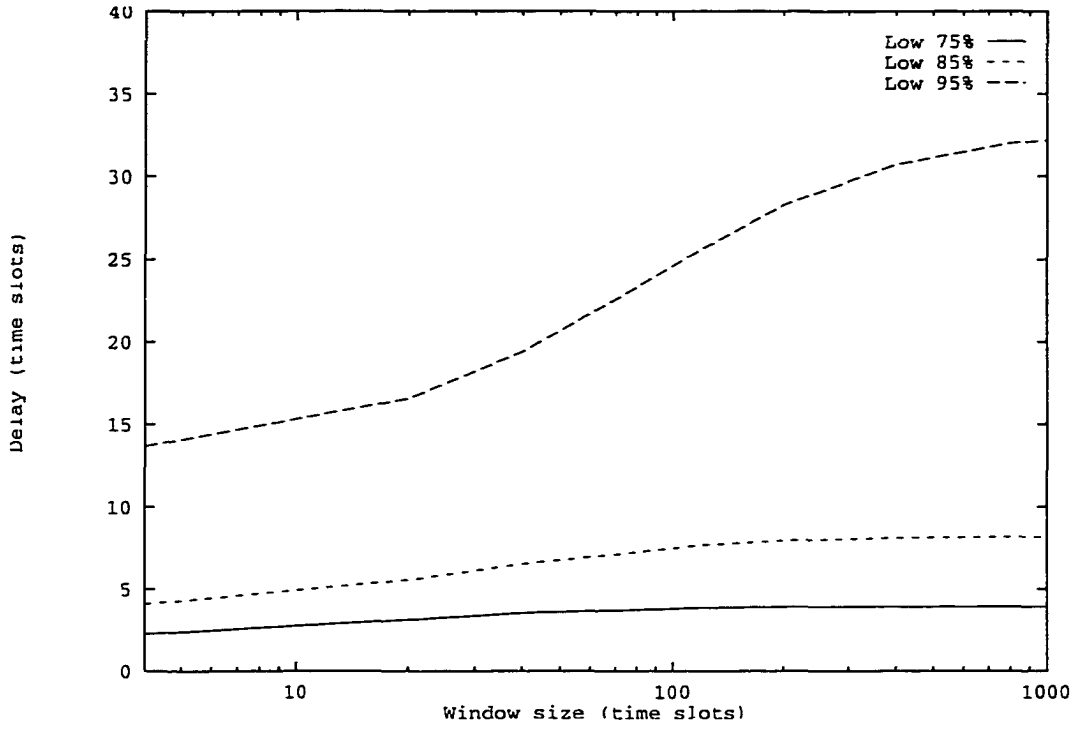

Figure 3.14 Window size versus delay for low class.

3.5.4 Cell Buffer Requirements

Under a 90% load condition we now find the cell buffer size required to have zero cell-loss. The maximum buffer allocation has been recorded for each priority queue. Simulation results that have been captured in Table 3.1 depict the minimum queue sizes required for the window-based scheme to achieve zero loss under 90% load. The minimum queue size is recorded as the maximum number of cells that are stored at any point of time during the simulation. The lowest priority queue needs the greatest buffer size. The difference between the size of buffer required for low-priority and high-priority queues is rather small.

Figure 3.15 shows the buffer size required to achieve a certain limit of cell-loss for the four supported classes under 85% load. The buffer distribution for the different classes is unequal. The low-priority queue needs more buffer than for the higher priority queue to achieve the same cell-loss. Once again, the difference between the size of the buffer

required for low-priority and high-priority queues is small.

Queue Priority	Minimum Queue Size
Serious priority	25
High priority	27
Medium priority	41
Low priority	63

Table 3.1. Maximum queue sizes.

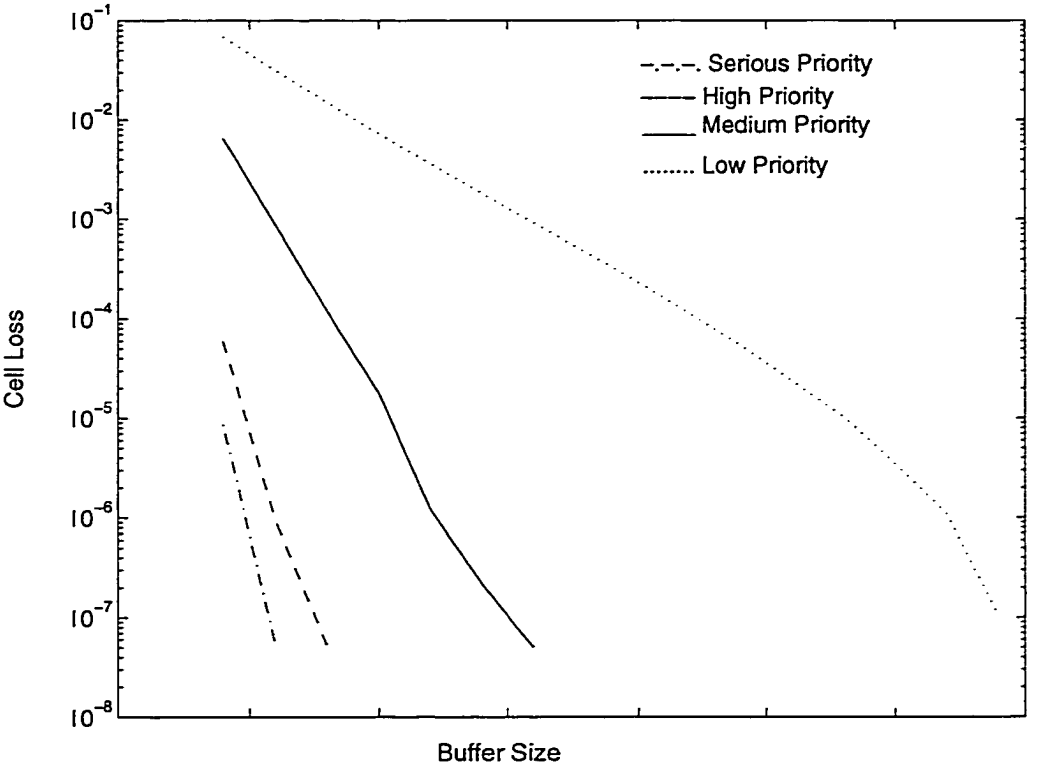

Figure 3.15 Effect of buffer size on cell-loss for different QoS (load=85%).

3.6 Conclusions

In this chapter, we proposed a window-based approach as a cell selection mechanism for ATM switches. The scheduler slightly relaxes the delay requirements in the highest priority class in order to greatly improve cell delay in the lower priority classes. The cell delay is distributed in proportion to the QoS of the different queues supported. The hardware implementation and buffer size required for this scheme were discussed. This scheme guarantees fair allocation of the total link capacity to low-priority traffic. It also maintains the QoS of the high-priority queues. Simulations with realistic bursty traffic models showed the effect of the algorithm in reducing the delay for lower priority classes. It also showed how the window size can affect the cell-delay of ATM cells for different input loads.

Chapter 4 uses the shift-register switch architecture which has been introduced in chapter 2 along with the cell scheduler proposed in chapter 3 to build a high-capacity switch which can support traffic in the range of terabits per second.

Chapter 4

A High-Capacity ATM Switch Architecture

Terabit switches are required to support the great traffic volume anticipated in ATM networks [72]. A terabit switch is defined as a high-capacity switch that receives and sends cells at the rate of terabits per second. The capacity of the switch is the sum of its input port capacities. A port capacity is defined as the external link capacity connected to that port. Three high capacity ATM switch architectures are introduced in this chapter. The first and second proposed switch architectures can support a great traffic volume but they have some technology and cost limitations that prevent them from being scaled up to the terabit capacity. The third architecture can be scaled up to the terabit capacity using current technology with a reasonable cost.

This first proposed switch architecture is a shared-memory switch. The memory has been designed as a wide-word memory to increase the read/write accessing speed. The design of the second and third switch architectures is based on interconnecting a large number of small capacity ATM switching modules. These switching modules have already been introduced in Chapter 2. The second architecture is a three-stage switch architecture where the first and second stages are completely interconnected. The third architecture uses a space switch to interconnect a number of switching modules. A distributed routing algorithm has been designed to speed up the switching process and increase the scalability range.

This chapter is organized as follow. Section 4.1 provides a survey of the previous work on high capacity switches. Section 4.2 introduces a shared-memory switch architecture. A new three-stage switch architecture is introduced in Section 4.3. Section 4.4 introduces a scalable space-switch based architecture with a distributed routing algorithm. The conclusions for this chapter are provided in Section 4.5.

4.1 Previous Work

Several proposals have been made for a terabit ATM switch architecture [71, 72, 73, 74, 45]. Almost all use electronic switching components exclusively. The possibility of purely photonic packet switching is, at present, severely limited by the difficulty of buffering cells and reading cell headers in the optical domain. Several switches that were previously reported will be reviewed in this section.

4.1.1 Shared Memory Design

In common memory architectures [80] shown in Figure 4.1, cells from each input port are placed in a shared buffer. Thus memory access speed must equal the switching speed. Common memory switches are based on a centralized memory scheduler which puts an upper limit on the scalability of the switch. The scheduler operating speed must keep up with the gross cell rate of incoming cells. The use of a central scheduler limits the failure resilience for this type of switch. If the scheduler fails, the whole switch fails.

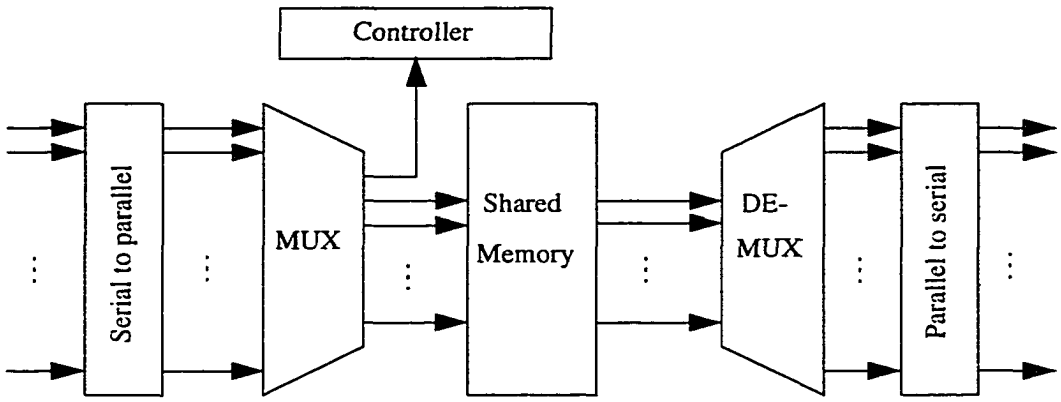

Figure 4.1 Shared memory architecture.

4.1.2 Three-Buffer-Stage ATM switch (Augmented Banyan)

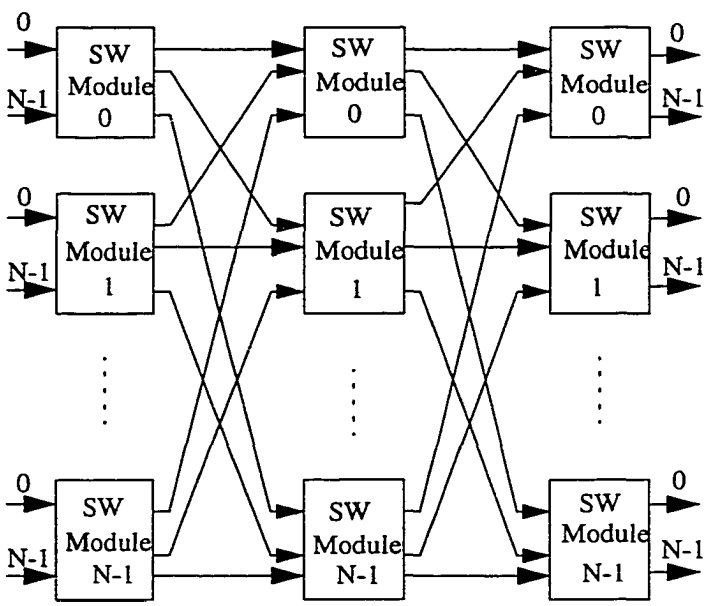

Figure 4.2 Three-buffer-stage growable ATM switch.

Figure 4.2 shows a three-buffer-stage configuration for an ATM switch [75, 76]. The main building block of this architecture is a switching module with size $N \times N$. In this architecture the capacity of each switching module is $N \times R$, where R is the link speed. $3N$ switch-

ing elements are used to build an ATM switch with $N \times N \times R$ capacity. The disadvantage in the growth characteristics of this approach is its limited scalability.

4.1.3 Space Switch Architecture

A space switch has been used to implement a scalable ATM switch architecture. The space switch connects input modules with output modules to provide a high capacity switch [75, 76]. The complexity of the central controller of the space switch is always a limit for the growability of the switch. A buffer-space-buffer switch which is a space switch based architecture is shown in Figure 4.3 [82]. This architecture implements input buffer queuing to store incoming cells. Information about the destination port is also stored with each cell. The transfer of cells between an input module and an output module is based on request and grant. With the receipt of a cell, an input module sends a signal to the central controller indicating the destined output port. The central control system receives many requests from different ports and, based on these requests, connections between input and output modules are established. An input switching module sends a cell to an output switching module which switches the cell to its destined port.

Arbitration logic is necessary to solve the problem of demand for connections. The arbitration logic needs information about the demands from all input ports. To limit the overhead of switching time, cells are transferred in groups. Connections are established to transfer a number of cells in a burst and after that the connections are reconfigured. The burst length is controlled by the allowed delay. Increasing the burst length

reduces switching overhead but increases the delay inside the switch which may not be acceptable for some services.

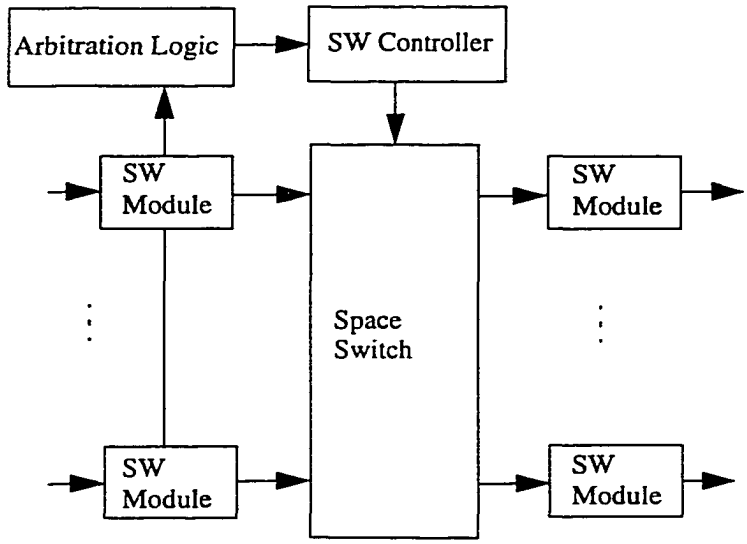

Figure 4.3 Buffer-space-buffer ATM switch architecture.

Since a buffer-space-buffer switch architecture relies on a central controller, the capacity of the switch is limited by the size and the complexity of the central controller. Utilization of the space switch depends on the reconfiguration speed of the controller.

4.1.4 Conveyor-Belt Switch Architecture

This switching system comprises N inlet modules, M outlet modules, P common memories, and two rotators. Each of the N inlet modules has M buffers. These buffers store cells according to the destination outlet modules. Each of the P common memories has M memory sections, each of which is able to hold a predetermined number of cells and is dedicated to an output module. The conveyor belt switch architecture is shown in Figure 4.4.

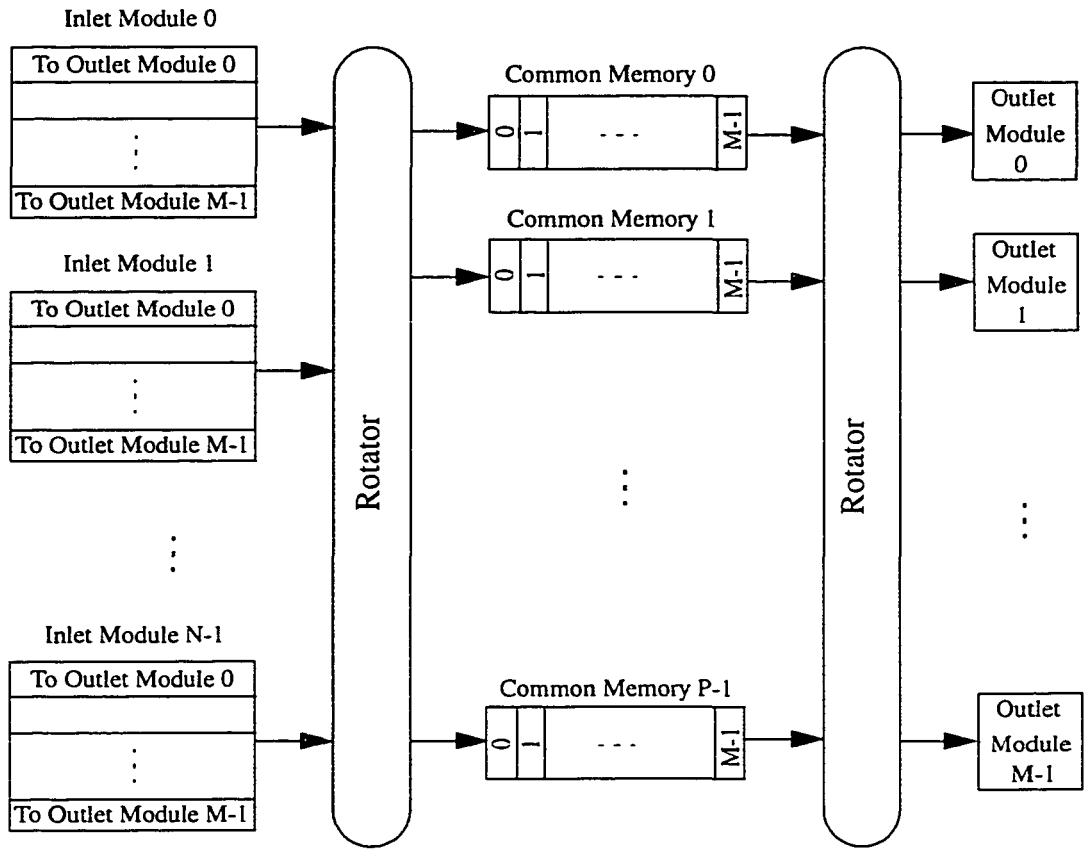

Figure 4.4 The conveyor-belt scalable ATM switch

The inlet rotator cyclically connects, in each access time, the N inlet modules and the P common memories so that respective cells are transferred from the N inlet modules and stored in respective sections in the common memory according to the destination outlet module of each cell. The outlet rotator cyclically connects, in each access time, the P common memories and M outlet modules so that respective outlet modules are connected to respective sections in the common memory for reading out cells contained therein. The transfer of data can be in bursts. A predetermined number of cells can be transferred among N inlet modules and the P common memories and between the P common memories and M outlet modules in each successive access time.

This architecture can not guarantee the QoS for all the services. The nature of the rotator introduces a priority level depending on the physical module of the arrived cells. For example if all the cells arriving from input modules 1 and 2 are destined to output module 3, module 1 traffic will be routed to output module 3 and module 2 traffic will be blocked. There is no maximum limit for cell delay in this architecture. Two rotators and a common memories are required to implement this architecture and this increases the implementation cost.

4.2 Proposed Wide-Word Memory Switch

A shared buffer architecture achieves better throughput performance than input or output queues [44]. The advantages of a shared buffer memory include better utilization of buffer space and less overhead owing to having less memory to manage [45]. The main disadvantage of shared queue switches is the high memory throughput requirements.

For an $N \times N$ switch, the shared buffer has to perform N write and N read operations per time-slot. Hence increasing the number of I/O links sharing a buffer places increased demand on shared memory speed, pin count, and power dissipation. If the access speed of the shared buffer is not fast enough to perform N read operations, output contention happens.

Although the shared buffer is reported to achieve the best performance, many designers consider it expensive because of the memory speed requirements. In this section, we introduce a folded shared memory ATM switch architecture. The main features of the proposed nonblocking ATM switch include

- High throughput shared buffer is implemented as a multi-bank memory.
- Shared bus is used between I/O port interfaces and shared memory.
- Parallel access to the memory banks makes the shared buffer appears as a wide high-speed memory.
- Required memory access speed is dramatically reduced.

4.2.1 Switch Architecture

Figure 4.5 shows a block diagram of the proposed folded architecture. The switch can be divided into three modules: Communication modules, buffering modules, and management modules.

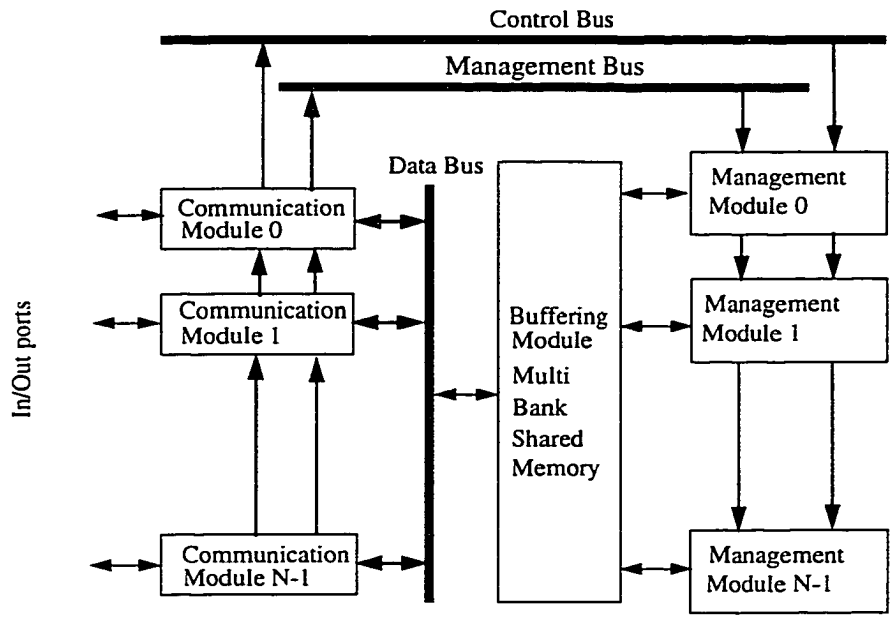

Figure 4.5 Block diagram of a wide-memory switch architecture.

There are three buses inside the switch: The data bus, control bus, and management bus. The data bus transfers a received cell from the communication module to the shared memory. The control bus carries information between the communication modules and the management modules. The management bus transfers resource management cells and OAM cells between communication and management modules.

Communication Module

Each port has a Communication Module (CM) and a management module. The CM, shown in Figure 4.6, stores incoming cells in communication buffers before sending them to the shared memory. Also, they are used to store outgoing cells before sending them to the physical interface. The CM receives cells from the physical interface. It separates the

cell header from the payload. The payload is stored in a buffer. The header processor checks the header and sends a message to the management module to update the virtual output queue of the destined VC, discussed in Section 4.2.2. Reading the header is done after storing the first 5 bytes of the cell and this speeds up the switching process. If the received cell is a resource management or an OAM cell, the whole cell is forwarded, using the management bus, to the management module associated with the outgoing port. Normal user data cells are sent to the shared memory via the shared bus.

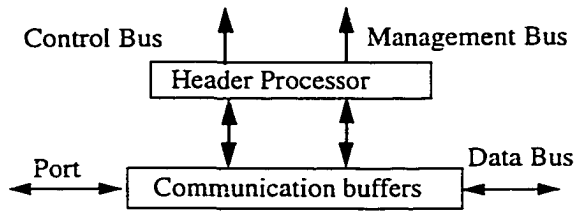

Figure 4.6 Communication module block diagram.

Buffering Module

The cell buffer is implemented as a common memory. The buffer memory is shared among all ports in the switch. This increases the utilization of the buffering space. In addition, the queuing structure used is independent of both the data path through the switch and the cell scheduling mechanism. The shared buffer is implemented as a multi-bank memory consisting of M memory banks.

Figure 4.7 shows the connection between the memory banks and the data bus. The data bus width equals $W \cdot M$, where W is the memory width per bank and M is number of memory banks. The first bank is connected to the first W lines of the data bus and the second bank is connected to the second W lines and so on. The data bus is shared between all the ports and it carries data to and from the different memory banks.

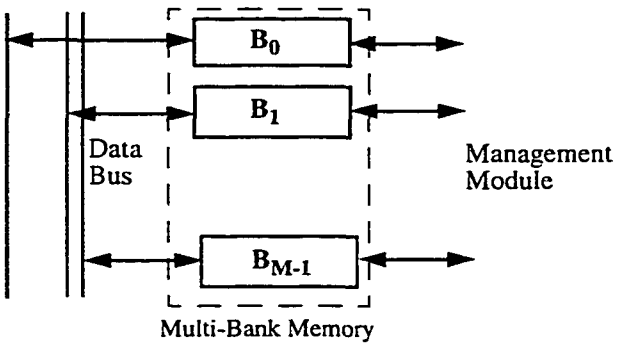

Figure 4.7 Buffering module block diagram.

N cells are written to the memory, one after the other, in one cell-time. Cell-time is defined as the time required to receive a complete cell from an input port. All memory banks are enabled at the same time to write a complete cell. The number of clock cycles required to write a complete cell (T_n) is:

$$T_n = \frac{53 \times 8}{B}$$

Where B is the bus width.

Reducing the width of the data bus can be achieved by sharing the same data lines among banks of memory. In current technology, bus speed is much faster than memory speed. A memory bank can latch data sent to it and free the data bus to transfer another data. Latching the data is much faster than writing them to the memory. Assuming that the bus speed is K times the memory speed. The shared bus width (B) is:

$$B = \frac{M \times W}{K}$$

The M parts of the incoming cell are stored at the same address in M banks. This simplifies the memory decoding circuit. Furthermore, only one management circuit is required for all memory banks. This eliminates the memory contention as reading or writing a cell is done by accessing all memory banks.

For an N-port switch, the required memory speed is

$$\text{Memory speed} = \frac{\text{Link Speed} \times 2N}{W \times M}$$

For a configuration with $M = 2N$, the memory speed required equals the link speed divided by the memory width. Further reduction of the memory speed can be achieved by increasing the number of memory banks.

Management Module

Each physical interface has a management module, shown in Figure 4.8. The management module has a virtual queue, and an output scheduler. The virtual queuing is implemented per VC which is more efficient and fairer than any other queuing strategy. The virtual queue stores the addresses of the cells destined to the port associated with that virtual queue. During each time-slot, the output scheduler selects a cell that will be sent.

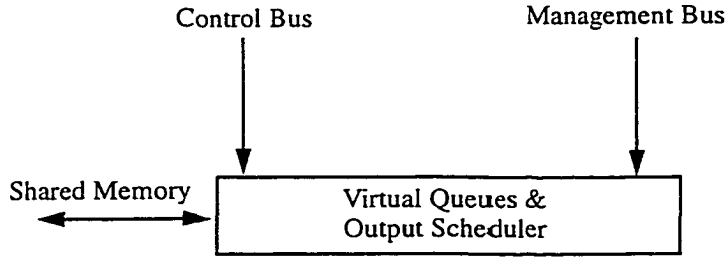

Figure 4.8 Management module block diagram.

4.2.2 Principle of Switching Operation

An ATM cell arriving at a communication module is temporarily stored in an input buffer. The routing information (i.e. header) of the cell is sent to header processor where VCI/VPI mapping is done.

The data bus carries cells from the communication modules to the shared memory Each communication module sends information about all of the incoming cells to the management module associated with outgoing port via the control bus. The manage-

ment module maintains per VC queuing for outgoing cells and every time-slot, it selects a cell to be sent out.

4.2.2.1 User cell flow

When a cell arrives at the communication module, it is stored in an input buffer. The header of the cell is checked and header translation is done by the header processor. Using the control bus, the header processor informs the management module of incoming cells. The management module updates the virtual queue of the destined VCs. After storing a user cell in the communication module, it is moved to the data bus to be stored in the shared memory. Always, the first part of any incoming cell is stored in the first bank and the second part in the second bank and so on.

The output operation is as follows: during each time-slot, each output scheduler decides which cell will be retrieved from the memory. There is no contention in accessing the shared memory as every cell is stored in all memory banks. To get a cell, all memory banks are accessed simultaneously. The simultaneous access makes the memory appear as a wide memory and this relaxes the speed requirements for each bank. The first $53*8/M$ bits are read from the first memory bank and the second $53*8/M$ bits are read from the second memory bank and so on. After reading the whole cell from the memory, it is directed to a buffer in the communication module which directs it to its destined port. Output schedulers in the management modules access the shared memory in order.

4.2.3 Wide-Bus Memory Vs. Interleaved memory

Interleaved memory is a multi-bank memory which provides a sequential access to its banks in order to satisfy the fast memory access requirements. The wide-bus memory organization proposed here has several advantages over interleaved memory organization [89]. For interleaved memory, the address for a memory bank is the same as that used by the previous bank in the previous time cycle. This requires address delay between the different banks which increases the complexity of the memory controller and prevents

sharing address lines to access different memory banks. In a wide-bus memory organization, the same address is simultaneously used for all the banks and this enables sharing the address lines for accessing different memory banks.

Both Interleaved and Wide-bus memories prevent memory contention by storing a cell in different memory banks. Storing a complete cell in one bank requires enough memory bandwidth to simultaneously serve all output ports, while storing a cell in M memory banks relaxes the memory speed requirements by factor M and prevents output contention. Retrieving a cell is done by accessing all the memory banks simultaneously so the retrieval time is much shorter than in the scenario when a complete cell is stored on one bank. If a cell is stored in M memory banks, the bandwidth available to read or write a cell equals the bandwidth of one memory bank multiplied by M while if a cell is stored in one bank the bandwidth to read or write a cell will be just the bandwidth for a single memory bank. If N cells are stored in M memory banks, the bandwidth available to simultaneously read or write the N cells is the bandwidth for a memory bank multiplied by M while if N cells are stored in one memory bank the bandwidth for reading the N cells simultaneously is the required bandwidth for one memory bank, thereby requiring a very fast memory.

4.2.4 Scalability Limitation

Increasing the capacity of the switch requires increasing the memory speed which can be achieved by increasing the number of memory banks. The highest switching capacity that can be achieved by the proposed switch architecture is when the complete ATM cell is written at once. Assume the external link speed is 622Mbps and the memory can run at 133Mhz and 3 cycles are required to perform a read or a write operation. The maximum number of ports (N) that can be supported equals the maximum number of cells that can be written to the memory in a cell-time. This number can be calculated as follow

$$N = \frac{53 \times 8/622}{3/133} = 30 \text{ ports}$$

The maximum capacity of the switch will be

$$\text{Maximum Capacity} = 622 \times N = 18.81 \text{ Gbps}$$

The number of memory banks (M) required will be

$$M = \frac{53 \times 8}{W}.$$

This architecture can not scale up to a capacity larger than 18.81Gbps without increasing the memory speed. This architecture can not support a terabit switching capacity. In the next section we will explore another high capacity switch architecture in an attempt to get a terabit switch architecture.

4.3 Three-Stage Switch Architecture

In practice, a variety of switch capacities is required. The majority of the demand is for small-capacity switches but requirements for various large switches are becoming very important. It is impractical to develop a switch optimized for each capacity required.

The proposed three-stage switch is built using small switching elements that can be built using inexpensive technology so that the cost of the switch is reasonable. A single switching element can be used as a stand-alone switch for low-capacity requirements whereas a high-capacity switching fabric can be easily built to satisfy high-bandwidth requirements. The proposed switch architecture can satisfy a wide range of applications. Expanding the switch capacity can be done by adding number of modules, making it suitable for a wide range of applications.

The switching module proposed in Chapter 2 is the main building block for the proposed three-stage switch. Some of the shift-register based switch features are directly inherited in this architecture. These features include:

- No HOL blocking. Virtual queues prevent the HOL problem and provide a flexible support for different QoS.
- Non-blocking switching architecture. The proposed architecture is non-blocking as

the switching module is non-blocking and the proposed connectivity scheme is non-blocking.

- Fast switching. The fast switching inside the switching modules with the fast inter-connecting networks leads to fast switching architecture.

The architecture of the switch is created by tying together a cluster of small-size switches, as shown in Figure 4.9. The three building blocks of the switch are a distributor, a selector, and an ATM switching module. The distributors forward incoming traffic to the header processing filters in all of the selectors. If the VPI and VCI values are within the acceptable range, the cell is stored. otherwise, the cell is dropped. Selector buffers are implemented using input queuing. Cells buffered in a selector are sent to the switching module connected to it. Each switching module switches incoming cells based on their header and an added field which identifies the incoming port. The proposed architecture has the following advantages:

- Fixed and small number of switching stages as only three switching stages are required.
- High reliability as cells belong to different input-output port pairs have different switching paths.
- The cost of adding new ports is almost constant.

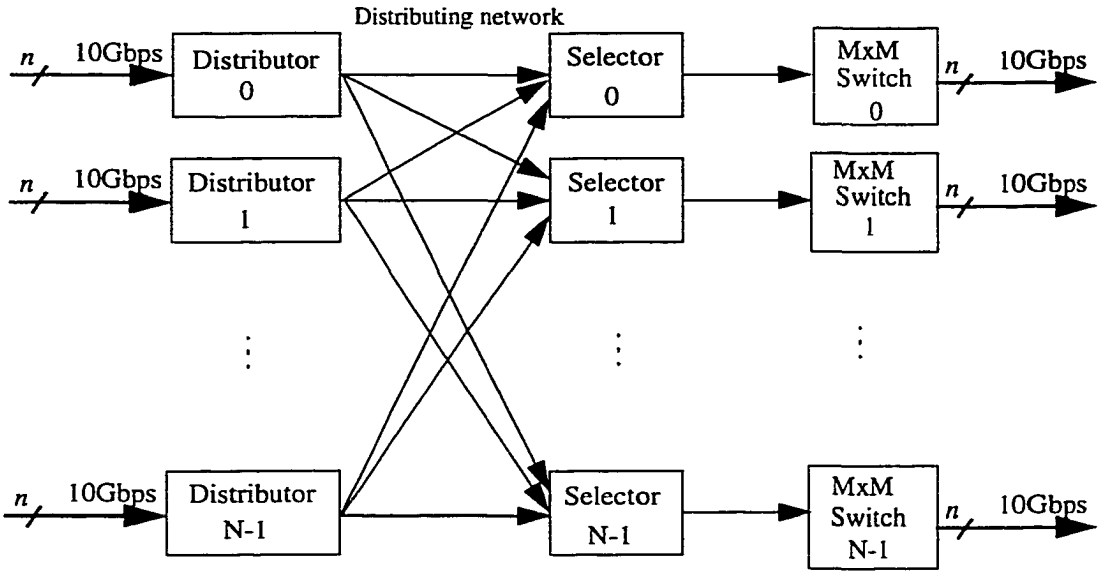

Figure 4.9 A scalable three-stage ATM switch architecture.

4.3.1 System Components

The three building components of the proposed switch are distributors, selectors, and switching elements. The next subsections describe each of them.

Distributor

A distributor sends copies of any incoming cell to N distributing buses, where N is the number of switching elements. Each of these buses is connected to a selector which is directly connected to a switching module, as shown in Figure 4.9. Information bytes which identify the incoming port are added to incoming cells. Each distributor forwards copies of incoming cells to all selectors without cell processing. The contents of the header and payload are not examined by distributors. A distributing bus speed equals the incoming link speed divided by the width of the bus. Assuming that the capacity of incoming link to a distributor is 10Gb/s and the bus width is 16 bits, the required bus speed is 625 Mhz.

Selector

A selector receives cells from all the distributors and filters out cells that are not destined to the associated switching module. The selector architecture, shown in Figure 4.10, has M (N/M)x1 multiplexors, M input queues, M header processors, and a round robin arbiter, where $M \leq N$. A multiplexor receives traffic from N/M distributors. Each multiplexor is connected to an input queue which is implemented as a multi-bank memory. This relaxes the memory speed requirements. The selector design is modular enough to facilitate smooth growability.

Each header processor gets a cell from an input queue and checks the header of this cell. If the cell is destined to the switching module connected to that selector, the cell is buffered, otherwise it is dropped. A round-robin arbiter is used to select a cell from the different input queues. The selected cell is sent to the switching module connected to the output of the selector.

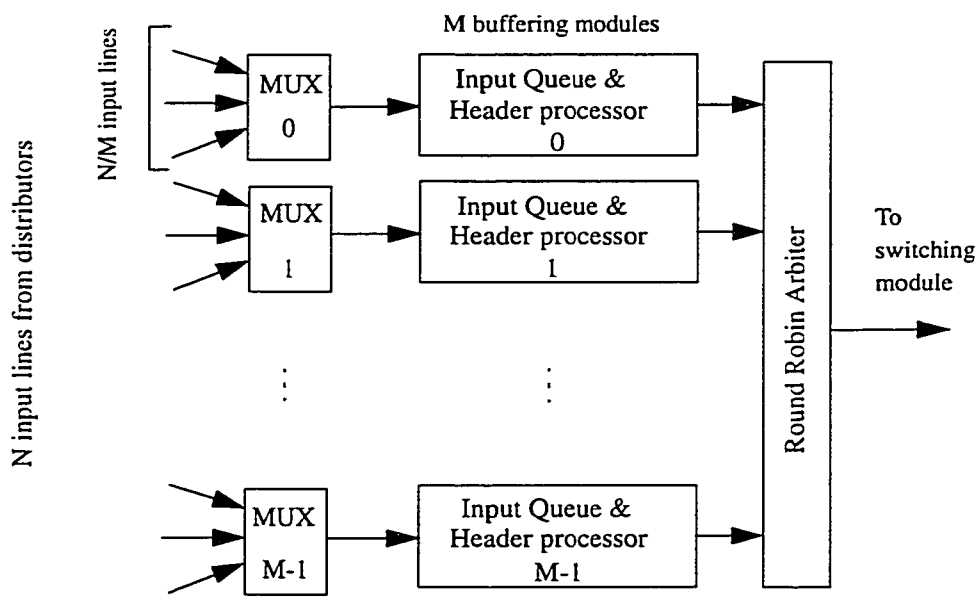

Figure 4.10 Block diagram of a selector.

Switching Element

The switching element is an ATM switch such as the one proposed in Chapter 2. Several ATM switch architectures have been proposed [80, 45]. Any of these can be used as the switching element. The switching element receives cells from the associated selector and switches them to their destination ports based on their headers and input port identifiers.

4.3.2 Switch Operation

The data stream coming to each distributor is directed to the N selectors. Every selector receives cells from N distributors. A selector output is connected to one switching element. If a selector receives a cell destined to its associated switching module, it buffers the cell in an input buffer, otherwise it drops the cell. The selector schedules cells out in a round-robin fashion. There is no head-of-the-line blocking problem as all the cells buffered in a selector are destined to only one switching module and the switching module is not blocking.

4.3.3 Reliability and Redundancy

Since large scale switches will be used in main switching stations, reliability of these switches is very important. We can use redundancy to provide fault protection [77] so that the proposed scalable architecture provides the high reliability required in switching systems. The scalable design of the switch offers multiple levels of redundancy which can be varied from component duplication to a complete fabric duplication. This duplication results in a cost penalty but it increases the reliability of the switch.

High reliability is an inherent feature of the proposed architecture. Cells destined to different switching modules use different independent elements. Failure of any element does not affect the remaining elements of the switch.

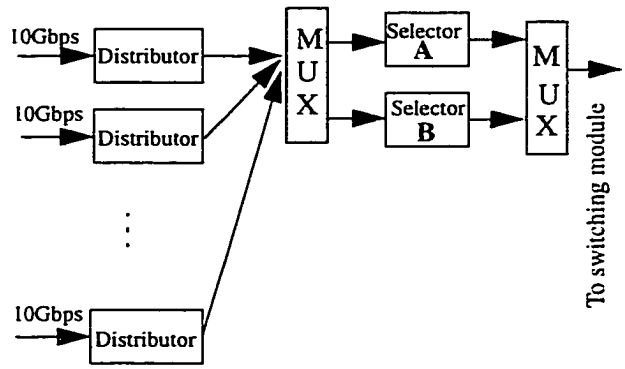

Figure 4.11 Example of selector redundancy.

Duplication does not have to be used for all elements in the cell path (distributor, selector and switching element). This adds more flexibility to the architecture. An example of duplicating the selector is shown in Figure 4.11. A multiplexer is added before the selector and a multiplexer is added after the selector. During normal operation all the cells go through selector *A*. After detecting its failure, all of the traffic is directed through selector *B*.

4.3.4 Switch Scalability

We should note that a cell crosses the same number of elements independent of the fabric's size. Thus, fabric growth does not have any major impact on cell-delay or on the throughput. The main complexity increase is the input buffering for the selectors. This is because a selector should be able to store cells coming from all the distributors.

To avoid any cell-loss, the capacity of the connection between a distributor and a selector is required to be the same as incoming link capacity (10G). The great number of high-capacity links between distributors and selectors limits the scalability of the switch. For this architecture to support a terabit capacity, 100 distributors and 100 selectors should be connected together. This mesh of connections between distributors and selectors is difficult to implement and maintain.

4.4 A Scalable Space-Switch Based Switch Architecture

As shown in section 4.2, the wide-memory switch architecture can not be scaled to the terabit range. The three-stage switch architecture suffers from the mesh connection between distributors and selectors which prevents it from being scaled to support traffic volume in the range of terabit per second. In this section, we introduce a new architecture which uses a space switch to connect switching modules. Figure 4.12 shows the proposed space-switch based architecture. N Switching Controller Modules (SCM) and N Switching Modules (SM) are linked together by a crosspoint switch. Each SCM module receives data from n input ports and sends multiplexed data to the crosspoint switch through a serial link. Each SM module receives data from the crosspoint switch and demultiplexes it to n output ports.

The SCM is a simplified switching module having a routing module and N output buffers. Each output buffer serves one SM which has a capacity equal to 10 Gbps and its architecture is as proposed in Chapter 2 or in [80, 82]. The routing module switches an incoming cell, based on its header, to one of the output buffers. Cell switching is done without any header translation.

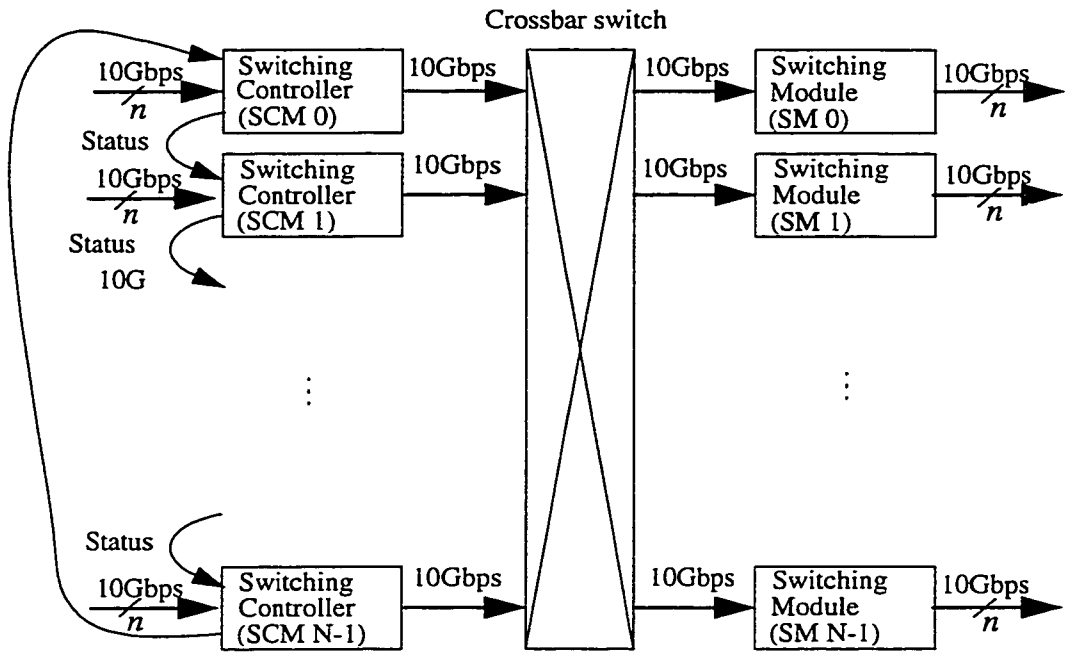

Figure 4.12 A scalable space-switch based ATM switch architecture.

An $N \times N$ crosspoint switch is used to connect the SCMs to the SMs. The crosspoint switch link capacity equals the aggregate capacity of one SCM which is 10 Gbps. Exchanging control information between the two switching stages (SMs and SCMs) is not required but exchanging control information between different SCMs is required. Each SCM is connected to one adjacent SCM. The first SCM is connected to the second and the second is connected to the third, etc. The last SCM is connected to the first one. These connections are used to transfer information between the SCMs. These local connections facilitate the implementation of a distributed routing scheduler instead of a centralized one.

4.4.1 Proposed Switch Characteristics

The proposed architecture has the following characteristics:

- Fixed and small number of switching stages. Cells pass through just two switching stages and this is independent of the switch capacity.

- Fair bandwidth allocation among sources. The bandwidth between any SCM and SM is fairly pre-allocated so that all input ports can send to all output ports. Any unused bandwidth is reassigned.
- Smooth scalability over a wide range. Increasing the number of SCM and SM increases the switching capacity.
- High delay/throughput performance
- No cell-loss. The crossbar switching between SCMs and SMs guarantees no cell loss. In addition, the SCM and SM can be designed with enough buffer space to provide zero cell loss.
- Support for multicast. Multicast is achieved by sending multicast cells to all destined SMs. Each SM sends the multicast cell to all destined ports.

4.4.2 Switching Operation

The crosspoint switch connects a SCM with a SM during a fixed interval equal to a time-slot. A time-slot is defined as the time required to transfer a cell from a SCM to the crosspoint. For a crosspoint switch with port speed of 10G/s, a time-slot is 40 ns. A switching cycle is defined as the time required to switch N cells. One switching cycle equals N time-slots. The time-slots are numbered from 1 to N in each switching cycle.

At the beginning of each switching cycle, the output bandwidth of each SCM is fairly divided among all the SMs. Every switching cycle, each SCM is allowed to send a cell to each SM. The pre-allocation of the bandwidth for the different SCMs is shown in Table 4.1. During each time-slot, every input link of the crosspoint switch is assigned to a SM. The SCM will use an input link of the crosspoint switch only if it has cells destined to the SM associated with this link at the current time-slot.

The first SCM is allowed to send a cell to the kth SM during the kth time-slot and a cell to the $(k+1)$th SM during $k+1$ time-slot. The second SCM is allowed to send a cell to $(k+1)$th SM during the k time-slot and a cell to the $(k+2)$th SM during $k+1$

time-slot and so on. According to that pre-allocation scheme, the bandwidth between any SCM and a SM is the same for any pair of SCM and SM. To achieve low cell-delay and for better switching utilization, re-assigning the unused bandwidth is also implemented.

Switching Controller Module	Pre-allocation connection sequence to Switching Modules
1	1, 2, 3, ..., N-1, N
2	2, 3, 4, ..., N-1, N, 1
3	3, 4, 5, ..., 1, 2
...	...
N	N, 1, 2, 3, ..., N-2, N-1

Table 4.1. Pre-allocation for switching bandwidth to different switching modules.

Based on the pre-allocated bandwidth, during every switching cycle each SCM schedules cells to be sent in the next switching cycle. As the incoming traffic may not be evenly distributed to the different outgoing ports, more than one cell may be destined to a particular SM and no cells to others. As a result, switching bandwidth is wasted.

To get better switching utilization, the pre-allocated bandwidth can be changed according to a distributed routing scheme. Control information about the time-slots is exchanged between the different SCMs. Each SCM sends its scheduled cell status to the next SCM. When a SCM receives this status, it locates the unused time-slots and identifies their associated SMs. The SCM may use any unused time-slot if it has a cell destined to the SM that was pre-assigned to that time-slot. Of course, the total bandwidth allocated to each output SM can not exceed the link capacity of the space-switch but different SCMs can use different portion of this bandwidth.

The bandwidth between a SCM and a SM is initially allocated to be $1/N$ of the space-switch link bandwidth but this bandwidth can be changed. A minimum of 1 cell every N cells from any source to any destination can be guaranteed. This feature enables support of CBR traffic.

The distributed routing scheme does not schedule more than one cell from an input module in a time-slot. In addition, no more than one cell can arrive at an output module in a time-slot. This prevents any possible contention and guarantees the maximum use of the bandwidth. The reallocation of the bandwidth is described in more detail below.

4.4.3 Routing Algorithm

The purpose of a routing algorithm is to direct input cells through the space switch to their desired SMs. The routing algorithm enables sharing switching bandwidth between SCMs. To attain this purpose, either a centralized or a distributed algorithm can be applied. A central controller could be used to collect traffic information from different SCMs to drive an optimal scheduler. However, the centralized gathering and distribution of information may become a bottleneck which may severely limit the utilization of the switching bandwidth. In addition the required processing speed is very high and needs excessively complex hardware. Therefore, we introduce a distributed scheduling algorithm that can achieve high performance and is simple enough to be implemented in hardware.

The routing algorithm always schedules cells one switching cycle ahead. At the beginning of a switching cycle the allocation of bandwidth between different SCMs and SMs is well defined. Thus the sequence of connecting SCMs and SMs is defined and this simplifies the control circuit for the crosspoint switch. In addition, it enables accelerating the controller switching speed and this increases the switching link utilizations.

During every switching cycle, each SCM can transmit at most N cells. Without sharing bandwidth, these N cells must be transmitted to different SMs in a time-slotted fashion. In order to share bandwidth between SCMs, bandwidth allocation is performed in N steps in the switching cycle preceding the present one. During every time-slot, an N-bit vector representing Scheduled cell Status of SCM_k (SS_K) is transferred between $SCM_{i \bmod N}$ and $SCM_{i+1 \bmod N}$. Figure 4.13 and Figure 4.14 show the transfer of scheduling status between different SCMs in two consecutive time-slots. Every bit of a SS_k represents a time-slot assigned to a SM. For SS_k, the first time-slot is assigned to SM_k and the second

time-slot is assigned to $SM_{k+1 \bmod N}$ and so on. Every SCM maintains N bits representing the cell buffer status (CBS) for all SMs.

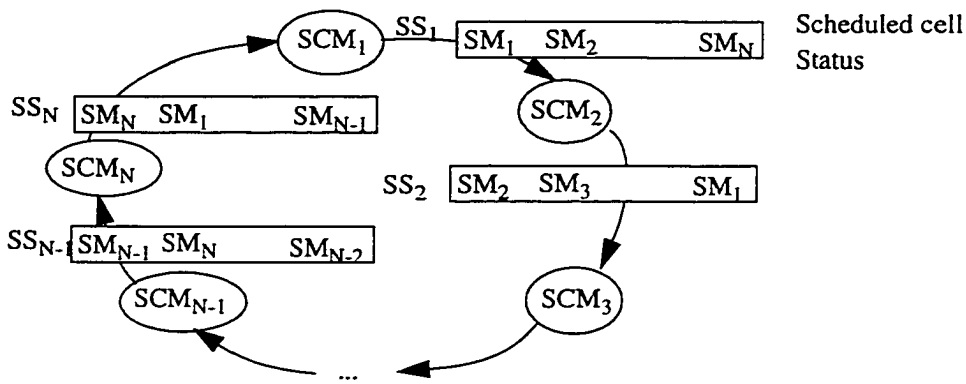

Fig.4.13.a

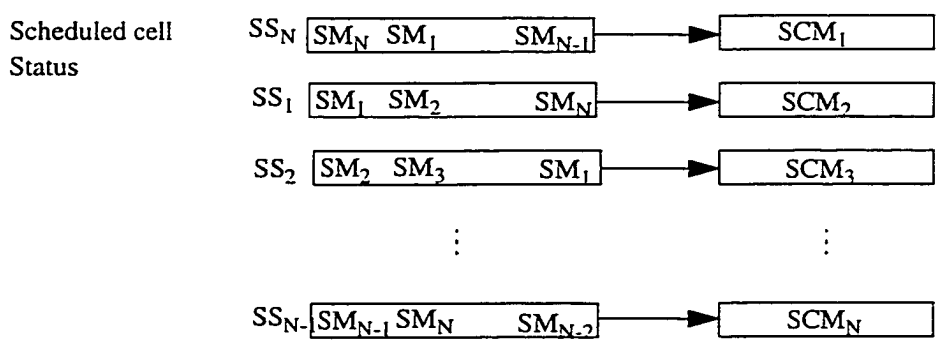

Fig.4.13.b

Figure 4.13 Transfer of status information at first time-slot.

Updating the SS vector is the basic operation in the distributed routing algorithm. At time-slot k, $SCM_{i \bmod N}$ sends SS_k to $SCM_{i+1 \bmod N}$. $SCM_{i+1 \bmod N}$ performs a matching operation between the free time-slots received in SS_k and the stored cell status in CBS and updates SS_k. The updated SS_k is sent to $SCM_{i+2 \bmod N}$. Each SCM updates the received SS based on the state of its CBS. In hardware terms, the SS and CBS are two N bit vectors. "1" and "0" in the status information denotes the occupancy and

availability of a time-slot, respectively, while "1" and "0" in the buffer status represent whether or not an output buffer has cells. Updating the N bits of SS is done in parallel. Status update hardware is shown in Figure 4.15. The hardware is simple and can be implemented to run at very high speed.

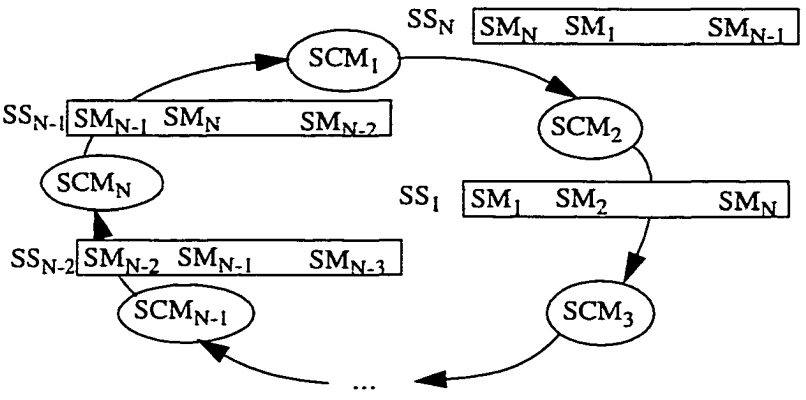

Fig.4.14.a

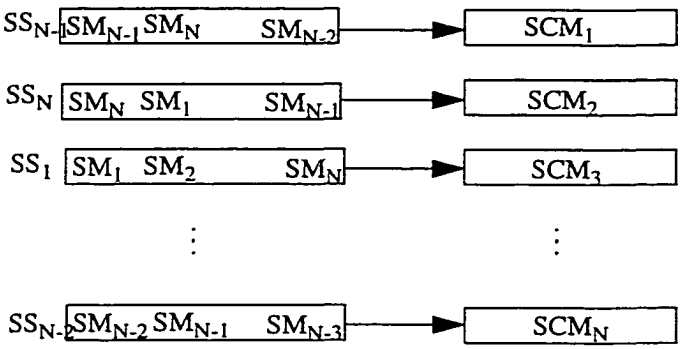

Fig.4.14.b

Figure 4.14 Transfer of status information at second time-slot.

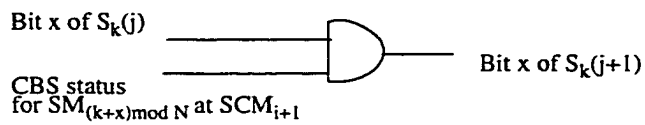

Figure 4.15 A simple hardware architecture for updating scheduling status.

4.4.4 Multicast

Different schemes have been proposed for supporting multicast [84, 85]. The proposed architecture supports multicast to any port. These ports may be in different switching modules or in one switching module. When a multicast cell is received, it is stored and scheduled to all destined switching modules. Only one copy is sent to a switching module. If a cell is destined to more than one port inside a switching module, the switching module sends it to the required ports. This can be easily done as the switching module described in chapter 2 provides multicast functionality.

4.4.5 Switching Performance

By definition, a switch is considered non-blocking if the blocking of an incoming request is determined solely by the destined outgoing link. The proposed switch is non-blocking in the sense that any unused bandwidth allocated to a SCM is available to other SCMs. This pre-allocated bandwidth is changed according to the incoming cell distribution.

Cell-level performance is normally expressed in terms of cell-loss probability and cell-delay variation. The round trip delay for ATM switches, traditionally specified for circuit switches, is required to be less than one millisecond. The round trip delay is the sum of the delay from inlet port X to outlet port Y plus the delay from inlet port Y to outlet port X. The two delays are not equal and each varies from one cell time to N cell times. For this architecture, the summation of these two delays depends only on the delay inside the SM and SCM. It does not depend on the relative position between SM and SCM since the space switch delay is always constant regardless of the position of

the SCM and SM. The round trip for the proposed switch equals one switching cycle plus the delay inside a SM and a SCM. In a 128 port switch, with a switching element speed of 10 Gb/s, the switching cycle is 53*8*128/10000=5.5 μsec.

The Cell Delay Variation (CDV) is very critical in performance evaluation. CDV determines the size of the smoothing buffer for CBR connections. The accepted CDV lies below 250 μsec. The proposed switch provides CDV of one switching cycle which is 5.5 μsec.

4.4.6 Simulations

Switches with different numbers of SCMs and SMs are simulated. The number of SMs are chosen to be equal to the number of SCMs and this number is called number of ports for the switch. The capacity of an SM or SCM is assumed to be 10 Gb/s. The links connected to an SCM or SM are represented as one link running at 10Gbps. The simulation assumptions are [81]

- The simulator is written in a C code and it simulates the behaviour of the proposed terabit switch and the routing algorithm.
- Simulation has been done for 16, 64, 128 port switches.
- The traffic arriving at an input port is a multiplex of traffic streams generated by several sources. The traffic generated by each source is simulated using the ON-OFF traffic model described in Chapter 3.
- The traffic load is assumed to be 80%.
- The number of cells processed in each simulation case is 10^9.

4.4.6.1 Simulation Analysis

Figure 4.16 shows the probability of the cell-delay for switches with 16, 64, and 128 ports. The x axis indicates the different cell delay that may occur while the y axis indicates the probability of a cell to exceed a certain delay. For 128 ports the switch capacity is 1.28 terabits/sec and the probability of a delay more than 300 cell-time ($\frac{300 \times 53 \times 8}{10G} \cong 12\,\mu\text{sec}$)

is less than 10^{-9}. As expected, the delay increases with the increase of the number of ports. For a 16 port switch the probability for a delay more than 100 cell-time is less than 10^{-9}.

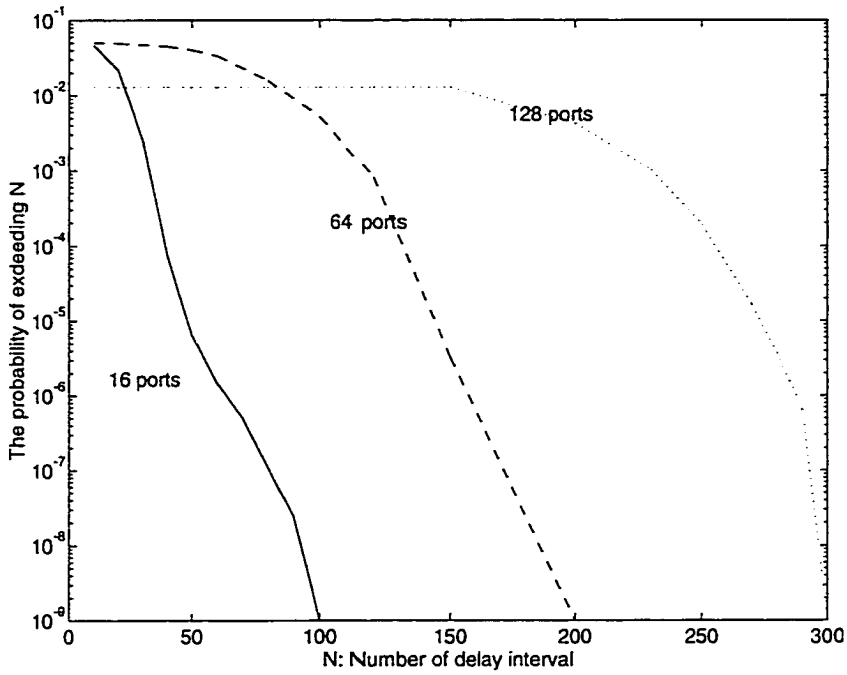

Figure 4.16 Cell-delay distribution.

Figure 4.17 shows the inlet buffer probability of overflowing the inlet buffers with different sizes for switches with 16, 64, and 128 ports. The x axis indicates the number of buffers for an input module while the y axis indicates the probability of exceeding a given buffer size. It shows that for a cell-loss of 10^{-7} a 12 cell buffer size is required. It is clear that the size of the required buffer is small and this simplifies the design of the memory manager. For probability of cell-loss less than 10^{-4}, the buffer size required for 16 port, 64 port, and 128 port switches is the same. The buffer size required is almost independent of the number of ports. This feature facilitates the scalability of the switch as the switching module which is used to build a switch with a small number of ports can be used to build a switch with a large number of ports.

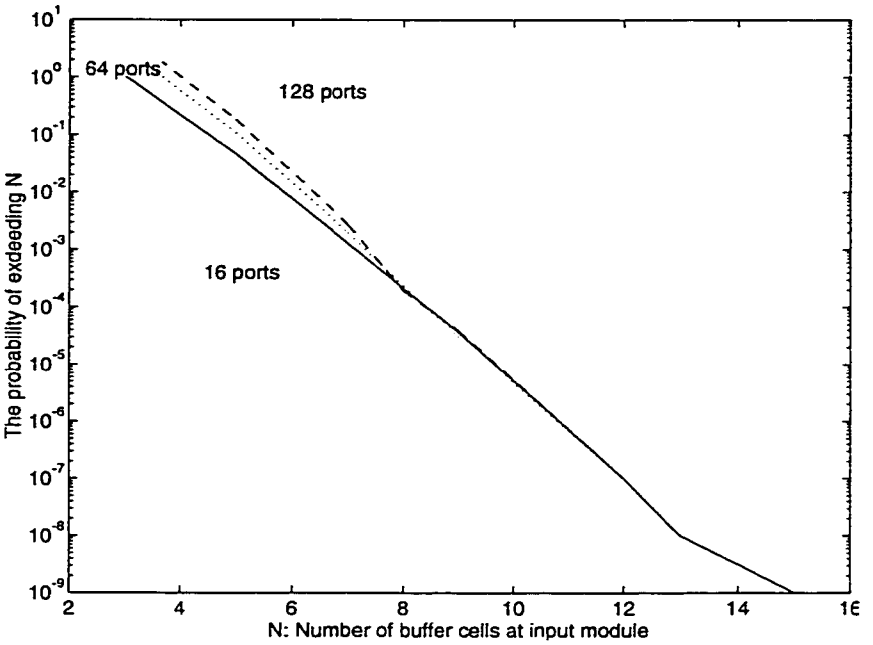

Figure 4.17 Inlet buffer distribution function.

4.4.7 Fault Tolerance and Recovery

A terabit switch should be resilient enough to tolerate different kinds of possible failures [77]. In the proposed switch architecture, failure may happen at a SCM, SM or crosspoint switch. Failure of a SCM or SM wastes only 1/N of the total switch capacity as only the bandwidth allocated to this SCM or SM will be wasted. High fault tolerance can be achieved by connecting the SCMs by a space switch and keeping a number of SCMs in standby mode to substitute any failed modules. At the beginning, the space switch will be configured to connect each SCM to the next one, as explained before in the architecture description. If an SCM fails the space switch will be configured to connect the two SCMs that were connecting to the failed SCM with one of the standby modules. A number of SMs will be kept in a standby mode to replace any failed SM. Figure 4.18 shows the architecture of the protected switch. The probability of failure for a crosspoint switch is small but if it is required to have a complete fault tolerance, 1:1 redundancy for the crosspoint switch can be implemented.

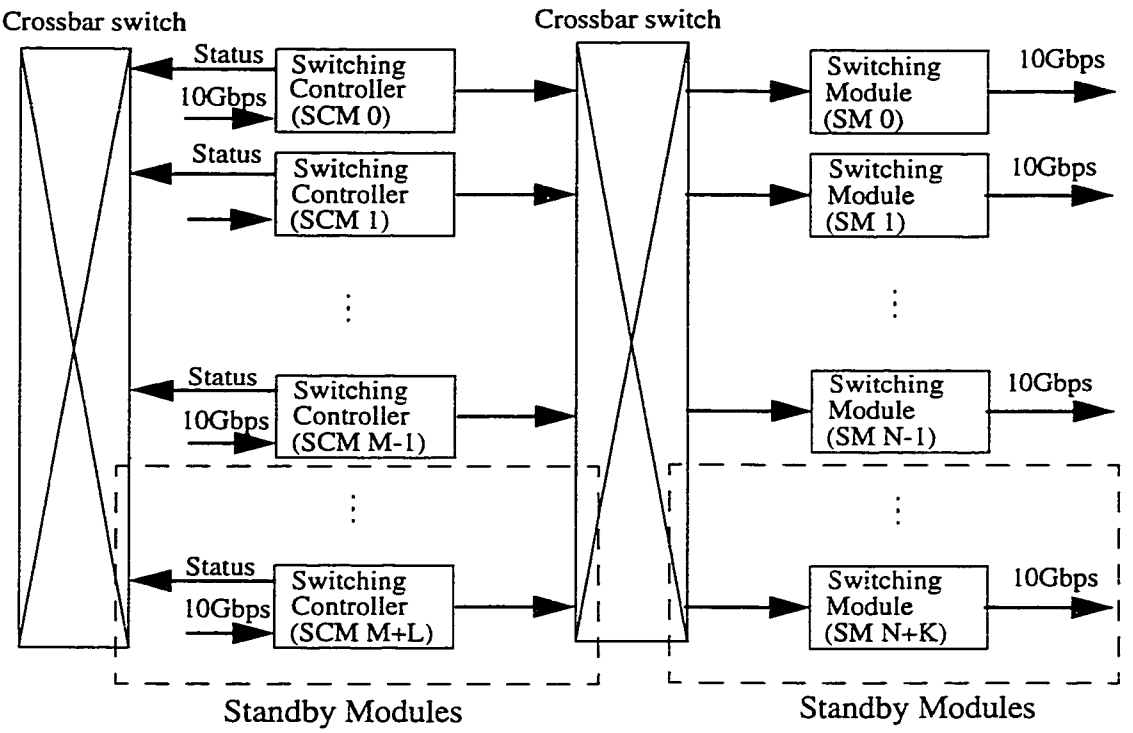

Figure 4.18 A fault-tolerant architecture for the scalable space-switch.

4.5 Conclusions

We have proposed three growable ATM switch architectures. The first architecture is a wide-memory switch architecture. This switch has all the benefits of the shared buffer without the high speed requirements. The scalability of this architecture is limited.

The second architecture is a three-stage switch architecture which has a constant cell-delay regardless of the switch size. The number of switching stages is limited to three and this limits the delay inside the switch to a small value and there is no cell-loss inside the switch. The modularity of the proposed architecture makes it possible to attain high system reliability. Due to independent cell paths, there is minimal degradation due to isolated faults. The great number of interconnections between the switch stages puts an upper limit on the switch scalability.

The third proposed switch architecture is based on using a space switch and output queues, which yields to the best delay/throughput performance. It can easily grow over a wide capacity range. An intelligent distributed routing algorithm is employed to enable the sharing of output module bandwidth between different input modules. The use of a distributed algorithm eliminates the need for a high-speed centralized controller and facilitates the growability of the switch capacity. Aside from its scalability, the proposed architecture has many attractive features, including high delay/throughput performance, support of broadcast and multicast functionalities and fair allocation of bandwidth between different input and output modules. Fault tolerance for this switch was also discussed.

In the next chapter we introduce a fast end-to-end error recovery protocol which is required to achieve a reliable communication in terabit networks.

Chapter 5

End-to-End Error Recovery Protocol

A fast end-to-end error recovery protocol is a must for high-speed ATM networks to support loss-sensitive services which cannot tolerate any cell loss. Selective Repeat Protocol (SRP) has the best performance when compared to other Automatic Repeat Request (ARQ) protocols. In ATM networks, a fast error recovery protocol is required for AAL5 applications. The error recovery protocol in ATM networks is based on frame retransmission rather than cell retransmission. If a cell in a frame is lost the whole frame will be retransmitted. The error detection capability of AAL5 determines whether a frame is successfully received. The challenge is to design an error recovery protocol which can work efficiently in high speed network environments. Also the protocol should be flexible enough to adapt itself to work in congested and uncongested states of the network.

In this chapter, we introduce a new error recovery protocol which minimizes the number of control cells required to achieve reliable communication. The proposed protocol is an improvement over SRP to handle flow control in high-speed networks. The proposed protocol minimizes the number of control cells without noticeably affecting the throughput. The proposed protocol adapts itself to react efficiently to changes in network status. It also works efficiently in both congested and uncongested states.

This chapter is organized as follows: The proposed end-to-end error recovery protocol is introduced in Section 5.1. The throughput analysis of the proposed protocol is provided in Section 5.2. Section 5.3 analyzes the number of control packets required in congested and

uncongested states. Packet delay is discussed in Section 5.4. The conclusions are provided in Section 5.5.

5.1 The Proposed Periodic SRP Protocol

The proposed Periodic Selective Repeat Protocol (PSRP) can be summarized as follows:

- No positive acknowledgments are sent. Negative acknowledgments are grouped together and sent as a STAT frame.
- During an uncongested state, STAT frames are sent at regular time intervals determined by allocated bandwidth. The STAT frame contains the sequence number of the last received frame and the sequence numbers of any lost frames. If there are no lost frames, the STAT frame will contain only the sequence number of the last received frame. Figure 5.1 illustrates the acknowledgment sequences.

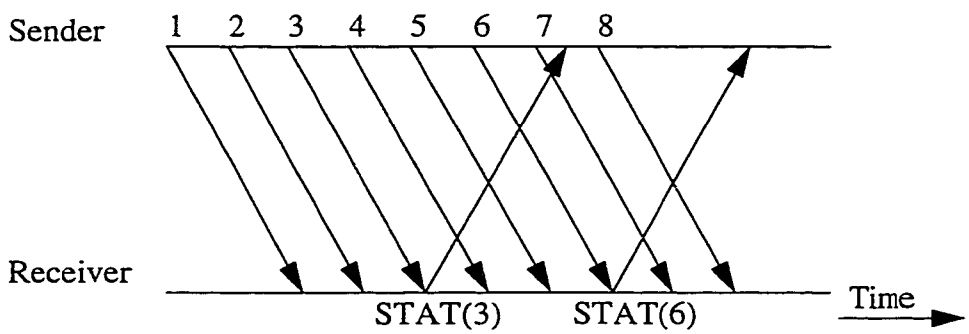

Figure 5.1 Example of events with PSRP. Down arrows represent transmitted frames and up arrows represent STAT frames.

- When the sender receives a STAT frame, any lost frames are retransmitted.
- STAT frame is always sent with the current status. If after sending the STAT frame the receiver does not receive any of the lost frames, the STAT frame will be sent with the updated status of all lost frames. The lost frames in the last STAT frame may have been in the previous STAT frame or were lost in the

period between two successive STAT frames. The error recovery procedure is shown in Figure 5.2.

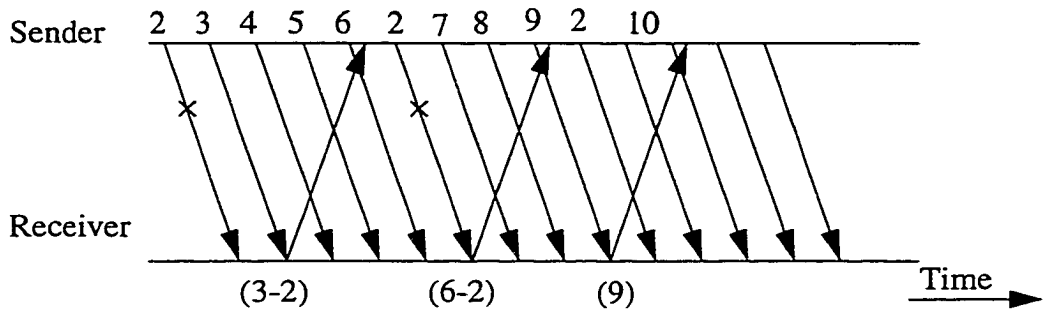

Figure 5.2 Error recovery in PSRP for uncongested state. Crossed arrows represent lost frames.

- If the sending window is full, the sender can ask the receiver to transmit a STAT frame. This helps flush out the sender's window. The request for STAT will be done by sending a POLL frame. The POLL frame is a normal frame with a bit indicating that a STAT frame is required. After the sender sends the POLL frame, it starts a time-out period. If a STAT frame is not received before the time-out, the POLL frame is considered to be lost and another POLL frame should be issued.

- When the sender receives the STAT frame, it frees the correctly received frame buffers and sends a copy of any lost frames. If, after receiving the STAT frame the window is still full, the sender will send another POLL frame. Otherwise, it will send new cells and delay the POLL frame until the window is full.

- If the network is about to become congested [7], the receiver will not issue STAT frames unless it receives a request from the sender. The sender will issue a POLL frame only when the window is full. After the sender issues the POLL frame request, it will begin a time-out for this request. The request will be sent again if the STAT frame is not received before the end of the time-out period.

The error recovery in the congested state is diagrammatically illustrated in Figure 5.3.

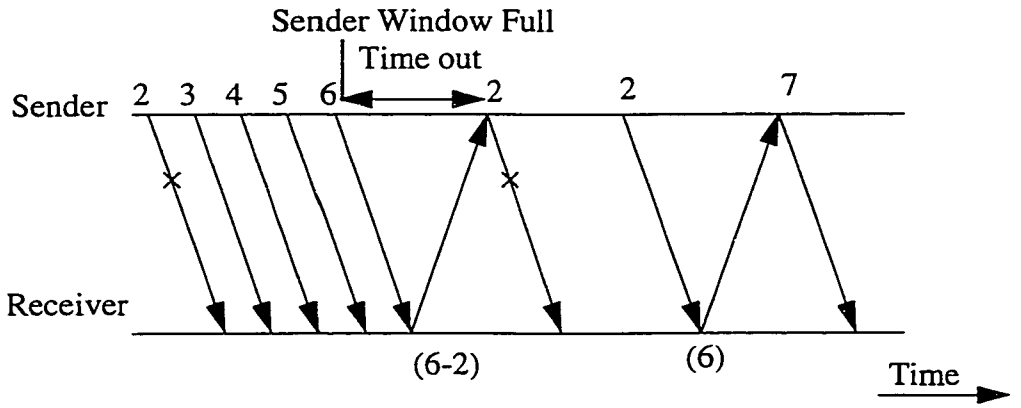

Figure 5.3 Error recovery in congested state. Crossed arrows represent lost frames.

- When the network becomes uncongested, STAT frames will be periodically issued again.

Unlike SRP, the proposed PSRP sends control cells periodically instead of sending a negative acknowledge (NACK) after receiving an erroneous frame and an acknowledge (ACK) after receiving a correct frame. In addition, PSRP is flexible enough to adapt itself according to the network status. It minimizes the number of control cells when the network is congested.

5.2 PSRP Analytic Model

The following notation will be used to define the protocol parameters [5]:

c = channel capacity (bps).

d = frame size (bits).

t_f = frame transmission delay (sec).

t_p = frame propagation delay (sec), i.e., one-way link delay.

p = probability of corrupted or lost frame.

W = window size (frames).

τ_1 = time interval between two STAT frames (sec).

τ_2 = time-out period (sec) ($\geq 2(t_p + t_I)$).

k = maximum number of STAT frames that can be sent before the window becomes full.

k can be given by

$$k = \left\lfloor \frac{Wt_I}{\tau_1} \right\rfloor$$

Frame transmission delay (t_I) can be expressed as

$$t_I = d/c.$$

5.3 Throughput Analysis

Throughput is defined as the maximum rate of successful bit transmissions when the channel is heavily loaded. This quantity is always normalized to the channel transmission rate. The throughput efficiency is derived based on the concept of average transmission time per frame. The average transmission time depends on the transmission, propagation and retransmission times. In this section, we will calculate the protocol throughput for congested and uncongested states.

5.3.1 Uncongested State

In the uncongested state PSRP average frame transmission time is given by

$$T_1 = t_I + t_1 + t_2 \tag{5.1}$$

where,

t_1 = retransmission time while window is not full; and

t_2 = retransmission time while window is full.

The time required to retransmit a frame correctly before the window becomes full (t_I) is given by

$$t_1 = t_I \times p(1-p) + 2t_I \times p^2(1-p) + 3t_I \times p^3(1-p) + \ldots + (k-1)t_I \times p^{k-1}(1-p)$$

$$= \sum_{i=1}^{k-1} ip^i(1-p)t_I$$

$$= t_I \frac{p[1-kp^{k-1}+(k-1)p^k]}{(1-p)} \tag{5.2}$$

The time required to retransmit a frame correctly after the window becomes full (t_2) is given by

$$t_2 = [\tau_2 \times p^k(1-p) + 2\tau_2 \times p^{k+1}(1-p) + 3\tau_2 \times p^{k+2}(1-p) + \ldots] + p^k(t_p + t_I)$$

$$= p^k(t_p + t_I) + \sum_{i=k}^{\infty} ([(i+1)-k]p^i)(1-p)\tau_2$$

$$= p^k(t_p + t_I) + \tau_2(1-p)p^{k-1} \sum_{j=1}^{\infty} jp^j$$

$$= p^k(t_p + t_I) + \tau_2(1-p)p^{k-1}\frac{p}{(1-p)^2}$$

$$= p^k(t_p + t_I) + \frac{p^k}{1-p}\tau_2 \tag{5.3}$$

The first term in equation (5.3) represents the transmission and propagation time required to retransmit a cell when the window is full in the uncongested state. The second term represents the waiting time-out period before retransmission when the window is full.

The throughput for uncongested transmission (ζ_1) is given by

$$\zeta_1 = \frac{t_I}{T_1}. \tag{5.4}$$

For an infinite window size, the throughput of PSRP will be equal to the original SRP. This is because as $W \rightarrow \infty$ so does k and equation (5.4) becomes

$$\hat{\zeta} = (1-p)$$

which is the same for the original SRP [95].

An approximate expression for the throughput can be obtained for networks with probability of cell-loss error less than 10^{-6}. For these networks, all the terms that contain p^n with $n \geq 2$ can be neglected. An approximate expression for t_I can be written as

$$t_1 \approx \frac{p}{1-p} t_I$$

The approximate value of t_2 will be almost zero so T_I can be written as

$$T_1 \approx \frac{t_I}{1-p}$$

The approximated expression for the throughput can be written as

$$\zeta_1 \approx 1-p$$

which is the same for the original SRP [95].

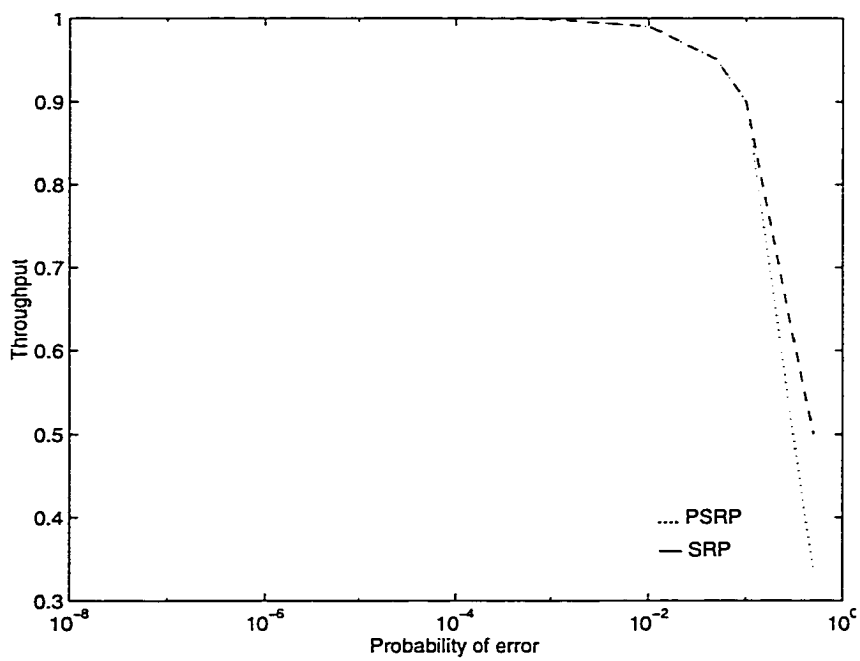

Figure 5.4 Throughput versus probability of error in normal state [K = 10, $\tau_2/t_I = 30$, $t_p/t_I = 1000$].

Figure 5.4 shows the throughput of the PSRP versus the probability of error in the uncongested state. It is clear from Figure 5.4 that the throughput is almost the same for SRP and PSRP especially when the probability of error is less than 10^{-2}. Therefore, for the uncongested state, time-out does not have a significant effect in the throughput. For ATM networks where the probability of error is less than 10^{-6}, PSRP provides the same throughput as SRP. For probability of error less than 10^{-1}, the PSRP throughput decreases.

PSRP sends an acknowledgment for a group of cells. The sender window is flushed only when all the sent cells have already been acknowledged. For networks with a high error rate, the probability that the sending window will get full while a cell is still retransmitted is high. When the sender window gets full, no new cells are sent and the throughput goes down.

5.3.2 Congested State

During congestion, the STAT frame will not be sent until the sender requests it. The sender will ask for this frame when the window is full. In this case, retransmission always happens when the window is full. Retransmission will be based on time-out on the sender's side.

The average time required to successfully transmit a frame during congestion can be given by:

$$T_2 = t_I + t_3$$

Where t_3 is the retransmission time when the window is full and is given by

$$\begin{aligned} t_3 &= p(t_p + t_I) + [\tau_2 p(1-p) + 2\tau_2 \times p^2(1-p) + \ldots] \\ &= p(t_p + t_I) + \tau_2 \sum_{i=1}^{\infty} i p^i (1-p) \\ &= p(t_p + t_I) + \frac{p\tau_2}{(1-p)} \end{aligned} \tag{5.5}$$

The first term in equation (5.5) represents the transmission and propagation time required to retransmit a cell when the window is full in the congested state. The second term represents the waiting time-out period before retransmission when the window is full.

The average time required to transmit a frame during congestion is given by

$$T_2 = t_I + p(t_p + t_I) + \frac{p\tau_2}{(1-p)} \tag{5.6}$$

The throughput (ζ_2) is given by

$$\zeta_2 = \frac{t_I}{T_2}.$$

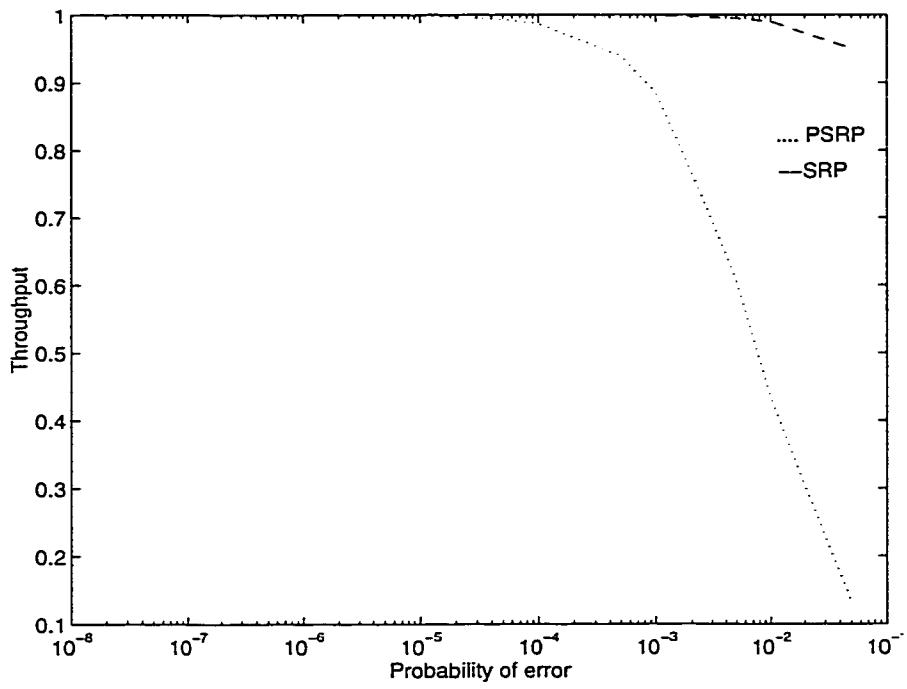

Figure 5.5 Throughput versus probability of error in congested state [K = 10, $\tau_2/t_I = 30$, $t_p/t_I = 100$].

Figure 5.5 shows the throughput of the PSRP versus the probability of error in the congested state. High speed networks typically have a probability of error less than 10^{-4}. For this probability of error the throughput of PSRP and SRP is almost the same. For probabil-

ity of error higher than 10^{-4}, the throughput in SRP is better than in PSRP. This happens because in congested networks, PSRP reduces the number of control cells to reduce the network load and allow fast recovery from congestion. For networks with low error-rate ($P<10^{-4}$), for which this protocol has been developed, the probability of loosing a group acknowledgments is low. Using the group acknowledgments reduces the number of control cells while keeping the throughput as high as for SRP.

For $p < 10^{-6}$ an approximate expression for t_3 can be written as

$$t_3 \approx p(t_i + t_p) + \frac{p\tau_2}{1-p}$$

and T_2 can be approximated as

$$\begin{aligned} T_2 &\approx t_I(1-p) + pt_p + \frac{p\tau_2}{1-p} \\ &\approx \frac{t_I(1-2p) + pt_p + p\tau_2}{1-p} \end{aligned}$$

An approximated expression for the throughput can then be written as

$$\zeta_2 = \frac{1-p}{(1-2p) + p(t_p + \tau_2)/t_I}$$

5.4 Control Cell Analysis

Control cells are necessary overhead to achieve reliable communication. The number of control cells is reduced dramatically in PSRP. In the original SRP, an acknowledgment is sent for any received frame and a negative acknowledgment is sent for any lost frame. Lost frames are retransmitted until they are correctly received. For each retransmission an acknowledgment or negative acknowledgment is sent. The number of control cells per W transmitted frames is given by

$$\begin{aligned} N &= W + W[p(1-p) + 2p^2(1-p) + \ldots] \\ &= \frac{W}{(1-p)} \end{aligned}$$

The control cell overhead is defined according to the following formula

$$\frac{N}{W} = \frac{1}{1-p} \tag{5.7}$$

5.4.1 Uncongested State

In PSRP, the number of the control cells changes according to the network status. For the uncongested state, the number of control cells can be given by

$$N_1 = n_1 + n_2 \tag{5.8}$$

where,

n_1= number of STAT frames required to send W frames; and
n_2= number of STAT frames required to recover the lost frames when the window is full.
n_1 is given by

$$n_1 = W\frac{t_I}{\tau_1} \tag{5.9}$$

In order to estimate n_2, we assume a worst-case scenario in which the sender's window is full and X_1 frames have not been acknowledged, where X_1 is given by

$$X_1 = \frac{\tau_1}{t_I}$$

Under these conditions, the sender will stop transmitting any new frames and the lost frames will be retransmitted. At the end of every retransmission of lost frames, the sender will send a POLL frame and will wait a time-out period to receive a STAT frame. If STAT frame is not received, the sender will resend the POLL frame. Based on normal operation, the number of frames to be retransmitted will progressively decrease as more frames are received without error. This can be diagrammatically illustrated as shown in Figure 5.6.

The number of states X_1 equals the number of frames that are lost. The starting state depends on the number of lost frames in the first transmission just after the sender's window became full. The sender can be in one of several states. State *i* indicates that *i* frames are still in error and have to be retransmitted. State transitions can only occur from a

higher to a lower state. The transfer from one state to another depends on the number of successfully received frames sent in that state. The time-out mechanism will be applied to the last frame sent in any state therefore we will be sure that at least one frame will be received correctly before each state transition.

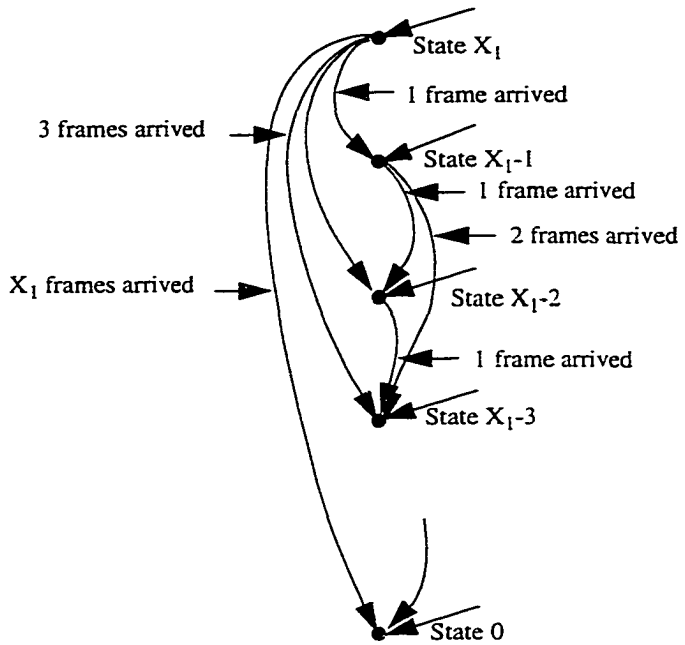

Figure 5.6 State transitions when window is full.

For instance, in state X_1, X_1 frames are sent and the last frame is a POLL frame. The time-out mechanism will be initiated for the last frame. The new state will be identified according to the correctly received frames. If only one frame is received, we will go to state X_1-1. If x frames are received, we will go to state X_1-x. We will move from one state to another until all the lost frames are received.

The probability of getting into state X_1 is given by

$$P_{X_1} = p^{X_1}$$

The probability of getting into state (X_1-1) is given by

$$P_{X_1-1} = \binom{X_1}{X_1-1} p^{(X_1-1)}(1-p) + \left[\binom{X_1-1}{X_1-1} p^{(X_1-1)}\right][P_{X_1}]$$

The probability of getting into state (X_1-2) can be given by

$$P_{X_1-2} = \binom{X_1}{X_1-2} p^{(X_1-2)}(1-p)^2 + \left[\binom{X_1-1}{X_1-2} p^{(X_1-2)}(1-p)\right][P_{X_1}] + \left[\binom{X_1-2}{X_1-2} p^{(X_1-2)}(1-p)\right][P_{X_1-1}]$$

The general form for the probability of getting to state (X_1-i) is written as

$$P_{X_1-i} = \binom{X_1}{X_1-i} p^{(X_1-i)}(1-p)^i + \sum_{j=0}^{i-1} \binom{X_1-j-1}{X_1-i} p^{(X_1-i)}(1-p)^{(i-j-1)} P_{(X_1-j)}$$

As there is only one control cell sent per state, the number of control cells when the window is full can be given by

$$n_2 = p^k \sum_{i=0}^{X_1} P_i \tag{5.10}$$

The overhead of the control cells can be given by

$$\frac{N_1}{W} = \frac{t_I}{\tau_1} + \frac{p^k \sum_{i=0}^{\cdot} P_i}{W}$$

$$N_1 = \frac{Wt_I}{\tau_1} + p^k \sum_{i=0}^{X_1} P_i \tag{5.11}$$

In equation (5.11) Wt_I/τ_I represents the minimum number of control cells required to deliver W frames. However, with SRP, W is the minimum number of required control cells to deliver W frames. Instead of sending an acknowledgment for every received frame, PSRP groups the acknowledgments and sends them every t_I seconds, thereby dramatically reducing the number of control cells. Using Equation (5.11), Figure 5.7 has been gener-

ated to illustrate the reduction in the number of control cells for PSRP in the uncongested state. The number of control cells can be easily approximated as

$$\frac{N_1}{W} \approx \frac{t_I}{\tau_1}$$

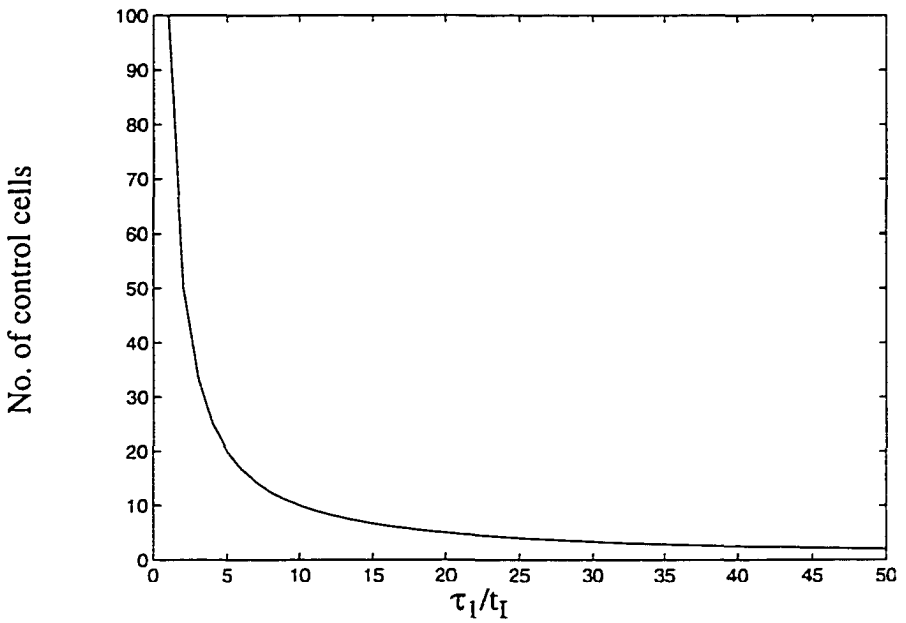

Figure 5.7 Number of control cells in the uncongested state [W=X_1=100, P=10^{-4}].

5.4.2 Congested State

For the congested state, there are no periodic STAT frames. The number of frames that would be sent before the window becomes full can be expressed as $X_2 = W$.

With the same analysis as above we can find out that the number of control cells required to send N_2 frames can be given by

$$P_{X_2-i} = \binom{X_2}{X_2-1} p^{(X_2-i)}(1-p)^i + \sum_{j=0}^{i-1} \binom{X_2-j-1}{X_2-i} p^{(X_2-i)}(1-p)^{(i-j-1)} P_{(X_2-j)}$$

As there is only one control cell sent per state, the number of control cells when window is full (n_3) can be given by

$$n_3 = \sum_{i=0}^{X_2} P_i$$

The overhead will be given by

$$\frac{N_2}{W} = \frac{\sum_{i=0}^{X_2} P_i}{W} \tag{5.12}$$

Using Equation (5.12), Figure 5.8 has been generated to show the number of control cells in the congested state. This diagram demonstrates that the number of control cells in the congested state is dramatically fewer than that required in the uncongested state. This reduces the load in the congested network and allows it to recover from congestion. For SRP, the number of control cells in the congested is always greater than *W*. From Figure 5.7 and Figure 5.8 it is clear that PSRP uses less control cells than SRP in uncongested and congested states.

For probability of error less than 10^{-6}, the starting state can be considered to be state 0 or state 1.

The approximate probability of getting into state 1 is given by

$$P_1 \approx \binom{X_1}{1} p(1-p)^{X_1 - 1}$$

The approximate probability of getting into state 0 is given by

$$P_0 \approx P_1 + \binom{X_1}{0} (1-p)^{X_1}$$

Therefore, the approximated expression for the overhead is written as

$$\frac{N_2}{W} \approx \frac{1 + 2pX_1}{W}$$

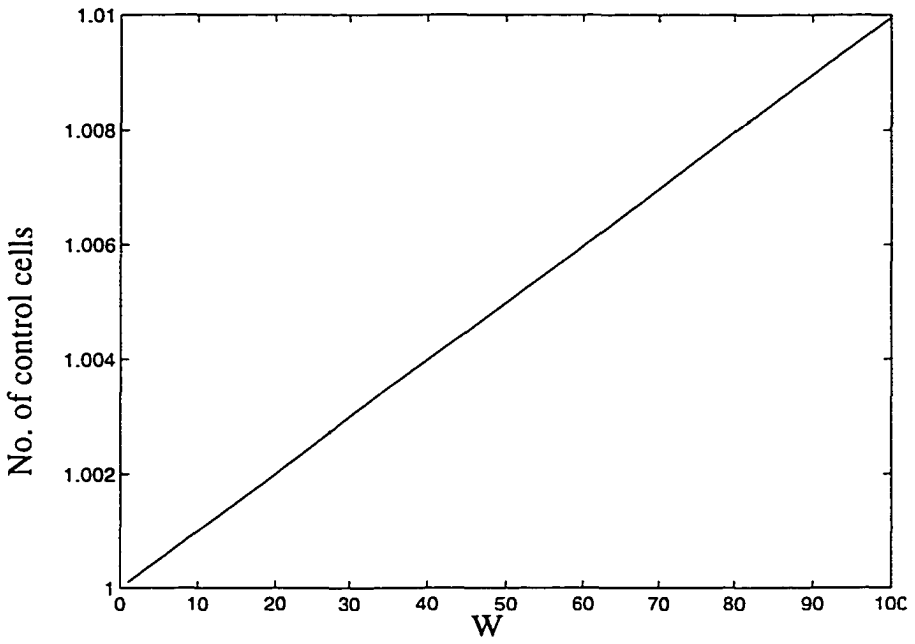

Figure 5.8 Number of control cells in the congested state [W=X_2=100, P=10^{-4}].

5.5 Frame Delay Analysis

In PSRP, frames are buffered at the receiver if any of the previous frames are lost. The required buffering and resequencing cause frame delay. The analysis of this delay can be done as follows.

The total time delay of a frame (T_t) can be written as [6]:

$$T_t = t_d + t_w$$

where,

t_d = the time required to transmit a frame; and

t_w = the average resquence time for all previous frames.

The average time required to transmit a frame can be written as

$$t_d = t_p + \frac{t_I}{(1-p)} + NT$$

Where,

N = the average number of retransmission trials; and

T = the average time between retransmission.

The average number of retransmissions (N) can be written as

$$N = \frac{p}{(1-p)}$$

The average time between retransmissions (T) can be given by

$$T = \frac{\tau_1 \sum_{i=0}^{k-1} p^i(1-p) + \tau_2 \sum_{i=k}^{\infty} p^i(1-p)}{\sum_{i=0}^{\infty} p^i(1-p)}$$

$$= \frac{\tau_1 \frac{(1-p^k)}{(1-p)} + \tau_2 \frac{p^k}{1-p}}{\frac{1}{(1-p)}} = \tau_1(1-p^k) + \tau_2 p^k$$

The average resquence time (t_w) can be defined as the time required to retransmit all of the lost frames before frame number X.

In order to calculate the resequence time (t_w), we need to calculate the time required to retransmit any of the lost frames before the current one.

The average number of frames that should be received before the current frame can be given by

$$X = \frac{\sum_{i=0} i}{W}$$

The time required to retransmit frame number i from its transmission until its successful arrival can be given by

$$t_r = \sum_{l=0}^{\infty} Tl(1-p)p^l$$

If we consider that any of the lost frames may be retransmitted in the time between its first transmission and the transmission of frame number X, the time required for successful retransmission of frame number i ($i<X$) can be written as:

$$t_r = \sum_{l=0}^{\infty} Tl(1-p)p^{\left\lfloor \frac{t_d + t_I \cdot (X-i)}{T} + l \right\rfloor}$$

Upon receipt of frame number X, the probability that frame number i is still lost is given by

$$P_i = \left[1 - \sum_{j=0}^{\left\lfloor \frac{t_d + t_I \cdot (X-i)}{T} \right\rfloor} p^j(1-p) \right]$$

The average resquence time (l_w) can be written as

$$t_w = P_1\left(T\sum_{l=0}^{\infty} l(1-p)p^{\left\lfloor \frac{t_d + t_I \cdot (X-1)}{T} \right\rfloor + l} \right) + P_2\left(T\sum_{l=0}^{\infty} l(1-p)p^{\left\lfloor \frac{t_d + t_I \cdot (X-2)}{T} + l \right\rfloor} \right)$$

$$+ \ldots + P_X\left(T\sum_{l=0}^{\infty} l(1-p)p^{\left\lfloor \frac{t_d + t_I \cdot 1}{T} + l \right\rfloor} \right)$$

The total delay time (T_t) for frame number X is given by

$$T_t = t_d + t_w$$

$$T_t = t_p + \frac{t_I}{1-p} + NT + \sum_{i=0}^{X}\left(\left[1 - \sum_{j=0}^{\left\lfloor \frac{t_d + t_I \cdot (X-i)}{T} \right\rfloor} p^j(1-p) \right] \sum_{l=0}^{\infty} Tl(1-p)\left\lfloor p^{\frac{t_d + t_I \cdot (X-i)}{T} + l} \right\rfloor \right)$$

For probability of error less than 10^{-6} we can neglect all the terms that contain p^n where $n \geq 2$. The approximate expression for T_t can be written as

$$T_t \approx t_p + \frac{t_I}{1-p} + \frac{p\tau_1}{1-p}$$

An approximate expression for normalized frame delay can then be written as.

$$\frac{T_t}{t_I} \approx \frac{t_p}{t_I} + \frac{1}{1-p} + \frac{p}{1-p}\frac{\tau_1}{t_I} \tag{5.13}$$

Figure 5.9 has been generated to show frame delay as a function of probability of error. For a probability of error less than 10^{-2}, the frame delay is kept to a very small value. For a probability of error more than 10^{-1} frame delay increases since there is a high probability that a frame may be retransmitted more than once.

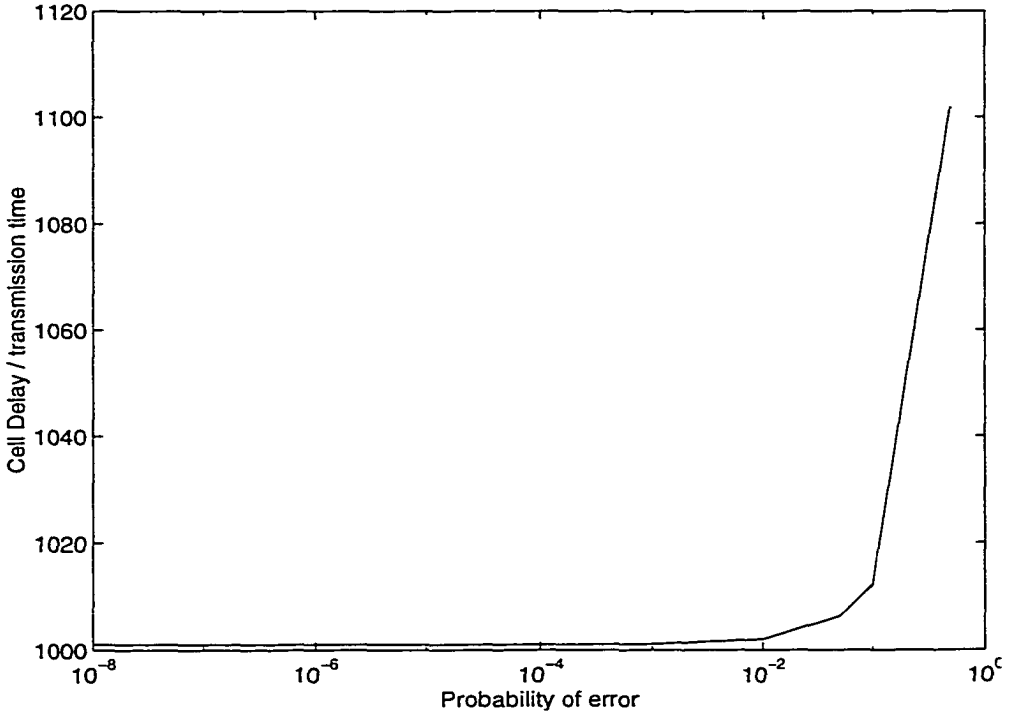

Figure 5.9 Frame delay versus probability of error in PSRP [$\tau_1/t_I = 100$, $t_p/t_I = 1000$].

5.6 Conclusions

In this chapter, we presented a fast end-to-end error recovery protocol. The proposed PSRP *minimizes* the number of control cells in both normal and congested states. We obtained exact and approximate expressions for throughput, average number of control cells and average frame delay in both normal and congested states. The computed results illustrate the high throughput and limited frame delay of the proposed PSRP scheme. They also illustrate the ability of PSRP to work efficiently in both congested and uncongested states. The PSRP is simple enough to work efficiently in ATM networks. The simulations showed that for networks with low probability of error the PSRP achieves the same performance as SRP with fewer control cells.

In the next chapter, the thesis conclusions are introduced and directions for further research are discussed.

Chapter 6

Conclusions and Directions for Future Research

6.1 Thesis Conclusions

The goal of this thesis was to exploit two of the key issues required to develop high-capacity high-speed ATM networks. This thesis introduces:

- A high-capacity ATM switch architecture. This architecture provides advanced features in addition to its high-speed performance.
- A fast error recovery protocol. This protocol enables reliable communication over ATM networks.

Memory access speed is the main factor in designing ATM switches because numerous ATM switching functions are memory-intensive. We have defined, explored and provided an architectural design for an input buffer based ATM switch architecture. The proposed architecture implements cell buffers as groups of parallel shift registers. This implementation uses shift registers as a fast memory.

Cell blocking for the proposed architecture is addressed. Cell blocking occurs whenever cells at multiple inputs are destined for the same output in the same cycle. To solve this problem a conflict resolution mechanism has been implemented. The proposed switch architecture provides an efficient conflict resolution protocol which is able to solve the HOL problem without consuming switch resources. The conflict resolution is based on using a virtual output queue. Cell scheduling is implemented per

output port. This scheme selects a virtual queue and gets that HOL cell from that queue. Cell scheduling per output port avoids cell blocking and flexibly supports priority control scheme.

A priority scheduling scheme has been introduced. The proposed scheme distributes cell delay over different QoS based on their priorities. The proposed scheme works efficiently when real-time and non-real-service are simultaneously supported.

This thesis also proposed a high-capacity switch architecture. The proposed switch architecture is based on connecting a number of switching elements that have been proposed in Chapter 2 together to achieve the high-capacity requirements. The proposed switch architecture uses a space switch to connect number of switching elements. The proposed switch has the following features:

- An efficient distributed routing algorithm has been designed to achieve highcapacity with simple hardware architecture.
- Cell delay inside the switch depends on the switch size. Simulations show that for up to 128 switching elements, the delay inside the switch is remarkably low.
- The proposed architecture has a good fault tolerance. Failure of a switching element does not break down the whole switch, it just reduces the capacity of the switch.

In the efforts to develop a terabit switch architecture, two other ATM switches have been introduced. The two architectures are scalable but cannot be economically scaled to the terabit range. The first architecture implements cell buffer as a multi-bank shared memory. Use of a shared bus and multi-bank memory provides high memory throughput without requiring fast memory. The second architecture comprises interconnected modules in three stages. Cell delay inside the switch is fixed and does not depend on the switch size. The scalability of the switch is limited because of the mesh connections between the first and the second switching stages. These connection will be hard to be implemented in a large scale switch.

To achieve reliable communication over a terabit network a fast end-to-end error

recovery protocol was introduced. The proposed protocol achieves a fast error recovery with a limited number of control packets. The protocol is robust enough to work efficiently in congested and uncongested states. The proposed protocol is based on using periodic group acknowledgments and this minimizes the number of control cells and saves the control bandwidth while achieving high throughput.

6.2 Direction for Future Research

- The main functions of an ATM switch are to set up, serve, and release ATM connections. Much attention is required for achieving high speed in setting up and releasing a connection. The number of connections that can be established per second is an open field for research. In addition, further effort is required for implementing an efficient signaling manager to support a great number of connections per second.
- Adapting Available Bit Rate (ABR) flow control may increase the switch complexity considerably. A fast, simple, and fair flow control scheme is still an open subject for further research.
- Future work in developing an efficient flow controller is driven by market forces. Currently a lot of research are trying to design an efficient flow controller but until now the optimal controller has not been designed. A truly innovative flow control approach may allow beneficial product differentiation.
- The design of the shift-register based memory mentioned in Chapter 2 is a subject for future research. This kind of memory may be used in applications other than ATM. The memory speed, the memory density, and the complexity of the memory controller are the main issues that should be considered.
- Low power and low supply noise may be important considerations for future ATM systems, either to enable personal communication systems or simply to keep power dissipation under control.

- For ATM to get into the wireless world some issues should be considered while designing ATM switches. Due to the propagation delay in wireless transmission, the buffer size of the switch should be carefully chosen. The ATM switch architecture for wireless networks is an interesting subject for future research.
- Building a complete ATM photonic switch may be important for future ATM networks, either to provider a shorter delay or to provide higher capacity switches. Designing photonic switches is a wide field for future research, especially in cell buffering and other functions requiring data storage.

Bibliography

[1] Philip Smith, *Frame Relay Principles and Applications*, Addison-Wesley Publishing Ltd, England, 1993.

[2] S. Personick, "The Evolving role of telecommunications switching", *IEEE Communications Magazine*, vol. 31, pp. 20-24, January 1993.

[3] S. Minze, "Broadband ISDN and Asynchronous transfer mode (ATM)", *IEEE Communications Magazine*, vol. 27, pp. 17-25, September 1989.

[4] CCITT, *Recommendations on BISDN*, Geneve, 1990.

[5] S. Chowdhury, "Bandwidth allocation algorithm for packet video in ATM networks", *Computer Networks and ISDN systems*, vol. 26, pp. 1215-1223, May 1994.

[6] ATM User-Network Interface Specifications, Version 3.1, ATM FORUM, 1994.

[7] Raj Jain, "Congestion control in computer networks: Issues and trends", *IEEE Network Magazine*, vol. 4, pp. 24-30, May1990.

[8] J. Boudec, "The Asynchronous transfer mode: A tutorial", *Computer Networks and ISDN Systems*, vol. 24, pp. 279-309, May 1992.

[9] B. Vickers and T. Suda, "Connectionless services for public ATM networks", *IEEE Communications Magazine*, vol. 32, pp. 34-42, August 1994.

[10] P. Barri and J. Goubert, "Implementation of a 16x16 switching element for ATM switches", *IEEE Journal on Selected Areas in Communications*, vol. 9, No. 5, pp. 751-757, June 1991.

[11] M. De Prycker and M. De Somer, "Performance of an independent switching network with distributed control", *IEEE Journal on Selected Areas in Communications*, vol. 5, No. 8, pp. 1293-130, October 1987.

[12] G. Perucca, "Research on advanced switching techniques for the evolution to ISDN and broadband ISDN", *IEEE Journal on Selected Areas in Communications*, vol. 5, No. 8, pp. 1356-1364, October 1987.

[13] Y. Shobatake and T. Kodama, "A cell switching algorithm for the buffered Banyan network", *Proc. Int. Conf. Commun. (ICC'90)*, vol. 2, pp. 754-760, April 1990.

[14] J. Turner, "Design of a broadcast packet switching network", *IEEE Trans. Commun.*, vol. 36, No. 6, pp. 734-743, June 1988.

[15] A. Forcina, T. Distefano, and E. Taormina, "A multicast broadband switch module in a hybrid ATM environment", *Proc IEEE Int. Conf. Commun. (ICC'89)*, vol. 1, pp. 111-117, Boston, Mass., June 1989.

[16] T. Koinuma, "An ATM switching system based upon a distributed control architecture", *Proc. Int. Switching Symp. (ISS'90)*, vol. 5, pp. 21-26, Stockholm, May 1990.

[17] P. Newman, "A fast packet switch for the integrated services backbone network", *IEEE Journal on Selected Areas in Communications*, vol. 6, No. 9, pp.1468-1479, December 1988.

[18] Q. Ta and J. Meditch, "A high speed integrated services switch based on 4x switching elements", *Proc. IEEE INFOCOM*, pp. 1164-1171, San Francisco, CA, June 1990.

[19] K.E. Batcher, "Sorting networks and their applications", *Proc. AFIPSSrping Joint Comp. Conf.*, vol. 32, pp. 307-314, 1968.

[20] G. Bruzzi and A. Pattavina, "Performance evaluation of an input-queued switch with internal speed-up and finite output queues", *Proc. IEEE GLOBECOM'90*, San Diego, CA, pp. 1455-1459, December 1990.

[21] R.S. Bubenik and J.S. turner, "Performance of a broadcast packet switch", *IEEE Transaction in Communications*, vol. 37, No. 1, pp. 60-69, January 1989.

[22] D.X. Chen and J.W. Mark, "Performance analysis of output buffered fast packet switches with bursty traffic loading", *Proc. IEEE GLOBECOM'91*, Phoenix, AZ, pp. 455-459, December 1991.

[23] X. Chen, I. Lambadris, and J.F. Hayes, "A General unified model for performance analysis of multicast switching", *Proc. IEEE GLOBECOM'92*, Orlando, FL, pp. 1498-1502, December 1992.

[24] H. Ahmadi, and W.E. Denzel, "A survey of modern high-performance switching techniques", *IEEE journal on Selected Areas in Communications*, vol. 7, No. 7, pp. 1091-1103, September 1989.

[25] W. T. Chen, Y. R. Chang, and C. F. Huang, "A low-cost self-routing multicast network", *Proc. IEEE ICC'93*, Geneva, Switzerland, pp. 1691-1695, May 1993.

[26] J. Ding, "Nonuniform traffic analysis of multistage interconnection networks with split buffers", *Proc. IEEE ICC'93*, Geneva, Switzerland, pp. 1691-1695, May 1993.

[27] G. Bianchi and J. S. Turner, "Improved queuing analysis of shared buffer switching networks", *IEEE/ACM Transaction Networking*, vol. 1, No. 4, pp.482-490, August 1993.

[28] X. Chen, J. Hayes, and M. Ali, "Performance analysis of cyclic-priority input access method for a multicast switch", *Proc. IEEE INFOCOM'91*, Miami, FL, pp. 1189-1195, April 1991.

[29] M. Decina, P. Giacomazzi, and A. Pattavina, "Connectionless switching by the asynchronous shuffle out network", *Proc. IEEE INFOCOM'91*, Miami, FL, pp. 1189-1195, April 1991.

[30] G. Anido and A. Seeto, "Multipath interconnection: a technique for reducing congestion with fast packet switching fabrics", *IEEE Journal on Selected Areas in Communications*, vol. 6, No. 9, pp. 1480-1488, December 1988.

[31] M. de Prycker, *Asynchronous Transfer Mode*, Second Edition, Ellis Horwood Limited, UK, 1993.

[32] D. McDysan and D. Spohn, *ATM: Theory and Application*, McGraw-Hill, Inc., USA, 1995.

[33] H. Saito, *Teletraffic Technologies in ATM*, Artech House, USA, 1994.

[34] A. Acampora, *An Introduction to Broadband Networks*, Plenum Press, New York, 1994.

[35] M. Montpetit and M. Cote, "LAN interconnection by on board switching satellites", *Proceedings of the Canadian Conference of Electrical and Computer Engineering*, Montreal, Que., pp. 145-147, September 1995.

[36] J. Terry, "Alternative technologies and delivery systems for broadband ISDN system access", *IEEE Communications Magazine*, vol. 30, pp. 58-64, August 1992.

[37] R. Gejji, "Mobile multimedia scenario using ATM and microcellular technologies", *IEEE Trans. Vehic. Tech.*, vol. 43, pp. 699-703, August 1994.

[38] *ATM User-Network Specification*, Version 3.1, The ATM Forum, 1994.

[39] Fayez Elguibaly, "Design and analysis of arbitration protocols", *IEEE Transactions on Computers*, vol. 38, No. 2, pp. 161-171, February 1989.

[40] Y. Yeh, M. Hluchyj, and A. Acampora, "The knockout switch: A simple, modular architecture for high-performance packet switching", *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 1274-1282, October 1987.

[41] P. Newman, "ATM technology for corporate networks", *IEEE Communications Magazine*, vol. 30, pp. 90-101, April 1992.

[42] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space-division packet switch", *IEEE Trans. Comm.*, vol. COM-35, pp. 1347-1356, December 1987.

[43] A. Pattavina, "Nonblocking Architecture for ATM switching", *IEEE Communications Magazine*, pp. 38-48, February 1993.

[44] M. Hluchyj and M. Karol, "Queueing in high-performance packet switching", *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1587-1597, December 1988.

[45] J. Garcia-Haro and A. Jajszcyk, "ATM shared-memory switching architectures", *IEEE Network Magazine*, vol. 8, pp. 27-40, July/August 1994.

[46] E. Zegura, "Architectures for ATM switching systems", *IEEE Communications Magazine*, vol. 31, pp. 28-37, February 1993.

[47] C. Fayet, A. Jacques, and G. Pujolle, "High speed switching for ATM: the BSS", *Computer Networks and ISDN Systems*, vol. 26, pp. 1225-1233, May 1994.

[48] W. Denzel, A. Engbersen, and I. Iliadis, "A flexible shared-buffer switch for ATM at Gb/s rates", *Computer Networks and ISDN Systems*, vol. 27, pp. 611-624, January 1995.

[49] H. Kim, "Design and performance of Multinet switch: a multistage ATM switch architecture with partially shared buffers", *IEEE/ACM Trans. Networking*, vol. 2, pp. 571-580, December 1994.

[50] R. Bianchini and H. Kim, "The Tera project: a hybrid queuing ATM switch architecture for LAN", *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 673-685, May 1995.

[51] T. Chen and S. Liu, "Management and control functions in ATM switching systems", *IEEE Network Magazine*, vol. 8, pp. 27-39, July/August 1994.

[52] R. Black, I. Leslie, and D. McAuley, "Experiences of building an ATM switch for the local area", *Proceedings of SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, London, England, pp. 158-167, August 1994.

[53] J. Cox Jr., M. Gaddis, and J. Turner, "Project Zeus", *IEEE Network Magazine*, vol.7, pp. 20-30, March 1993.

[54] M. Collivignarelli, A. Daniele, P. De Nicola, L. Licciardi, M. Torolla, and A. Zappalorto, "A complete set of VLSI circuits for ATM switching", *1994 IEEE GLOBECOM*, San Franciso, CA, pp. 134-138, December 1994.

[55] C. Mead and L. Conway, *Introduction to VLSI System Design*, Addison-Wesley Publishing Company, USA, 1980.

[56] N. Weste and K. Eshragian, *Principles of CMOS VLSI Design: A Systems Perspective*, Addison-Wesley Publishing Company, Don Mills, 1985.

[57] D. Perry, *VHDL*, Second Edition, McGraw-Hill, Inc., USA, 1994.

[58] B. Prince, *Semiconductor Memories*, Second Edition, John Wiley & Sons Ltd., England, 1991.

[59] A. McAuley and C. Cotton, "A self-testing reconfigurable CAM", *IEEE J. Solid-State Circuits*, vol. 26, pp. 257-261, March 1991.

[60] S. Nojima, E. Tsutsui, H. Fukuda, and M, Hashimoto, "Integrated services packet network using bus-matrix switch, *IEEE Journal on Selected Areas in Communications*, vol. 5, No. 10, pp. 1284-1292, October 1987.

[61] F. Tobagi, "Fast packet switch architecture for broadband integrated services digital networks", *Proc. IEEE*, vol. 78, No. 1, pp.133-166, January 1990.

[62] M. Hluchyj, and M. Karol, "Queuing in high performance packet switching", *IEEE Journal on Selected Areas in Communications*, vol. 6, No. 9, pp. 1587-1597, December 1988.

[63] R. Awdeh and H. Mouftah, "Broadband packet switch architectures", *Proc. photonics'93: 3rd IEEE Int. Workshop on Photonic Networks, Components, and Applications*, Atlanta, GA, pp. 183-188, September 1993.

[64] M. Listanti and A. Roveri, "Switching structures for ATM", *Computer Commun.*, vol. 12, No. 6, pp. 349-358, December 1989.

[65] Y. Oie, "Survey of the performance of non-blocking switches with FIFO input buffers", *Proc IEEE ICC'90*, pp. 737-741, April 1990.

[66] M. Akata, S. Karube, T. Sakamoto, T. Saito, S. Wakasugi, S. Yoshida, H. Matsuno, and H. Shibata, "A Scheduling Content-Addressable Memory for ATM Space-Division Switch Control", *ISSCC Dig. Tech. Papers*, pp. 244-245, 1991.

[67] K. Eng, M. Hluchyj, and Y. Yeh, "A Knockout switch for variable length packets", *Proc IEEE ICC'87*, Seattle, WA, pp. 794-799, June 1987.

[68] J. Choi, C. Un, and B. Shin, "Design of cascade ring multicast switch", *Electronics Letters*, vol. 28, No. 14, July 1992.

[69] S. Kumar and D. Agrawal, "On Design of a Shared-Buffer based ATM Switch for Broadband-ISDN", *In Proceedings of 13th International Pheonix Conference on Computers and Communications*, Arizona, pp. 377-383, April 1994.

[70] J. Burgin and D. Dorman, "Broadband ISDN Resource Management: The Role of Virtual Paths", *IEEE Communications Magazine*, vol. 29, pp. 44-48, September 1991.

[71] C. Park and C. Un, "On the design of large ATM switch using star couplers and tunable devices with restrained tuning range", *IEICE Transactions on Communications*, vol. E77, pp. 469-476, April 1994.

[72] E. Munter, J. Parker, and P. Kirkby, "A high-capacity ATM switch based on Advanced Electronics and optical technologies", *IEEE Communications Magazine*, pp. 64-71, November 1995.

[73] A. Cisneros and C. Brackett, "A large ATM switch based on memory switches and optical star couplers", *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1348-1360, October 1991.

[74] A. Jajszczyk and W. Kabacinski, "A growable ATM switching fabric Architecture", *IEEE transactions on Communications*, vol. 43, pp. 1155-1162, February 1995.

[75] K. Eng, M. Karol, and Y. Yeh, "A growable packet (ATM) switch architecture: Design principles and applications", *IEEE Transaction on Communications*, vol. 40, pp. 423-430, February 1992.

[76] M. Karol and C. Lin, "Performance analysis of a growable architecture for broad-band packet (ATM) switching", *IEEE Transaction on Communications*, vol. 40, pp. 431-439, February 1992.

[77] T. Zhang and A. Somani, "DIRSMIN: A fault-tolerant switch for B-ISDN applications using dilated reduced-stage MIN", *IEEE INFOCOM'95*, Boston, Mass., pp. 643-649, April 1995.

[78] K. Eng, M. Karol, G. Cyr, and M. Pashan, "A 160-Gb/s ATM switch prototype using the concentrator-based growable switch architecture", *GLOBECOM'95*, Singapore, pp. 550-554, November 1995.

[79] J. Garcia-Haro and A. Jajszczyk, "ATM shared-memory switching architectures", *IEEE Network Magazine*, vol. 8, pp. 18-26, July 1994.

[80] R. Awdeh and H. Mouftah, "Survey of ATM switch architectures", *Computer Networks and ISDN Systems*, vol. 27, pp. 1567-1613, November 1995.

[81] M. Beshai and E. Munter, "High Capacity ATM switch", US patent 5,745,486, 1998.

[82] N. Mirfakhraei, "Design of a CMOS buffered switch for a gigabit switching network", *Journal of Solid-State Circuits*, vol. 30, pp. 11-18, January 1995.

[83] F. El-Guibaly, A. Sabaa, and D. Shpak, "A New Shift-Register Based ATM Switch", *In Proceedings of EtaCom.*, Portland, Oregon, May 1996.

[84] H. Saito, H. Yamanaka, Y. Yamada, and M. Tuzuki, "Multicast function and its LSI implementation in a shared multibuffer ATM switch", *In Proceedings of IEEE INFOCOM'94 The Conference on Computer Communications*, Toronto, pp. 3a.4.1-3a.4.8, June 1994.

[85] H. Chao and B. Choe, "Design and analysis of a large-scale multicast output buffered ATM switch", *IEEE/ACM Transactions on Networking*. vol. 3, pp. 126-138, April 1995.

[86] M. Anagonostou and E. Protontarios, "Performance analysis of the selective repeat ARQ protocol", *IEEE Trans. Comm.* vol. 34, pp. 127-135, 1986.

[87] M. Peyravian, "An improved selective repeat protocol and its performance in high-speed environments", *Computer Networks and ISDN systems*, vol. 26, pp. 1595-1605, September 1994.

[88] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip", *IEEE Journal on Selected Areas in Communication*, vol. 9, pp 1265-1279, October 1991.

[89] M. Katevenis, P. Vatsolaki, and A. Efthymiou, "Pipelined memory shared buffer for VLSI switches", *In Proceedings of SIGCOMM'95*, Cambridge, pp. 39-48, May 1995.

[90] H. Kroner, G. Hebuterne, P. Boyer, and A. Gravey "Priority management in ATM switching nodes", *IEEE Journal on Selected Areas in Communications*, vol. 9, No. 3, pp 418-427, April 1991.

[91] I. Habib and T. Saadawi, "Multimedia traffic characteristics in broadband networks", *IEEE Communications Magazine*, pp 48-54, July 1992.

[92] H. Saito, M. Kawarasaki, and H. Yamada, "An Analysis of statistical multiplexing in an ATM transport network", *IEEE Journal on Selected Areas in Communications*, vol. 9, No. 3, pp 359-367, April 1991.

[93] P. Sen, B. Maglaris, N. Rikll, and D. Anastassiou, "Models for packet switching of variable-bit-rate", *IEEE Journal on Selected Areas in Communications*, vol. 7, No. 5, pp 865- 869, June 1989.

[94] D. Ferrari and D. Verma, "Buffer Allocation for Real-time Channels in Packet Switching Networks", *Tech Rep. TR-90-022, International Computer Science Institute*, Berkeley, CA, June 1990.

[95] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis* Addison-wesley, Reading, Massachusetts, 1987.

[96] ITU-T Recommendation I.610, B-ISDN Operation and Maintenance Principles and Functions, November 1994.

[97] GR-1248-CORE, Generic Requirements for Operations of Broadband Switching Systems, Bellcore, Aug. 1994.

[98] M. Hluchyj and A. Bhargava, "Queueing disciplines for integrated fast networks", *In Proceedings of ICC'92*, Chicago, pp. 990-996, June 1992.

[99] N. Oba, K. Suzuki, H. Kobayashi, and T. Nakamura, "Startcore: A high-speed ATM switching system", *In Proceedings of GLOBECOM'94*, San Fransisco, pp. 139-143, December 1994.

[100] F. Bonomi and K. Fendick, "The rate-based flow control framework for the available bit rate ATM service", *IEEE Network Magazine*, pp. 25-39, March/ April 1995.

[101] H. Kung and Robert Morris, "Credit-based flow control for ATM networks", *IEEE Network Magazine*, pp. 40-48, March/April 1995.

[102] K. Ramakrihnan and P. Newman, "Integration of rate and credit schemes for ATM flow control", *IEEE Network Magazine*, pp. 49-56, March/April 1995.

[103] F. El-Guibly, S. Agarwal, "Design and performance analysis of shift register-based ATM switch", *IEEE pacific Rim Conf. on Communications, Computers and Signal Processing*, Victoria, B.C., pp. 70-73, August 20-22, 1997.

[104] S. Agarwal and F. El-Guibly, "Modeling of shift register-based ATM switch", *IEEE 8th Great Lakes Symposium on VLSI*, Lafayette, Louisiana, pp. 146-151, Feburay. 19-21, 1998.

[105] S. Agarwal, "Design and performance of a new ATM switch", Master of Applied Science, Electrical and computer Engineering Department, University of Victoria, 120 pages, December 1998.