



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Performance Evaluation of an ATM Multiplexer

by
Anil Chandan

A THESIS

submitted to the School of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of

MASTER OF APPLIED SCIENCE
in
Electrical Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-60029-2

Canada



UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Table of Contents

Abstract	iii
Acknowledgements	iv
List of Figures	v
List of Tables	viii
Acronyms	ix
1 Introduction	1
1.1 Background.....	1
1.2 Problem Statement.....	5
1.3 Thesis Outline	9
2 B-ISDN	11
2.1 Introduction	11
2.2 Driving Forces Behind B-ISDN	11
2.3 B-ISDN Service Classes	12
2.4 B-ISDN Service Rates.....	13
2.5 Architectural Model	14
2.6 Reference Configurations	15
2.7 Protocol Model	16
2.8 Network Aspects	17
2.9 Evolution Towards B-ISDN	20
3 ATM	21
3.1 Introduction	21
3.2 The ATM Cell.....	22
3.2.1 The Cell Header	22
3.2.2 The Cell Information Field.....	23
3.3 Virtual Path Capabilities	23
3.4 Cell Synchronization.....	27
3.5 B-ISDN User Network Interfaces.....	28
3.6 ATM based Protocol Model for B-ISDN.....	29
3.7 Performance Measures for ATM.....	31
3.8 ATM Congestion Control	32
4 Model of an ATM Multiplexer	34
4.1 Traffic Model of a Bursty ATM Source.....	34
4.2 Queuing Model of an ATM Multiplexer.....	37
4.3 Solution.....	39
4.3.1 Markov Chain Model	39
4.3.2 Petri Net Model.....	41
4.3.3 Drawbacks with the above methods	44
4.3.4 Closed Form Solution	44
4.3.5 Results.....	46

5	Extensions to the Model	54
5.1	Non-identical Traffic Sources.....	58
5.2	Multimedia Sources	66
6	Conclusions	76

Appendices

A	SONET	79
A.1	Introduction	79
A.2	Synchronous Optical Equipment	80
A.3	STS-1 Frame.....	82
A.4	Frame Structure of The STS-N Signal.....	83
A.5	STS-Nc Signal.....	84
A.6	SONET Overhead Channels	84
A.7	Layered Overhead and Transport Functions	88
A.8	Multiplexing Synchronization Using Pointers	89
A.9	Sub STS-1 Multiplexing.....	91
A.10	Section/Line Data Comm. Channel Protocol Stack.....	91
B	Evaluation of α, β, λ and μ	93
B.1	Call Arrival Rate α and Call Service Rate β	93
B.2	Packet Arrival Rate λ and Packet Service Rate μ	93
C	Gauss-Seidel Method	94
C.1	Gauss-Seidel Method	94
C.2	Program Developed For Gauss-Seidel Method.....	94
D	Closed Form Model	97
D.1	One Type Of Traffic Sources	97
D.2	Two Types Of Traffic Sources.....	100
D.3	Multimedia Sources With Two Traffic Types	104
E	QNAP2 Programs	107
E.1	One Type Of Traffic Sources.....	107
E.2	Two Types Of Traffic Sources.....	110
E.3	Multimedia Sources With Two Traffic Types	114
	Bibliography	118

Abstract

Asynchronous Transfer Mode (ATM) has been proposed by the CCITT as the target information transport mechanism for the broadband network (B-ISDN) of the future. ATM promises high throughput capacity combined with flexibility and efficiency in terms of network resource (e.g. bandwidth) usage. Small fixed length cells carrying data and routing information are switched within the network at speeds up to a few Gigabits per second. ATM, being an asynchronous technique, suffers from cell loss due to congestion within the network nodes. It is important that this information loss be kept small; of the order of the information loss in the transmission medium (Bit error ratio $\approx 10^{-9}$ for optical fibre). The congestion within an ATM network also introduces cell delays. It is desirable that the cell delay within the network is kept at a constant small value.

In this thesis, a queuing model for an ATM based Multiplexing node is developed. In this model, it is assumed that no cell queuing is done within the multiplexer for the purpose of transmission at a later time. This introduces higher cell loss but simplifies the multiplexing (switching) algorithm which is key to ATM. This also gives constant cell delays per network node. The total network delay suffered by a cell is the sum of propagation delay, and cell transmission and buffering delay within each node the cell traverses.

A closed form analytical solution is derived for the above described model and ATM cell loss probabilities (or cell blocking probabilities) are computed for a multiplexer assuming identical sources at the input. This enables us to estimate the bandwidth gain for various levels of cell loss probabilities. The model is also extended to include non-identical traffic sources and multimedia sources at the input of the multiplexer. The cell loss probabilities are studied for these cases as well. All the results are verified via simulation.

Acknowledgements

First I would like to acknowledge my supervisor, Dr. Nicolas D. Georganas, for his excellent guidance. It has been a great opportunity to work under him and benefit from his deep knowledge of the subject. I appreciate the patience he has shown with me.

I am thankful to my employer, Bell Northern Research, for allowing me to continue my education.

I must also thank my colleagues Mr. Anil Gupta and Mr. Oscar Avellaneda for their valuable suggestions.

Last but not the least, I thank my wife, Vandana, and my sister, Brij, for their continued encouragement.

List of Figures

1.1	Telecommunication Service Requirements.....	3
1.2	Future Transport Technologies.....	4
1.3	Typical Telecommunication Network.	6
1.4a	Multiplexing Node.....	6
1.4b	Switching Node.....	7
1.5	Model of a Multiplexing Node.....	10
2.1	Classification of B-ISDN Services.....	12
2.2	Basic Architectural Model of B-ISDN.	14
2.3	ISDN (B-ISDN) Reference Configuration.	15
2.4	B-ISDN Physical Configuration.....	16
2.5	B-ISDN Protocol Stack.....	17
2.6	Termination of Adaptation Layer.....	18
2.7	Interworking of B-ISDN with existing Networks.....	20
3.1	ATM Cell.....	24
3.2a	ANSI ATM Cell Header.....	25
3.2b	ETSI ATM Cell Header.....	25
3.2c	CCITT ATM Cell Header for UNI.....	25
3.2d	CCITT ATM Cell Header for NNI.....	25
3.3a	Cell Error Detection/Correction Scheme.....	26
3.3b	Operation of an ATM Cell Receiver.....	26
3.4	ATM Adaptation Layer Overhead and Cell Information Field.....	26
3.5a	Self Structured ATM Cells.....	27
3.5b	Self Structured ATM Cell Stream within an External Synchronous Frame.....	28
3.5c	ATM Cell Stream Synchronized within and External Synchronous Frame.....	28
3.6	Structure of 600 Mbps UNI.....	29

3.7a	User Plane (Flow of Data).	30
3.7b	Control Plane (Flow of Signalling)	30
3.8	Synchronous and Asynchronous Multiplexing.	32
3.9	ATM Cell Source Preventive Control at the input of the Multiplexer.	33
4.1	Model of a Bursty ATM Traffic Source.	36
4.2a	Call Level Markov Chain.	37
4.2b	Packet Level Markov Chain.	37
4.3	Model of an ATM Multiplexer.	38
4.4a	Markov Chain For Multiplexer with N inputs.	40
4.4b	Transitions from state i,j in the Multiplexer Markov Chain	41
4.5	Petri Net Model of Multiplexer with N inputs.	42
4.6	Cell Loss Probabilities vs Trunk Capacity.	47
4.7	Variation of $P(x)$ for different values of CCS.	50
4.8	Probability of cell loss against multiplexer trunk capacity (for different values of CCS).	51
4.9	Prob. of Cell Loss Vs Source Utilization for different values of Bandwidth ratio provided.	51
4.10	Variation of $P(x)$ for different values of Burstiness Factor.	52
4.11	Probability of cell loss against multiplexer trunk capacity (for different values of Burstiness Factor).	52
4.12	Bandwidth Gain achieved for a given source utilization and cell loss probability.	53
4.13	Bandwidth Gain achieved for a given Burstiness Factor and cell loss probability.	53
5.1a	Complete Partitioning of Cell Buffers.	56
5.1b	Complete Sharing of Cell Buffers.	56
5.1c	Sharing with Minimum Allocation.	57
5.1d	Sharing with Maximum Limit.	57
5.2	Cells Generated By Type-1 and Type-2 Sources as they appear on the output Trunk.	58
5.3	Cells Loss Probabilities for High and Low Priority Cells Generated by different sources in Complete Sharing.	62

5.4	Cell Loss Probabilities in Sharing with Minimum Allocation Scheme.....	64
5.5	Cell Loss Probabilities in Sharing with Maximum Limit Scheme	66
5.6	Markov Chain Model of Multiplexer	68
	(Each input source generates two traffic types).	
5.7	Cell Loss Probabilities For Multimedia Sources using Minimum	69
	Complete Partitioning scheme	
5.8	Cell Loss Probabilities for High and Low Priority Cells	72
	for Multimedia Sources with two traffic types in Complete Sharing.	
5.9	Cell Loss Probabilities for Multimedia Sources using Sharing	73
	with Minimum Allocation scheme	
5.10	Cell Loss Probabilities for Multimedia Sources using Sharing	75
	with Maximum Limit scheme	
A.1a	Point to point Configuration.	80
A.1b	Repeater Configuration.....	81
A.1c	Add/Drop Multiplexer Configuration.	81
A.1d	HUB Configuration.....	81
A.1e	DCS Configuration.....	82
A.2	STS - 1 Frame.	83
A.3	STS - N Frame.	83
A.4	Section, Line and Path Overhead Bytes.	85
A.5	SPE Pointer Function (STS - 1 Frame).....	86
A.6	Section/Line Data Comm. Protocol Stack.	86
A.7	SONET Overhead Boundaries.....	87
A.8	SONET Overhead Hierarchy.....	87
A.9a	SPE Pointer Action (-ve Stuffing Byte).	90
A.9b	SPE Pointer Action (+ve Stuffing).	90
A.10	VT Oriented STS - 1 Frame.....	92
A.11	Virtual Tributary Groups.	92

List of Tables

- 4.1 Probability of finding a given number of Multiplexer inputs transmitting..43
Using Markov Chain model (solved by Gauss-Seidel method), Petri Net
model (solved by GSPN, accuracy 0.0004) and closed form solution.
- 4.2 Cell Loss Probabilities from Closed Form Solution and Simulation.....48
- 5.1 Cell Loss Probabilities for High Priority and Low Priority Cells for two..61
types of traffic sources in Complete Sharing case.
- 5.2 Cell Loss Probabilities for High and Low Priority Cells for Multimedia .71
sources with two traffic types in Complete Sharing case.

Acronyms

AAL	ATM Adaptation Layer
ACF	Access Control Field
ADM	Add/Drop Multiplexer
ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
BBTG	CCITT study group XVIII task Group on ISDN Broadband Aspects
BER	Bit Error Ratio
BF	Burstiness Factor
B-ISDN	Broadband Integrated Services Digital Network
BOD	Bandwidth On Demand
BRA	Basic Rate Access (ISDN)
BWG	Bandwidth Gain
CBO	Continuous Bit Oriented service
CCITT	Comite Consultatif International Telegraphique et Telephonique
CCS	Hundred Call Seconds (per hour)
CO	Central Office
CP	Complete Partitioning, Control Plane
CRC	Cyclic Redundancy Check
CRF	Connection Related Function
CS	Complete Sharing
DCS	Digital Cross-connect Switch
DSI	Digital Signal Interface
ET	Extended Terminations
ETSI	European Telecommunications Standards Institute
GaAs	Gallium Arsenide

Gbps	Giga-bits (10^9 bit/sec) per second rate
GCF	Generic Control Field
GSPN	Generalized Stochastic Petri Net tool
HCS	Header CheckSum (ATM cell - ANSI)
HDTV	High Definition Television
HEC	Header Error Check field (ATM Cell - CCITT)
ISDN	Integrated Services Digital Network
LOH	Line Overhead
MUX	Multiplexer
NA	Network Adapter
NIWU	Network InterWorking Unit
NNI	Network Network Interface
OC-N	Optical Carrier signal - level N
OSI	Open System Interconnection
POTs	Plane Old Telephone services
PRA	Primary Rate Access (ISDN)
PSTN	Public Switched Telephone Network
QOS	Quality Of Service
RE	Remote Electronics
SMA	Sharing with Minimum Allocation
SML	Sharing with Maximum Limit
SOH	Section Overhead
SONET	Synchronous Optical NETWORK
STDM	Synchronous Time Division Multiplexing technique
STM	Synchronous Transport Multiplex
STS-N	Synchronous Transport Signal - level N
TA	Terminal Adaptor

TE	Terminal Equipment
TOH	Transport Overhead
UNI	User Network Interface
UP	User Plane
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
VT	Virtual Tributary

Chapter 1

Introduction

1.1 Background

The advancements in technology in recent years have given rise to new telecommunication services. Individualized newspapers and manipulatable movies have been proposed [LIPP87], video browsing and transport access to multiple databases have been demonstrated [BULI87]. In Europe, operational video communication networks have indicated a high user interest in audio, graphic and image interactive services [SEGU87].

Keeping in mind the rapid pace of development in technology today, it is quite difficult to predict the nature of user communication services that will emerge in the future. The type of requirements that these services [WRIG89] will demand from the communication network are equally unpredictable. With the conception of ISDN (Integrated Services Digital Network, CCITT I-series recommendation, 1984), there is a drive towards integration of all the telecommunication services into one network; all the services being available to the user via a single UNI (User Network Interface) or a finite set of UNIs. By looking at Figure [1.1] [WOOD89], even today's services have varied requirements [MINZ89]:

- Service Bandwidth requirements vary from low bit rate (a few bits per second for low speed telemetry data) to very high bit rate (a few Mbits per second for HDTV).
- Some services are quite delay sensitive (real time voice/video) whereas others can tolerate appreciable network delays (Electronic mail).
- For some services, information loss is tolerable (or is compensated by upper layer protocol without service degradation) while other services can not tolerate even slight information loss.
- Services can be CBO (Continuous Bit Oriented) or non-bursty in nature to very bursty, putting out high bit rates in short periods with silent periods in between.
- Isochronous services (voice, video) must preserve time transparency. Non-isochronous services (file transfer) can tolerate variable delay and even out-of-order delivery of information.
- Interactive services need two way (either symmetrical or asymmetrical) data paths (telephone). Distributive services require only one way path for the information (cable TV).
- Connection oriented services (telephone call) have a connection establishment phase followed by the information transfer phase. The connection is terminated at the end of the information

transfer. In connectionless services, there are no connection establishment/termination phases. The routing information and the data is sent in the same message or packet through the network (X.25).

- Some services require transparent transport of bits whereas others can take advantage of code compression to conserve bandwidth (compressed video) or code conversion to overcome terminal incompatibilities.
- Connection types vary from simple point-to-point unidirectional (or bidirectional) to parallel connections in multimedia communications among multiple users.

The requirements posed by a given service may be any combination of the above. Currently, due to cost and technology constraints, specialized networks [PRYC89] have been designed to optimally handle the requirements of individual services (PSTN for switched telephone service, X.25 for packet switched service). Very often, the designs (protocols and interfaces) have been proprietary to the manufacturer. It has been a nightmare to integrate even networks designed for the same type of services let alone different types of services.

B-ISDN (Broadband ISDN), currently being standardized by CCITT study group XVIII [CC0289], is being perceived as the network of the future. It will integrate the current and future services (Figure [1.1]) into one network. Recent advancements in technology have created a demand for new broadband services. At the same time, breakthroughs in single mode fibre optics and switching technologies [GIML85, HAYW87, TURN86, COUD87, GITM81] have made it feasible to design such a B-ISDN network.

In the past, various schemes based on current circuit switching [OREI88, WEIL88] and packet switching [TURN83], have been proposed for a future B-ISDN network by different authors. Various hybrid schemes [LUTZ87, WERN88, SPEA88] have also been proposed. The circuit switched networks have the following shortcomings in terms of B-ISDN requirements:

- Synchronous Time Division Multiplexing (STDM) techniques used in circuit switching have dedicated fixed rate channels and therefore limited bandwidth flexibility.
- STDM is unsuitable to bursty traffic. It has been designed for and is well suited to the continuous bit-oriented services. Peak rate allocation for each user results in very inefficient bandwidth utilization for bursty services.
- Bandwidth wastage due to non-asynchronous nature of transport. Even when a channel is not transmitting, it is using up the bandwidth dedicated to it.
- Mapping of low speed services to B-ISDN channel rates becomes too complex. There are about 2000 eight bit slots in a 150 Mbps interface [MINZ89] which gives an idea about the

complexity of dynamically allocating these slots to low speed circuits of varying bandwidth requirements.

- Sometimes, allocation of a low speed circuit assigned earlier may lockout a higher speed circuit because peak bandwidth of the high rate circuit is not available.

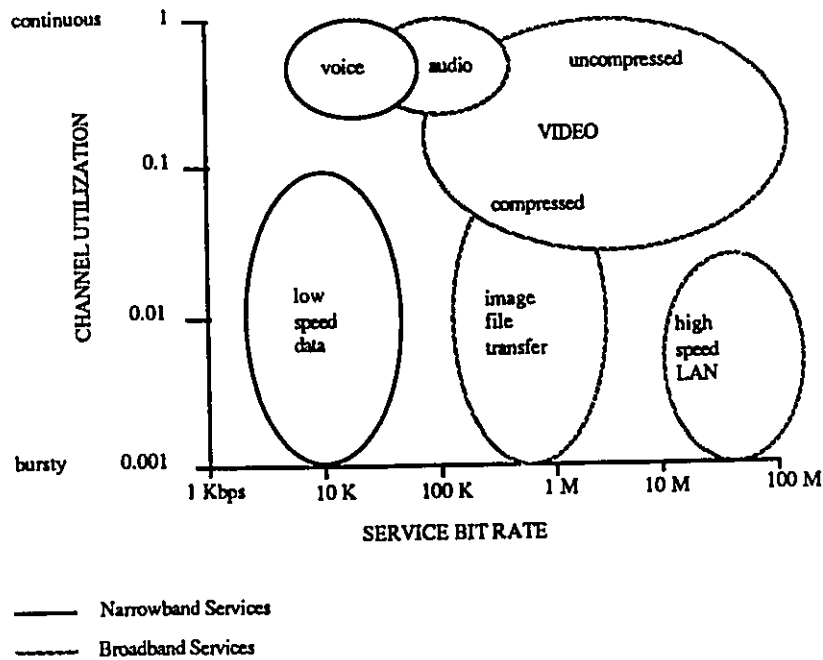


Figure [1.1]: Telecommunication Service Requirements (From [WOOD89])

The current packet switched networks have the following shortcomings in terms of B-ISDN requirements:

- packet switching uses cumbersome flow control and error control protocols between every node in the network [TENE81]. This introduces large nodal queuing delays which are unacceptable for a B-ISDN. Worst yet, these delays are variable from packet to packet.
- packet switching has low effective bandwidth.
- time transparency is lost in packet switched networks due to packet assembly/disassembly, and queuing delays.
- Information loss due to packet loss caused by buffer overflow or congestion.

Other hybrid solutions proposed for B-ISDN tend to resolve some of the problems but introduce complexity [OKAZ88, GRUB81] into the system.

New transport components will be required to satisfy the service flexibility and performance demanded from a B-ISDN network. A transport technology which allows for high communication speed and the largest degree of resource sharing will be required.

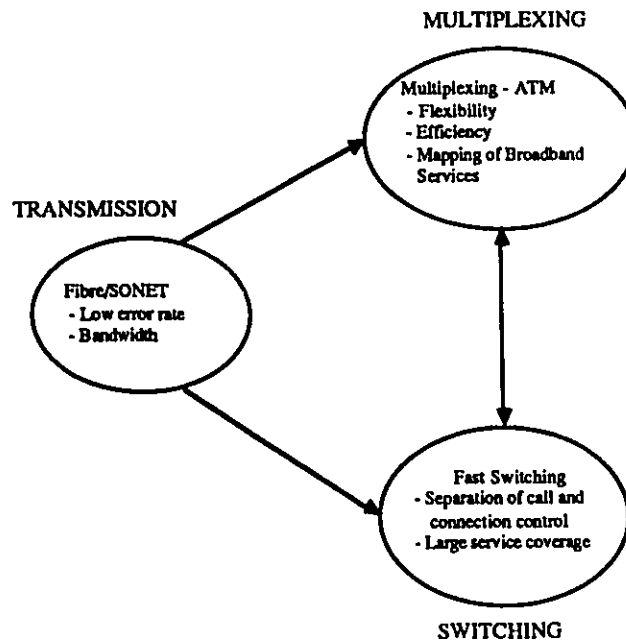


Figure [1.2]: Future Transport Technologies

As shown in Figure [1.2], the three main areas of transport technology are **Transmission**, **Switching** and **Multiplexing**. If we map these three on the OSI (Open System Interconnection) protocol layer model, in circuit switching all the three functions can be mapped to the physical layer. In packet switching, these functions are mapped to the first three layers. For a B-ISDN, a more structured approach is required.

Transmission technology is evolving towards a fibre based **SONET** (Synchronous Optical Network) environment. SONET [FLAN88], which will be described in Appendix A of this thesis, is definitely a good choice for high performance (low Bit Error Ratio (BER) and large bandwidth) transport media and synchronous layer-1 for B-ISDN.

In the switching area, fibre also has enabled switching technology that allows single cross-point to support a wide variety of traffic types with minimum compromise. Channel rates up to 1.2 Gbps are feasible today [T1S188] with silicon technology. Use of GaAs (Gallium Arsenide) technology in future will enable much higher rates.

CCITT recommendation I.121 [CC0289] designates ATM (Asynchronous Transfer Mode) as the target transfer mode for B-ISDN. The proposed ATM based protocol for B-ISDN (Figure [2.5]) is quite different from those used in current packet switched networks. The layer-2 and layer-3 protocols of packet switched networks (X.25) have been designed for high BER media (noisy telephone lines) and therefore have elaborate error checking and flow control mechanisms. Since fibre optic communication is virtually error free ($BER \approx 10^{-9}$), new protocols needed to be defined. In the proposed ATM based protocol stack for B-ISDN, error checking and flow control are pushed to higher layers so as to achieve fast switching and high throughput by simple protocols (that may be implemented in hardware) at the lower layers. The call control and the connection control are separated and assigned to different layers in the protocol stack. The ATM layer is responsible for connection control only. The call control is shifted to the signalling plane above. ATM, which is described in chapter [3.0], is a fixed block size (cell) based fast packet switching technique. Switched on the basis of VCI (Virtual Channel Identifier) and VPI (Virtual Path Identifier) in their headers, the ATM cells cut through the network nodes at a very high speed.

The effectiveness of fast switching depends on the connection multiplexing used. Multiplexing at the ATM layer is flexible and efficient. In terms of flexibility, ATM can handle bursty as well as stream (CBO) traffic. It gives bandwidth flexibility to the user through the same User-Network Interface. The cell header acts as a channel identifier (label) for an ATM channel much like a slot number in an STDM frame. The channel number of a cell changes as the cell moves from node to node in the network. In ATM, the service mix and the information transfer rates are decoupled from the characteristics of the switch. All the services share trunk bandwidth on statistical basis thus increasing the bandwidth efficiency (Figure [3.8]). The cells are assigned to a connection depending on the service requirements at the connection time. The ATM also promises Bandwidth On Demand (BOD) where a user can increase or decrease the bandwidth usage during the call period.

1.2 Problem Statement

In section [1.1], ATM has been introduced as the target transfer mode for B-ISDN. It offers great flexibility in bandwidth allocation through the assignment of small fixed length cells to services on an as required basis. It promises potential gains in bandwidth efficiency by statistically multiplexing bursty traffic sources. ATM appears to provide nearly ideal means of transport for the wide range of services to be supported by a B-ISDN.

However, statistical multiplexing through buffering at each node can lead to performance degradation in terms of variable network transfer delays and cell loss. The ATM must meet the most stringent performance requirements for all the services to be carried by B-ISDN. It must be able to transport (meet performance requirements of) the bursty packet mode services as well as Continuous Bit-Oriented (stream) services.

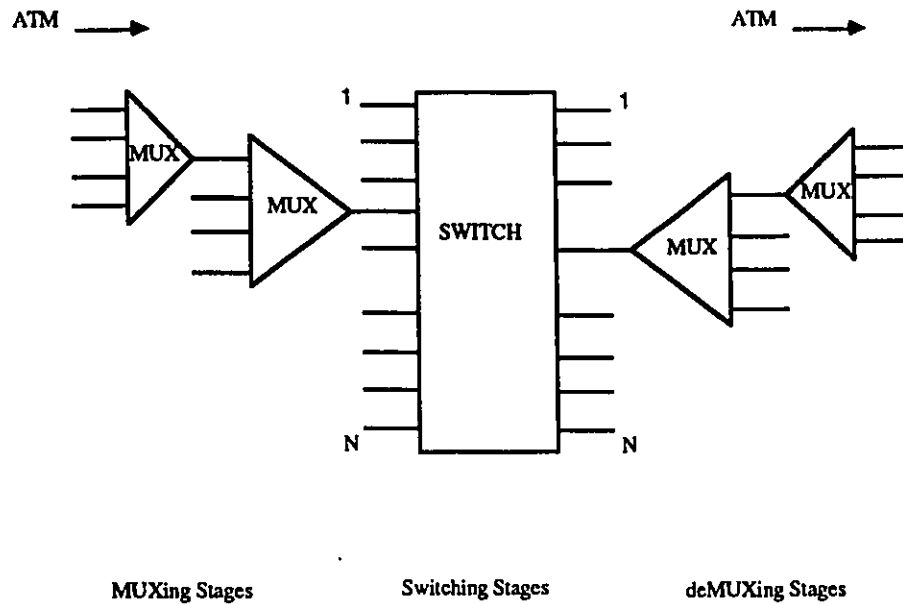


Figure [1.3]: Typical Telecommunication Network

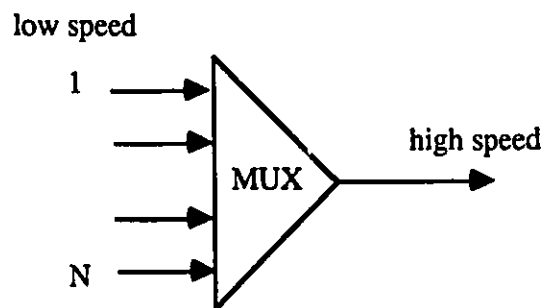


Figure [1.4a]: Multiplexing Node

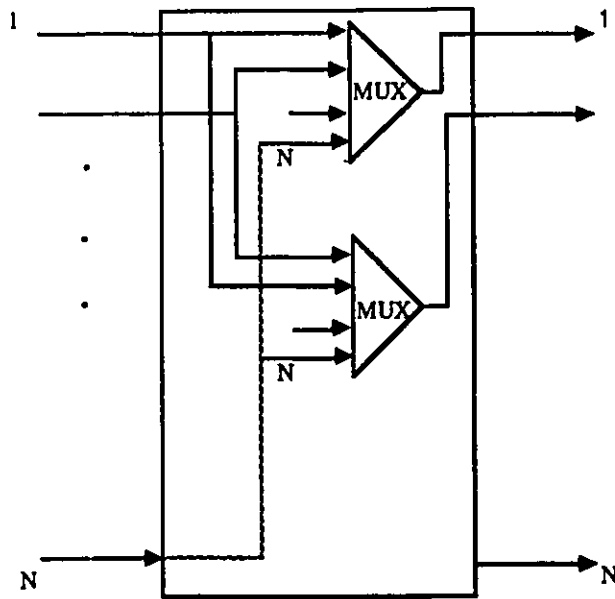


Figure [1.4b]: Switching Node

A telecommunication network (Figure [1.3]) consists of various multiplexing nodes and switching (Repeaters, Add/Drop Multiplexers, Concentrators and Cross-Connects) nodes. For the sake of simplicity, we assume that the performance of the whole network can be computed from the performance of individual network nodes. In this thesis, we will develop a queuing model of a multiplexing node as shown in Figure [1.4a]. In a multiplexing node, traffic from all the input sources is usually concentrated on a higher speed output trunk. In a switching node also, the traffic from the input sources is switched to the output links. Each input source may be transmitting to any of the output links at a time. Thus a switching node may be analyzed using the model of a multiplexer (Figure [1.4b]). Each output link is represented as a multiplexer which concentrates all the inputs of the switching node. The rate at which an input transmits into any one of these multiplexers is determined by the rate at which it (the input source) transmits into the output link of the switching node represented by the multiplexer.

Each input source connected to the multiplexer is assumed to be bursty in nature. A bursty source is characterized by a **Burstiness Factor (BF)** which is defined to be the ratio of peak rate to the average rate of the source. In order to fully characterize a bursty source, BF, peak rate and the average burst size must be specified.

In the queuing model proposed in this thesis, the ATM multiplexer uses a common pool of ATM cell buffers to be shared by all the services carried by the multiplexer. Other multiplexing schemes based on input buffering [FITZ89] and combination of input and output buffering [HLUC88] have been proposed in literature but it has been demonstrated that the common

buffering at the output gives best results. In our model, no cell queuing is done. This is necessary to eliminate the cell delay fluctuation introduced by the nodal queuing delays and to reduce the total cell delay.

Without queuing, the total cell delay is equal to the cell assembly/disassembly delays plus the cell transmission delay. This delay is of the same order as the delay in circuit switched networks. This scheme may have higher cell loss rate than those using cell queuing but it simplifies the design of the switch which is important for switches operating at broadband rates.

Figure [1.5] shows the proposed model of the multiplexer. R_i in Figure [1.5] is the peak input rate of the i th source and C is the output trunk capacity (rate). The multiplexer scans all its inputs every T seconds. The period T is determined by the fastest input source to be multiplexed and is equal to the time taken by the fastest source (when transmitting) to fill one cell buffer. The slower input sources (when transmitting) contribute a cell every k th scans where k is the ratio of the peak rate of the fastest source to that of the slower source. The cells collected every T seconds are stored in a common pool of cell buffers inside the multiplexer before being transmitted on the outgoing trunk. The total number of cell buffers inside the multiplexer is equal to the number of cells that can be transmitted on the high speed output trunk of the multiplexer in T seconds. If the number of cells collected in one scan exceeds the number of buffers, the excess cells are lost. We have studied the effect of traffic load and input burstiness on the probability of cell loss. Since no cell queuing is done, the burst length does not affect the blocking probabilities. We have also studied the bandwidth gain (ratio of the output trunk bandwidth required if input peak rates were allocated, to the actual output trunk bandwidth available in the case of ATM).

In the first model, all the inputs to the multiplexer are considered to be identical. Each input sends only one type of traffic into the multiplexer. The model is then extended to include non-identical input sources (with different service requirements) that can be classified into service type groups. The model is also extended to include identical input sources, each having more than one service type (multimedia [WERN89]) requirement. Different resource (cell buffer) sharing strategies [KRAI85, KRAI86, KRAI87, WONG86] have been considered for either case. These strategies are:

- Complete Partitioning (CP)

In this scheme, the cell buffers are divided among input sources on a static basis. A source can not use the buffers allocated to another source even if they were unoccupied.

- Complete Sharing (CS)

In this scheme, the cell buffers in a common pool are shared on an as required basis by the input sources.

- **Sharing with Minimum Allocation (SMA)**

In this scheme, some minimum amount of buffers are allocated to each source on static basis. The buffers that are not allocated to any specific source may be shared by all the sources on an as required basis.

- **Sharing with Maximum Limit (SML)**

In this scheme, the cell buffers are shared by all the input sources on an as required basis but there is a maximum limit on the number of buffers used by any particular source at one time.

In all the schemes that use sharing of buffers, the contention between the sources is resolved by assigning priorities to the sources during call establishment. For sources with the same priority, the buffer allocation is done by a predetermined scheme designed within the switch.

1.3 Thesis Outline

Chapter 2 of this thesis introduces activity relating to B-ISDN standards. It describes B-ISDN services, interfaces, reference model and the protocol model. This chapter also introduces ATM, the proposed transport mode for B-ISDN services.

Chapter 3 goes into details of ATM. It describes ATM cell fields, defines ATM performance measures, B-ISDN User-Network Interfaces and protocol model.

In chapter 4, our models for a bursty ATM source and a multiplexer are developed and solved. ATM cell blocking probabilities are computed and interpreted.

In chapter 5, the model is extended to include non-identical sources that may be classified into traffic groups. The model is also modified to include multimedia sources where each source transmits more than one type of traffic.

Chapter 6 presents the conclusions of this thesis.

Appendix A of this thesis introduces SONET as it stands now under definition by the T1X1 committee of the American National Standards Institute (ANSI). SONET can provide low bit error high bandwidth medium for B-ISDN.

In appendix B, parameters used in this thesis are derived from common telephony parameters. Call arrival rate, call service rate, packet arrival rate and packet service rate are derived from CCS (Hundred call seconds (per hour)), average call holding time, average talk spurt (packet) length and average pause (inter-packet) length between talk spurts.

A program developed to solve Linear Equations for Markov Model using the Gauss-Seidel method is given in appendix C. In appendix D, programs used to solve the closed form analytical model are presented. Appendix E gives the QNAP2 programs used to solve the queuing model of a multiplexer.

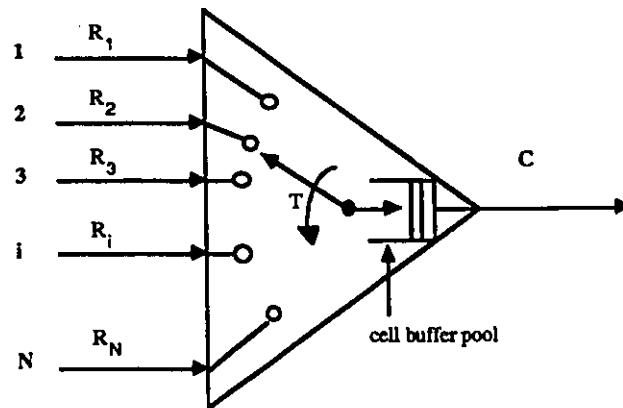


Figure [1.5]: Model of a Multiplexing Node

Chapter 2

B-ISDN

2.1 Introduction

The goal of B-ISDN (Broadband Integrated Services Digital Network) [CC0289, PRYC89] is to provide subscribers with an access to a diverse range of communication services through a single integrated access path. A B-ISDN should have the following characteristics:

- It should be efficient and cost effective.
- It should have good performance. It should either preserve or improve upon existing quality of service.
- It should provide integration of customer access, network resources, customer services and OAM (Operation, Administration and Maintenance) functions.
- It should support current as well as future service needs.
- It should have evolutionary growth capabilities. It should easily be interworked with the existing networks.
- It should be robust to changes in technology.

2.2 Driving Forces Behind B-ISDN

As the name suggests, B-ISDN is really an extension of Narrowband ISDN which has already been defined by CCITT [CC0289]. Over the last few years, it is becoming technologically feasible to implement B-ISDN. This has given a great impetus for the specification of B-ISDN by the CCITT. The following factors have contributed to the conception of B-ISDN:

- the emerging demand for broadband services
- the availability of high speed transmission, switching and signal processing technologies
- the need to integrate circuit and packet mode services; switched as well as non-switched
- the need to provide flexibility to both user and operator
- the need to integrate both interactive and distribution services

- the need for covering the broadband aspects of ISDN

2.3 B-ISDN Service Classes

A classification of B-ISDN service types [MARK89, CHEN89] is given in Figure [2.1]. Two main categories of broadband communication services are identified as interactive and distributive services. Interactive services provide two-way communication between the connected end-points. Distributive services, on the other hand, only provide one-way information with little or no control over the choice of information received by the user.

Interactive services can be conversational, messaging or retrieval services. Conversational services provide a two way dialogue between two connected points. An example of a conversational service is a videophone. Messaging services are not as real-time as conversational services. The messages may be stored, edited and then sent to the receiver. An example of such a service is electronic mail service. In retrieval services, the information is usually stored in large memory banks and it may be accessed by the user on demand. An example is an archive database.

The distributive services can be with or without user individual control. An example of a distributive service without user individual control is radio or television program service. All the subscribers get the same programs at precisely the same time. Users may be given some choice in the type of information they want to access over a broadcast channel by repeating the information broadcast periodically like in full channel broadcast videography.

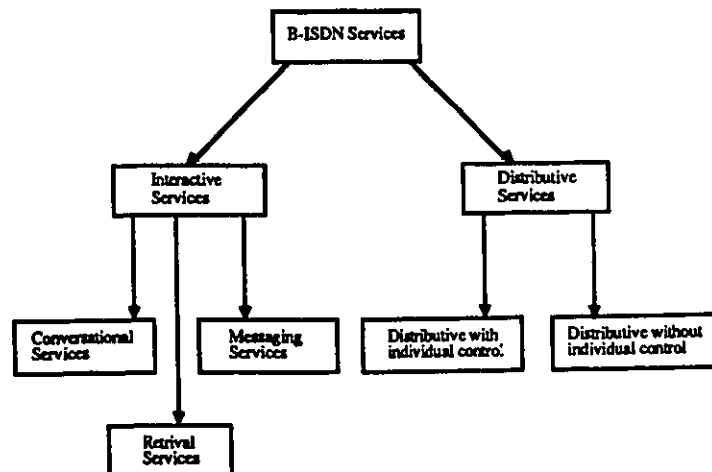


Figure [2.1]: Classification of B-ISDN Services

2.4 B-ISDN service rates

Narrowband ISDN defines the following channel types:

1. B-channel (64 Kbps information)
2. D-channel (16 or 64 Kbps signalling LAPD)
3. E-channel (64 Kbps signalling CCS #7)
4. H₀-channel (384 or 6x64 Kbps circuit switched leased)
5. H₁₁-channel (1536 or 24x64 Kbps semi-permanent leased)
6. H₁₂-channel (1920 or 30x64 Kbps)

ISDN provides access capabilities at Basic Rate Access (BRA) and Primary Rate Access (PRA) as below:

Basic Rate Access: $2B + D_{16}$

Primary Rate Access: $23B + D_{64}$ and $30B + D_{64}$

Primary Rate Access for multiple trunks from Central Office to Customer Premises Equipment :

$$m(nB) + r D_{64} \text{ or } r E_{64}$$

where m is an integer, $n \leq 24$ (or 30) and $r \leq 3$

B-ISDN will support the following broadband channels in addition to those defined for ISDN (as given above):

1. H₂₁-channel

The bit rate of this channel will be 32,768 Kbps

2. H₂₂-channel

The bit rate of this channel will be an integer multiple of 64 Kbps in the range of but less than DS3 channel.

3. H₄-channel

The bit rate of this channel will be an integer multiple of 64 Kbps in the range of 132 to 138 Mbps. The definition of this channel will take into account the definition of 150 Mbps UNI based on ATM. It also takes into consideration the existing and future digital hierarchies so that those channels can be used for B-ISDN purposes during the interim period to the introduction of B-ISDN. This channel may also be used to carry television signals.

The specification of H₂₂ and H₄ channels would be such that:

$$4H_{21} \text{ rate} \leq H_4 \text{ rate}$$

$$3H_{22} \text{ rate} \leq H_4 \text{ rate}$$

B-ISDN may provide access capabilities as below:

1. Communicative (Interactive Services):

$$H_4 + 4 H_1 + 2B + D_{64} \quad \text{Where } H_1 \text{ is either } H_{11} \text{ or } H_{12}$$

2. Distributive Services:

$$4H_4 + 16 H_1 + 2B + D_{64} \quad \text{Where } H_1 \text{ is either } H_{11} \text{ or } H_{12}$$

The channel types and the channelized service rates are defined so that interworking of ATM based B-ISDN with the existing circuit switched networks may be simplified.

2.5 Architectural Model

The main switching and signalling functional components [RIDE89, SPEA89] of B-ISDN for 64 Kbps based and broadband services are shown in Figure [2.2].

When a call is initiated, the Local Connection-Related Functions handle the user/network signalling. The user negotiates with the CRFs for the type of call and resource usage. The Inter-exchange signalling then determines if the resources are available across the network and if the other side is willing to accept the call. The call is established if the other user accepts the call and there are enough resources available for the network to handle the call.

If it is a 64 Kbps based (ISDN) call, narrowband functional entities are responsible for handling the service. The broadband functional entities are responsible for the broadband services.

When the user terminates the call, the CRF handles the resource deallocation and user billing.

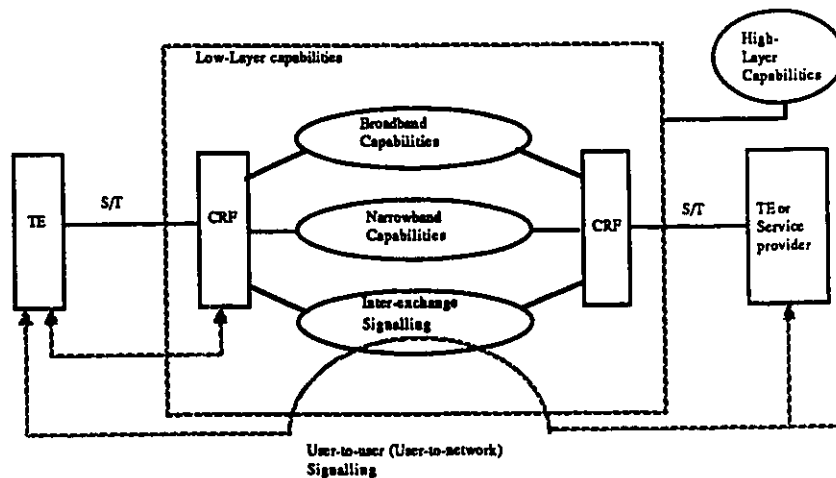


Figure [2.2]: Basic Architectural Model of B-ISDN

2.6 Reference Configurations

The reference configuration as shown in Figure [2.3] is applicable to both ISDN (Narrowband Basic/Primary rate) and B-ISDN [RIDE89]. For basic access, the interface reference points are as follows:

- R Existing interface specification, for example RS-232.
- S/T ISDN terminal interface characterized by four wires, (2B + D) user rate.
- U Under study.

For primary rate access, R and S interface reference points are the same. Other interface reference points are as follows:

- T Four wire. In North America (23B + D + overhead) and in Europe (30B + D + overhead).
- U Primary rate transmission system, T1.

TE1 is defined as an ISDN terminal equipment. TE2 is defined as existing terminal equipment which requires a terminal adaptor (TA) in order to interface with ISDN. NT1 is network terminating equipment that converts a loop transmission (U) to an S/T interface. NT2 is an additional network termination (e.g. PBX) which converts primary rate access to basic rate access.

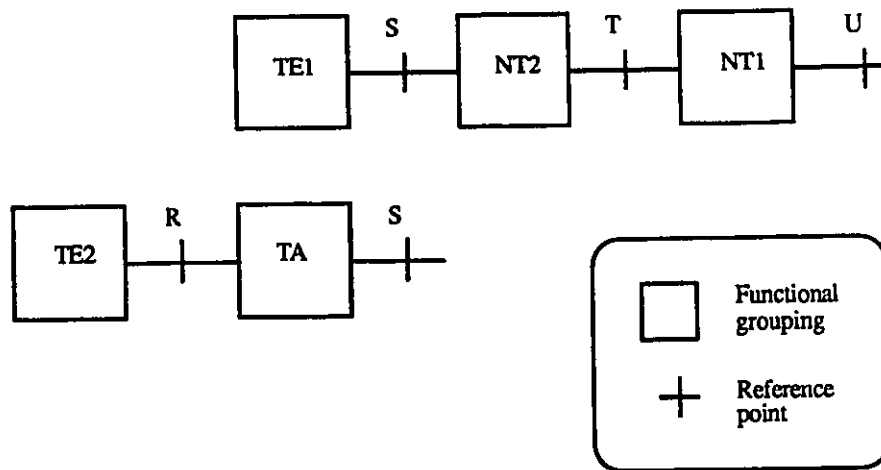


Figure [2.3]: ISDN (B-ISDN) Reference Configuration

Figure [2.4] illustrates combinations of physical interfaces at various reference points. The letter B indicates broadband reference point or functional grouping.

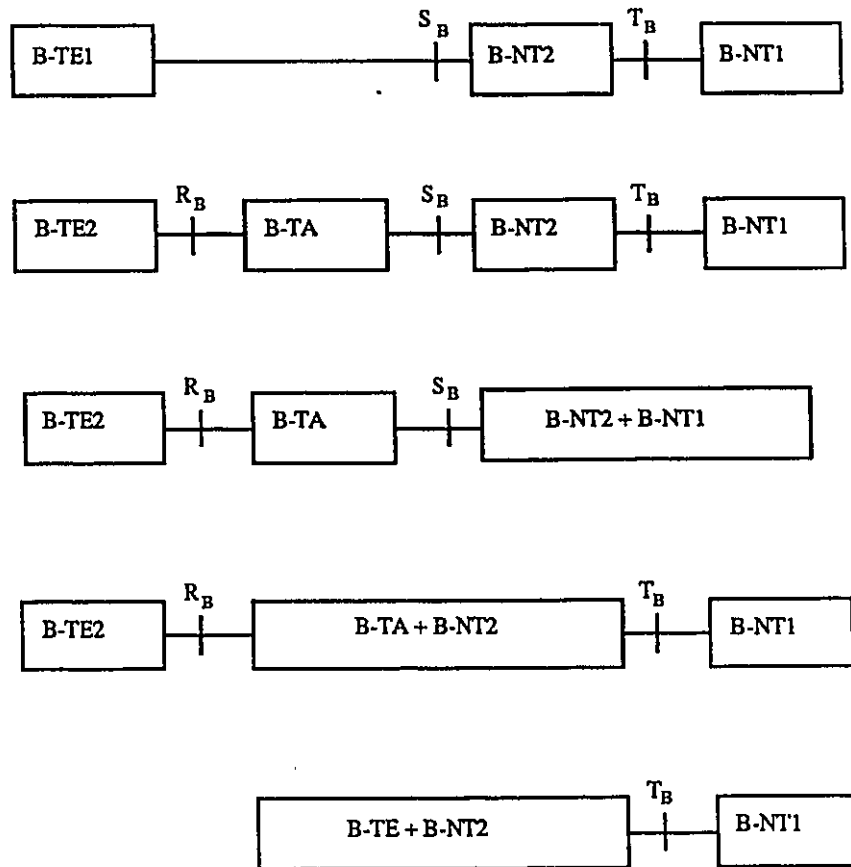


Figure [2.4]: B-ISDN Physical Configuration

2.7 Protocol Model

A layered structure has been recommended for the access protocol to a B-ISDN. The B-ISDN protocol model based on Asynchronous Transfer Mode (ATM), the recommended transfer mode for B-ISDN signals, is shown in Figure [2.5].

The physical media dependent layer could be any existing or future transmission system (e.g. T1 or SONET hierarchy) capable of handling the required service rates.

The ATM layer is designed to offer a flexible and efficient transfer capability common to all broadband services. It consists of a fast packet transfer mode using the Asynchronous Time Division Multiplexing technique. The information is carried in small fixed size information blocks called cells. Each ATM cell has an information field and a cell header. The information field carries the payload data and overhead information of the higher layer protocols. The cell

header contains minimum information to identify cells belonging to the same virtual channel so that it can be transported from its origin to its destination. The cell multiplexing and switching is performed at a very fast rate, much like the multiplexing and switching of fixed time slots in Synchronous Time Division Multiplexing technique. Cells are assigned to sources on demand depending on source activity and resource availability.

ATM is a connection oriented technique. The network resources are allocated on call setup and are released when the call is disconnected. Signalling and user information are carried on different virtual channels. ATM also preserves the cell sequence integrity.

The adaptation layer provides the interface between the ATM layer and the higher protocol layers. At the transmitting end, it maps the information sent by upper layers into ATM cells and at the receiving end it maps the cells received for the same virtual channels into information required by the upper layer protocol. The adaptation layer is protocol (upper layer) dependent which uses the protocol independent ATM layer for information transfer. So the boundary between the ATM layer and the adaptation layer is the boundary between the functions of the ATM cell header and the functions of its information field.

As shown in Figure [2.6], the adaptation layer could be terminated in NTs, network adapter (NA), terminal adaptor (TA), terminal equipment (TE) and exchange terminations (ET).

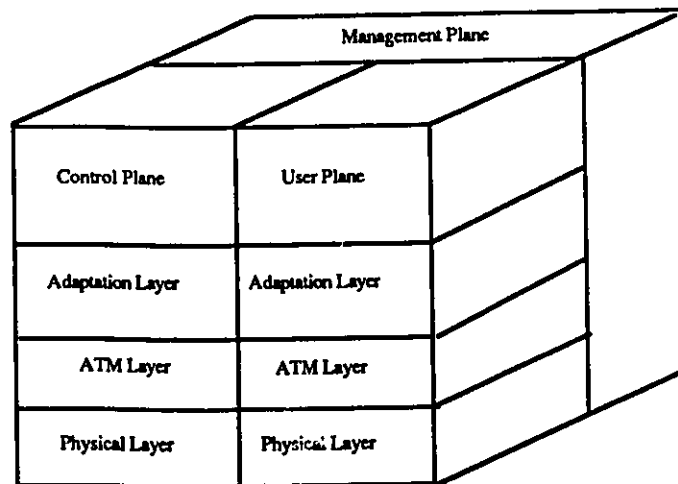


Figure [2.5]: B-ISDN Protocol Stack

2.8 Network Aspects

In broadband ISDN, the signalling and information are carried on separate ATM channels. The signalling message sent by the user to establish a call includes traffic characteristics (e.g. bit rate, burstiness) and the required network transport capabilities (Quality-Of-Service).

The burstiness of a source is defined as the ratio between peak service rate to the average service rate. The Continuous Bit-Oriented traffic sources (synchronous traffic source) have a burstiness of unity where as bursty sources have higher burstiness factors. The burstiness of variable traffic sources can be exploited by asynchronous protocols to utilize the network resources (buffers, bandwidth, etc.) more efficiently.

The quality of service requirement is dependent on the nature of the service itself. Some services require high accuracy of information while others can tolerate information loss to a certain degree. Some services are network delay (and delay fluctuation) sensitive whereas others are not affected by information transfer delays.

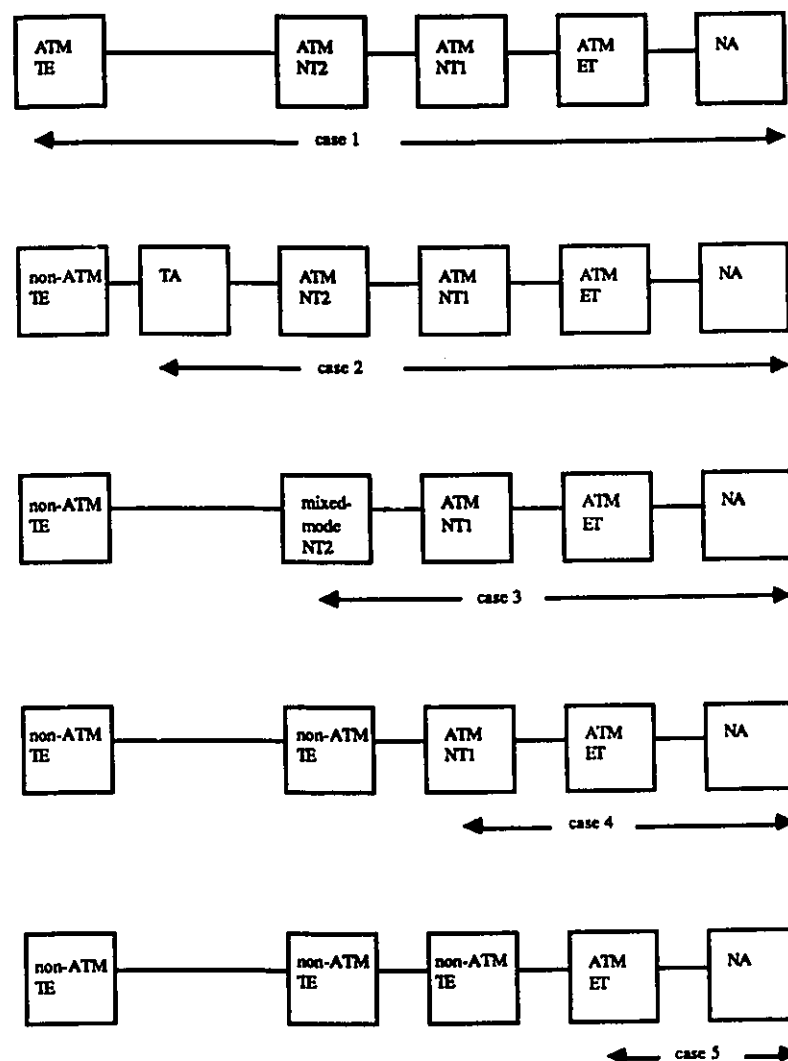


Figure [2.6]: Termination of Adaptation Layer

The network may manage the resources in several ways in response to a request from a source for service connection:

- It may dedicate peak resources to a connection like in STDM. These resources are released only at the termination of the call before they can be assigned to other calls. The advantage of this scheme is that the connected source always has resources available to be used and thus there is no information loss or delay associated with sharing. There is also no delay or delay fluctuation involved because there is no queuing for resources. The disadvantage is that new connections are denied when not enough unassigned resources are available even though the connected sources may not be using them all the time. This scheme is ideal if the sources are all CBO (Continuous Bit-Oriented) but results in inefficient resource usage for bursty sources.
- It may allow resources to be shared among all the connections. This allows for better resource sharing. The disadvantage with this scheme is that there is a chance of loss of information, if the number of sources transmitting simultaneously happens to be larger than the number that the network can handle. One of the schemes to minimize the chance of loss of information is to queue the sources requesting the resources. Unfortunately, this introduces queuing delays in accessing the resources. These delays vary with the number of sources requesting the resources at any instance. Delay variability is more difficult to cope with than the constant delay in communication networks.

This scheme does not give any advantage for CBO sources but it gives better resource efficiency for large number of bursty sources. Resources are provided at a level somewhere between the expected peak and average source bit rate of all the sources connected.

- It may allow resources to be shared among a class of connections. This works on the same principle as the common resource sharing but it can be better tuned to the type of connections in each group.

There are other various schemes derived from the resource sharing schemes given above. A minimum number of resources may be dedicated to each source or group of sources while the rest of sources are shared. This scheme is called Sharing with Minimum Allocation (SMA). A similar scheme is Sharing with Maximum Limit (SML) in which a limit is placed on the maximum number of resources each source or group of sources can take even when other resources are free. These schemes are also suited for bursty sources only.

Once the scheme to allocate the resources is chosen, the network has to monitor the usage by the individual sources (or groups of sources) so that they may not exceed their usage beyond the characteristics agreed at the call setup time. This is very simple if the sources are CBO. It is very challenging when the sources are bursty. It is important to detect the source that is violating

the resource usage in a shared strategy and take necessary action to protect rest of the sources in terms of their quality of service.

2.9 Evolution Towards B-ISDN

The introduction and deployment of B-ISDN may span over decades. It will follow the deployment of narrowband ISDN. The deployment of B-ISDN would likely be very gradual by step wise introduction of functions and services. In the interim period, it will also have to co-exist with and interface with the existing networks and services.

In an interim stage, Network Interwork Units (NIWU) will have to exist to help interface B-ISDN switches with the existing services as shown in Figure [2.7]. A B-ISDN network may utilize various combinations of STDM and ATM networks to support a variety of service characteristics (bit rates, burstiness etc.).

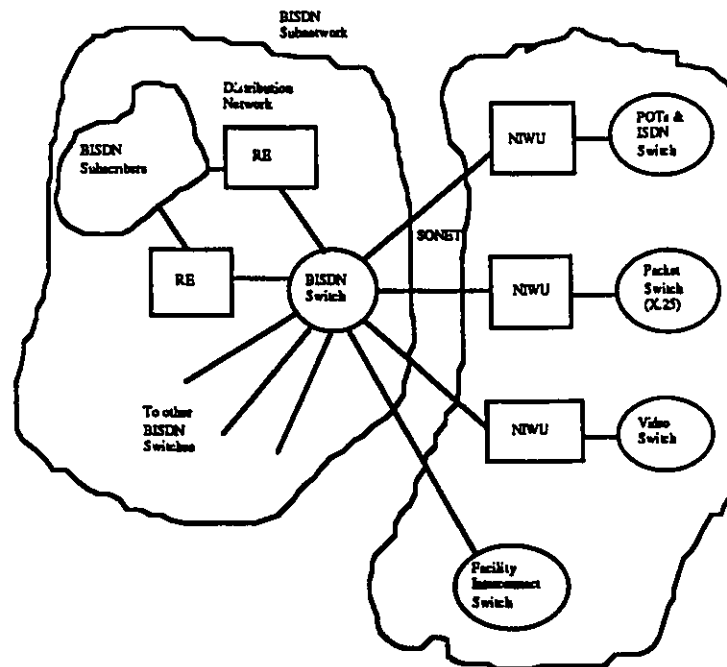


Figure [2.7]: Interworking of B-ISDN with existing Networks

Chapter 3

ATM

3.1 Introduction

Asynchronous Transfer Mode (ATM) is the transfer mode chosen by the CCITT study group XVIII task group on ISDN Broadband Aspects (BBTG) as the basis of B-ISDN [CC0289,COUD89]. As already introduced in chapter 2, ATM is a high-bandwidth low-delay packet-like switching and multiplexing technique. It is targeted to provide high throughput and low delays for B-ISDN signals.

ATM is best described as a compromise between the circuit switching (STDM) and the packet switching. It is intended to achieve time transparency provided by the circuit switching technique while maintaining the bandwidth sharing flexibility of packet switching.

ATM has various similarities with circuit switching. Like circuit switching, ATM is basically a connection-oriented technique. This is not to be confused with the type of services that ATM can carry. ATM does support both connection-oriented and connectionless services. No error or flow control is done on ATM links as is the case with circuit switching. These functions are moved to higher layer protocols at the edges of the network. ATM cells are fixed in length like the STDM time slots. The ATM cells are synchronized to the cell boundaries like STDM slots are synchronized to the frame boundaries. In this respect, the 'asynchronous' part in ATM may not be completely valid. The ATM cell header acts as STDM slot number used for channel identification for the information being carried.

ATM also has some similarities with packet switching. As in the case of a 'packet' in packet switching, each ATM cell consists of a header part and an information part. The cell header carries its routing information. ATM does not preserve the time transparency of the information it carries as is the case with packet switching. The time transparency is lost because of variable cell queuing delays experienced by the cells in the network. This lost time transparency is recovered at the edges of the network by upper layers of protocol.

ATM can support Continuous Bit Oriented (CBO) as well as bursty services. ATM operating in deterministic mode can support real-time CBO services by circuit mode emulation provided sufficient resources are available. The minimum end-to-end delay and loss requirements must be met for CBO services.

ATM may also be able to operate in a statistical mode to concentrate bursty data. Bursty traffic may be assigned lower priority so that it does not adversely affect the CBO performance. ATM may be more efficient for bursty traffic as compared to CBO traffic.

3.2 The ATM Cell

An ATM cell consists of a header field and an information field (Figure [3.1]). The function of the cell header is to route the information field from its origin to its destination. Different cell header and cell information field sizes have been proposed in past by different groups around the world (Figure [3.1]) [GONE87]. An ATM cell, as defined by CCITT, is 53 bytes in length out of which there is a five byte header and the rest 48 bytes constitute the cell payload.

3.2.1 The Cell Header

In theory, only a label (Virtual Channel Identifier and Virtual Path Identifier) is required within the cell header to perform switching and multiplexing functions. However, other fields are needed for performance enhancement. The ATM cell headers proposed by ANSI and ETSI (European Telecommunications Standards Institute) are given in Figure [3.2a] and Figure [3.2b] respectively. The cell headers being proposed by the CCITT for UNI and NNI (Network-Network Interface) are given in Figure [3.2c] and Figure [3.2d] respectively.

ANSI favoured an 8 bit Access Control Field (ACF) and a 20 bit Virtual Channel Identifier (VCI) inside the cell header. The ACF is used to control shared medium support. The ETSI position is that the medium access should not be standardized at this level. ETSI proposed a 24 bit VCI/VPI (Virtual Path Identifier) field. The VPI is meant to route groups of Virtual Channels between two end points within a network. It provides a cross-connection function on the basis of groups of channels. Since ANSI favours ATM within a SONET frame structure, it supports cross-connection on a Virtual Tributary (VT) or STS-1 basis. CCITT has defined a 4 bit Generic Control Field at UNI to assist customer premises solutions in controlling flow of traffic. In addition, a 24 bit VCI/VPI field is also defined. A maximum of 12 bits may be used for VPI and a maximum of 12 bits may be used for VCI at UNI. At NNI (Network-Network Interface), a 12 bit VPI and a 16 bit VCI is defined.

ANSI supports a 2 bit payload type field and ETSI has proposed a 1 bit payload type field in the cell header. CCITT recommended a 2 bit field for payload type. A 2 bit priority field is also included in ANSI's proposal. According to ETSI position, priority should be defined at the call setup time and stored within the switch as a parameter for the virtual channel. No cell priority field has been defined within the cell header by CCITT.

ANSI proposed an 8 bit Header Checksum (HCS) field. ETSI proposal contains a 5 bit HCS which is enough to correct 1 bit and to detect burst errors in 24 bit VCI/VPI field. CCITT has defined an 8 bit Header Error Check field. The polynomial proposed for the HEC is $x^8 + x^2 + x + 1$. The scheme to detect/correct errors and the operation of receiver using this scheme are given in Figure [3.3a] and Figure [3.3b] respectively.

3.2.2 The Cell Information field

The effect of the delay incurred in collecting a full cell of voice samples has been a major consideration in deciding the cell size. ETSI proposed a 32 byte information field which introduces a 4 msec delay in a 125 μ sec based voice protocol. This additional delay could be compromised by reducing the distance for installation of echo cancellers. A longer cell information field gives better throughput efficiency. It also gives more time to the switch for processing a cell header. ANSI favoured a 64 byte information field. Since echo cancellers are already installed in North America, higher delay due to cell assembly/disassembly may be tolerated. A 48 byte cell information field has been defined by CCITT. Out of these 48 bytes, 4 bytes are used for ATM Adaptation Layer (AAL) header (Figure [3.4]).

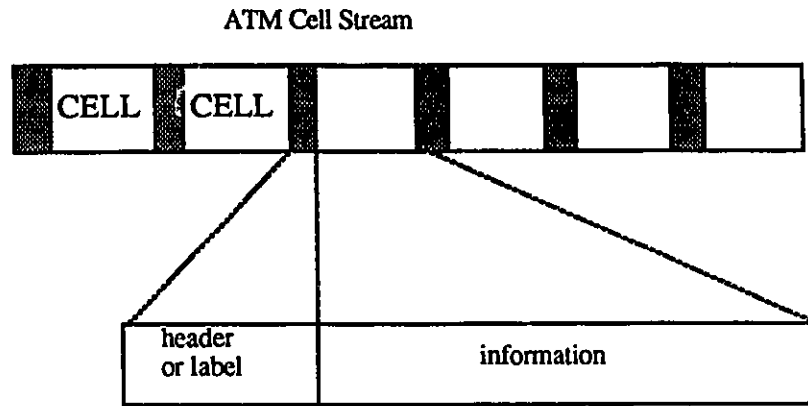
Within the AAL header, a 2 bit cell assembly/disassembly field has been defined. A 14 bit message identifier has also been defined by CCITT. This message identifier is used for cell sequencing in order to detect lost cells and to restore time transparency. There is a 6 bit partially filled cell identifier. Some cells are bound to be partially filled if the message (data) being segmented is not an even multiple of the cell information field size. Partially filled cells are also required when the service rate is too slow. The remaining 10 bits in the AAL header are used for CRC (Cyclic Redundancy Check) for the cell information to detect cell transmission errors for connectionless services.

3.3 Virtual Path Capabilities

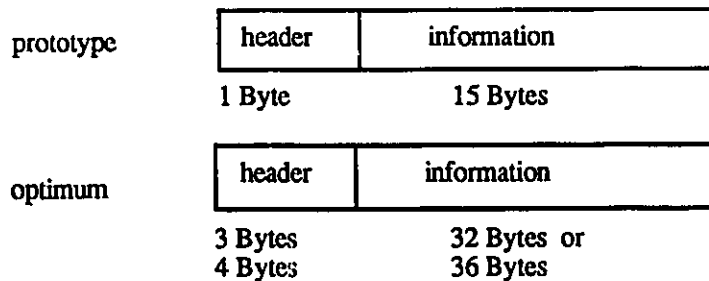
A virtual path is a common route between two end points in a network to be followed by all the ATM virtual channel within the same virtual path. It gives the capabilities to cross-connect virtual channels on the basis of a group of channels rather than on individual channel basis. Following recommendations are included in the CCITT report (19-30 June, 1989) [CC0189] regarding Quality of Service (QOS) requirements within a VP:

- A VP provides a single QOS for all VC's within it.
- VP's may carry channels with a range of QOS. The VP should guarantee the most stringent QOS requirement within the group.

- All channels within the VP should be able to tolerate the QOS as a result of asynchronous multiplexing of services.
- If the range of QOS required between two end points is broad, more than one VP's may be defined so as to cover the full range of QOS.



PRELUDE (CNET France)



CSELT (Italy)

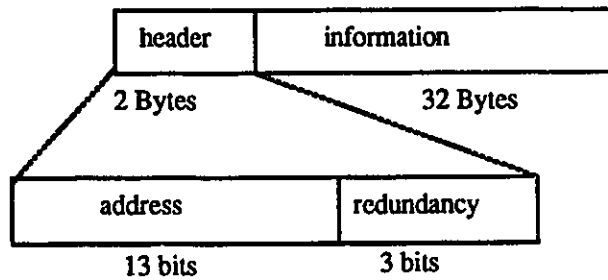


Figure [3.1]: ATM Cell

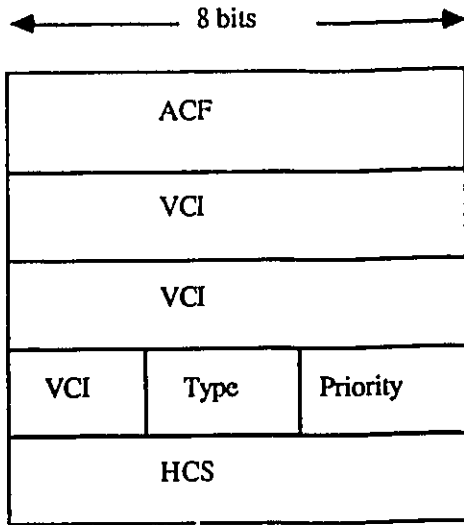


Figure [3.2a]:
ANSI ATM Cell Header

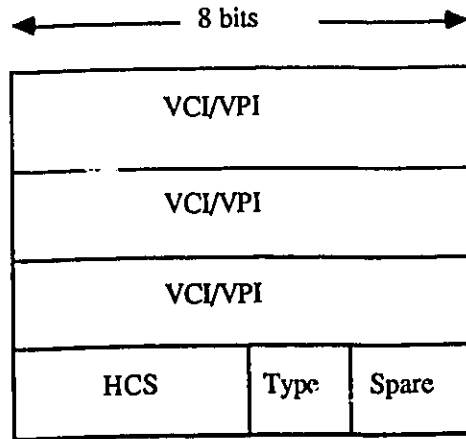


Figure [3.2b]:
ETSI ATM Cell Header

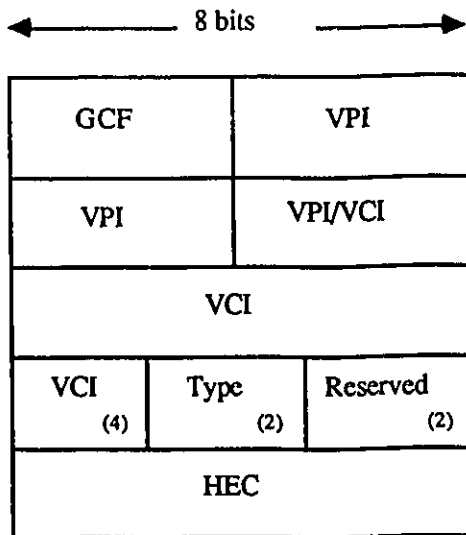


Figure [3.2c]:
CCITT ATM Cell Header for UNI

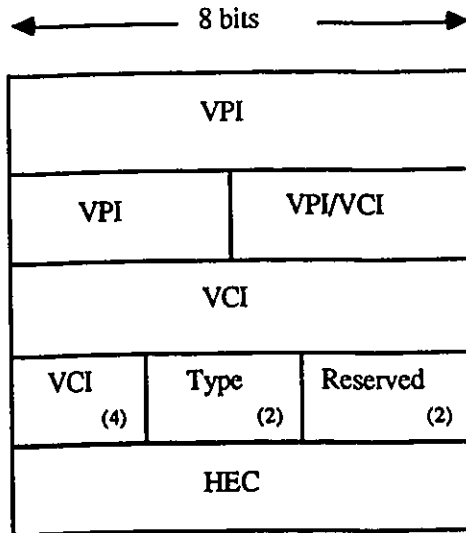


Figure [3.2d]:
CCITT ATM Cell Header for NNI

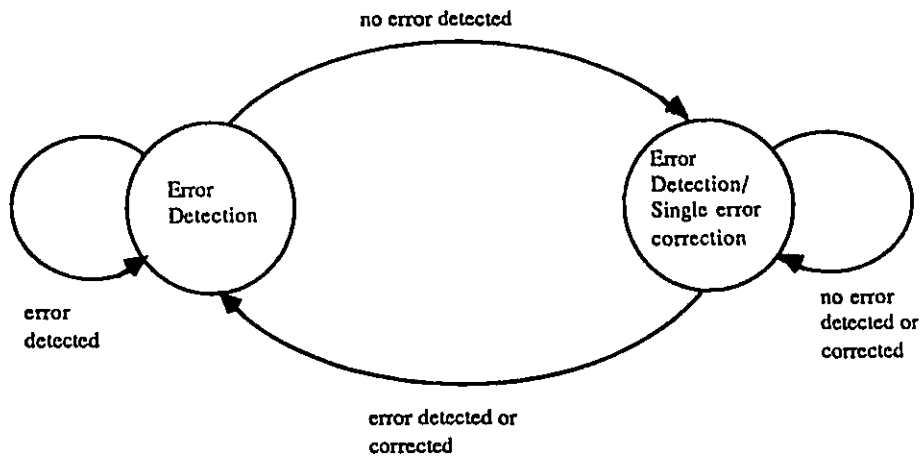


Figure [3.3a]: Cell Error Detection/Correction Scheme (From [CC0189])

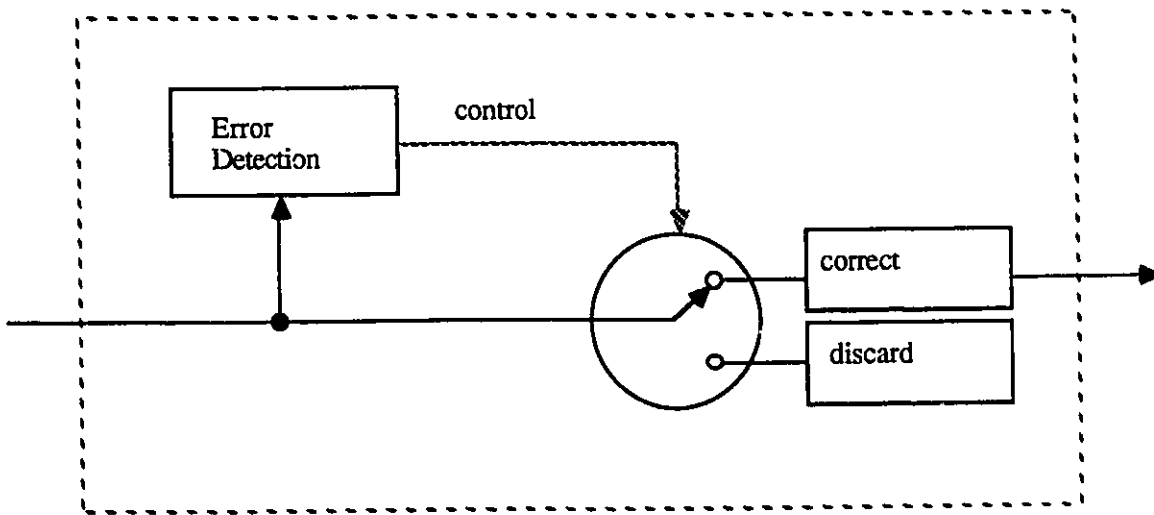


Figure [3.3b]: Operation of an ATM Cell Receiver

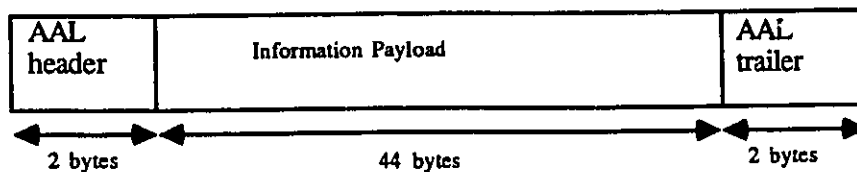


Figure [3.4]: ATM Cell Information Field (Payload and ATM Adaption Layer Overhead).

3.4 Cell synchronization

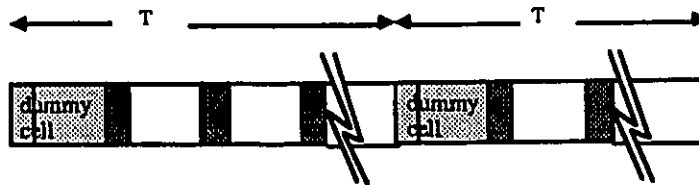
Figure [3.5] shows different possible schemes for ATM cell synchronization. They are:

- Self structured ATM: In this scheme, dummy synchronizing cells are transmitted in the cell stream. The synchronizing cells may be transmitted periodically or on a random basis (Figure [3.5a]).
- Self structured ATM within an external synchronous frame (e.g. SONET): The external frame is only used for synchronous transport. The maintenance overhead (Path Overhead of SONET) may or may not be used for ATM purpose (Figure [3.5b]).
- ATM within an external synchronous frame: The external frame provides the cell synchronization (Figure [3.5c]).

ANSI favoured an interface for B-ISDN that carries ATM cells within SONET payload capacity, with SONET providing the cell synchronization. ETSI, on the other hand, proposed a self structured ATM interface. CCITT has defined two UNIs for B-ISDN, one on the lines of ANSI's proposal and other on the lines of ETSI's proposal. In order to facilitate interworking between the two interfaces, either should support 155.52 Mbps (125 μ sec) signal.



Self Structured ATM with Random Synch Cells



Self Structured ATM with Frame of duration T

Figure [3.5a]: Self Structured ATM Cells

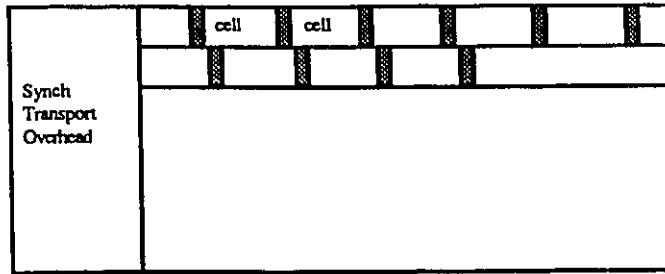


Figure [3.5b]: Self Structured ATM Cell Stream within an External Synchronous Frame

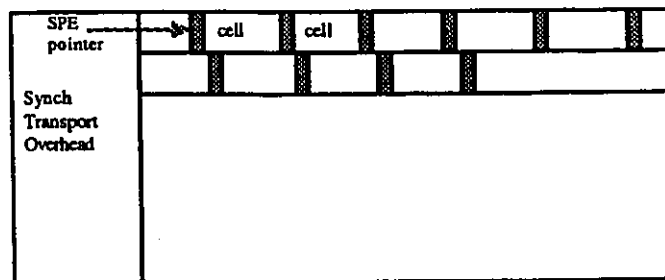


Figure [3.5c]: ATM Cell Stream Synchronized within an External Synchronous Frame

3.5 B-ISDN User Network Interface

As described in section [3.4], two interfaces for **150 Mbps UNI** are defined by CCITT. One interface (initially proposed by ANSI) defines ATM cells carried within the payload capacity of SONET. A SONET frame provides the cell synchronization to the ATM cell stream (Figure [3.5c]). The other interface is defined on the basis of self structured ATM cells (initially proposed by ETSI) (Figure [3.5a] and Figure [3.5b]). In either of the interfaces, the whole capacity is dedicated to transport of ATM. Both of these interfaces are also being considered for 600 Mbps UNI (only the bandwidth is higher).

For **600 Mbps UNI**, hybrid schemes to share the payload capacity with other non-ATM traffic are also being considered in addition to those defined for the 150 Mbps interface. As shown in Figure [3.6], the 600 Mbps bandwidth is divided into four sections of 150 Mbps each. Either all the sections may be used by ATM traffic or some of the sections may be used to carry other types of traffic (e.g. STDM).

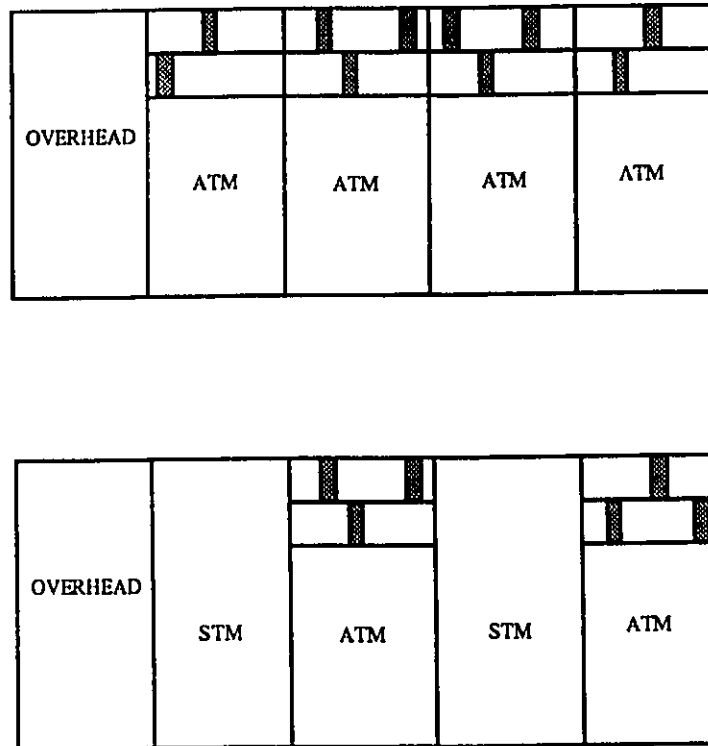


Figure [3.6]: Structure of 600 Mbps UNI

3.6 ATM based Protocol Model for B-ISDN

Figure [2.5] shows the protocol stack being defined for ATM based B-ISDN access. The physical layer should be a high throughput synchronous transport scheme (e.g. SONET). The most important layers, for the purpose of B-ISDN, are the ATM layer and the Adaptation layer.

ATM Layer

This layer provides transfer capabilities to cells from all current and future services. It handles the cell header functions and transports the information field as directed in the header part of the ATM cell. It provides multiplexing and switching functions for the ATM cells received from the adaptation layer.

Adaptation Layer

This layer maps the non-ATM services into ATM cells. User information as well as control messages sent by the higher level protocols are converted into cells and sent to the ATM layer. The Adaptation layer provides the following functions:

Continuous Bit Stream Oriented service adaptation functions:

- assembly/disassembly of information and control signals into cells
- compensation for variable delays
- action on lost cells
- clock recovery

Packet mode service adaptation functions:

- cell assembly and disassembly
- handling partially filled cells
- rate adaption
- action on lost cells

Figure [3.7] shows ATM protocol stacks for User Plane (UP) and the Control Plane (CP) used in an ATM network. For User Plane, the full stack is used at the User Network Interface but the intermediate nodes provide only cell relaying function and thus use only the ATM layer. For the Control Plane, full protocol is used at all the network nodes because the resource allocation is negotiated from all the network nodes along the way, when a call is setup or released.

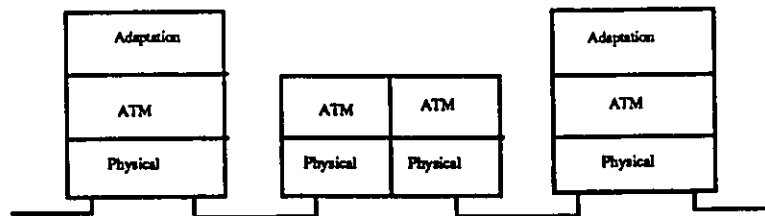


Figure (3.7a): User Plane (Flow of Data)

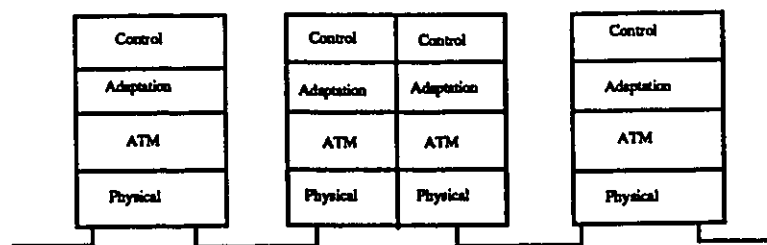


Figure (3.7b): Control Plane (Flow of Signalling)

3.7 Performance Measures For ATM

Even though ATM provides almost ideal transport mechanism for B-ISDN, it also poses the following challenges, to the ATM network designers, brought about by asynchronous multiplexing of cells.

1. Cell loss

Cell loss in the ATM network results due to:

- cell buffer overflow in the switch:

Cell buffer overflows when the network node does not have enough cell buffers to store all the incoming cells.

- cell buffer underflow:

Cell buffer underflow is caused when the cell arrives too late at the destination due to queuing within the network. The cell is usually replaced by a dummy cell. The real cell is discarded on arrival.

- bit error rate on the transmission line and within the network nodes.

2. Cell delay

Cell delay is caused by:

- queuing of cells at the network nodes.
- cell transmission delay.
- cell assembly and disassembly delay.

3. Cell delay variation

Delay variation is caused by stochastic nature of the queuing delays. It is also caused by cell loss. When a cell is lost, a dummy cell is introduced into the network after a maximum waiting time.

The appropriate performance measures for ATM [WOOD89] are the cell loss and the delay variation in cell arrival at the destination node across the network. The cross network cell delay variation can be minimized by discarding overflow cells at the input of the node and thus avoiding queuing of cells within the node. The cell discarding introduces some cell loss but simplifies the control to a great extent which is important for nodes operating at B-ISDN rates. The maximum cell delay variation must be known between two end-points in the network so that a smoothing delay [CHEN89] may be added when the first cell is received at the destination

node. The cell loss probability requirement is dependent on the application. The cell loss probability due to resource sharing may be reduced by reducing the number of input sources to the node. The cell loss rate due to resource sharing should only be reduced to the level of the other information loss rates in the network.

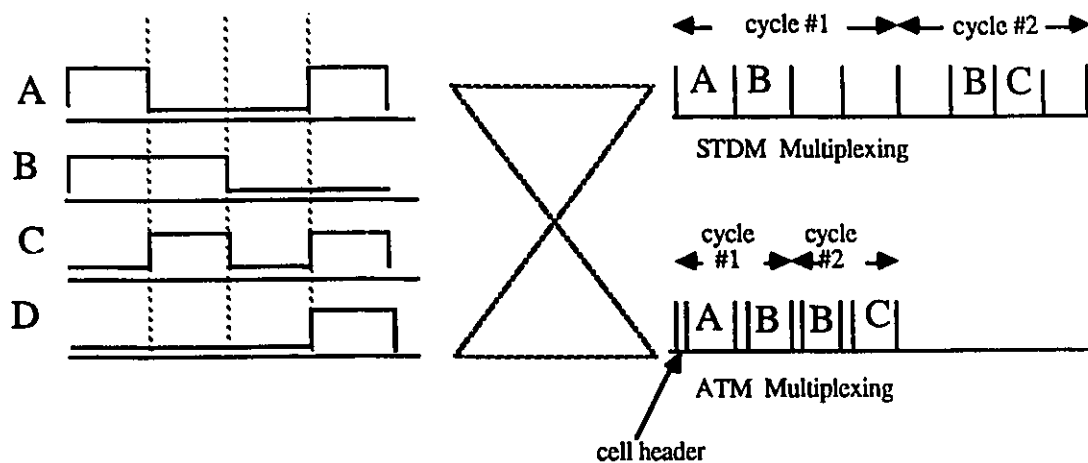


Figure [3.8]: Synchronous and Asynchronous Multiplexing

3.8 ATM Congestion Control

ATM is best suited for multiplexing bursty sources. It asynchronously multiplexes the bursty input sources to achieve better resource (bandwidth) utilization than would otherwise be possible by assignment of peak resource requirements in circuit switching (Figure [3.8]).

A bursty source can be fully characterized by its peak rate, its average burst size and its Burstiness Factor (BF). The burstiness factor is defined as the ratio between the peak service rate to its average service rate. As long as the sum of resource requirement of all the input sources in a node is less than the resources available, no information (cell) loss is experienced. Due to the stochastic nature of the bursty sources, it is possible that all (or most of) the sources start transmitting simultaneously and thus causing congestion. Congestion results in information (cell) loss. It also causes cell queuing delays. Although congestion may be avoided by allocating resources according to peak requirement of all input sources, this defeats the purpose of asynchronous multiplexing. Better resource utilization is achieved in asynchronous multiplexing at the expense of some information loss.

The possible congestion control approaches [WOOD88, RICH88] may be classified into two categories. In reactive control, the network nodes react as they detect the possibility of

congestion by informing the control to take the necessary action. In preventive control, the input sources are prevented from violating their agreed upon service characteristics when the call is established.

The reactive congestion control may be hard to implement for ATM. The time taken by the control message to travel to the network control may be significant. Since the network operates at such high speeds, the congestion may have gone away by the time the control is ready to react. Moreover, the speed with which the congestion would set in in such network will make it impossible to get a pre-warning.

The preventive congestion controls are more easily implemented although not as attractive as reactive ones in terms best resource sharing achieved. One of such schemes is given in Figure [3.9]. It detects the peak and average cell transmission by ATM sources and takes preventive action at the input so as to reduce the probability of congestion within the network.

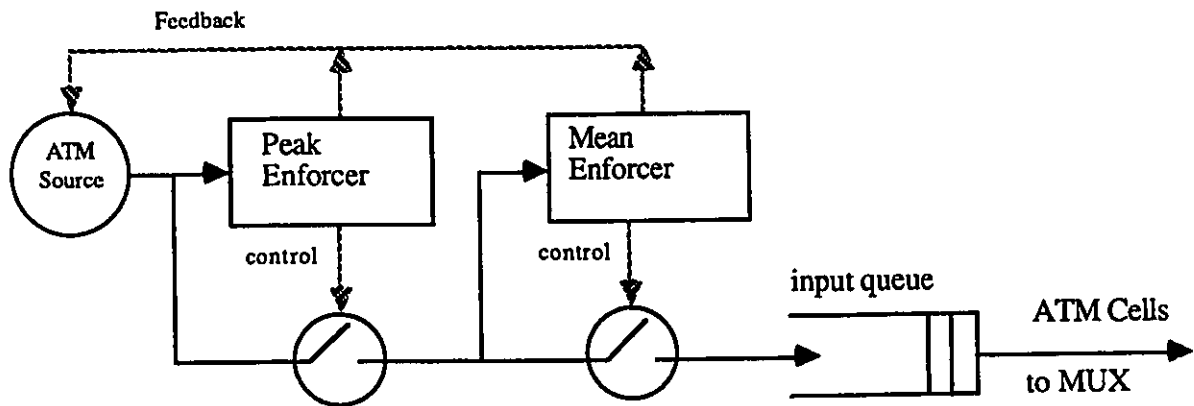


Figure [3.9]: ATM Cell Source Preventive Control at the input of the Multiplexer

Chapter 4

Model of an ATM Multiplexer

4.1 Traffic Model of a Bursty ATM Source

An ATM traffic source generates a sequence of ATM cells. In a bursty source, these cells are generated more frequently during the bursts. The bursts are separated by inter-burst silent periods when no cells are produced. The bursts of cells are produced as a result of messages or packets (referred to as packets from here on) sent by the upper layers (above adaptation layer) of the protocol at UNI (User-Network Interface). When modelling a single bursty ATM traffic source, we may look at it at three different levels [MARK89] as shown in Figure [4.1].

1. Call Setup Level

At the call setup level, the source is either connected or not-connected at a given time. In telephony terms, connected refers to the off-hook period and not-connected refers to the on-hook period. From here on, we will use the terms **connected** and **not-connected**. The source only transmits information packets during the connected period. In figure [4.1], the connected period A and the not-connected period B are assumed to be independent identically distributed (iid) random variables which follow a geometric distribution. This assumption is valid for voice calls. It is also applicable to other applications (video [KRAI87]).

$$P_C = \text{Prob (source is connected)} = \frac{\bar{A}}{\bar{A} + \bar{B}} \quad (4.1)$$

Where \bar{A} and \bar{B} are the average values of variables A and B respectively.

A two state Markov Chain for a source is shown in Figure [4.2a]. State-0 refers to the not-connected state and state-1 refers to the connected state. If the transition rate from state-0 to state-1 is α and the transition rate from state-1 to state-0 is β , then (from [KLEI75]):

$$P_C = \text{Prob (source is connected)} = \frac{\alpha}{\alpha + \beta} \quad (4.2)$$

2. Packet Level

A packet is generated when a connected source has a message to be transmitted. At packet level also, there are busy and silent periods. In figure [4.1], packet transmission period p and inter-

packet period q are both random variables. For voice communication, both these variables can be assumed to be geometrically distributed. In a voice telephone call, the packets refer to talk spurts and inter-packet silent periods refer to pauses in human voice. For compressed video communication, the frame length and the inter-frame periods may also be assumed to be geometrically distributed [KRAI87].

Here we refer to a source as **transmitting**, if it is sending a packet into the MUX, otherwise it is **not-transmitting**. Given that a source is connected,

$$P_T = \text{Prob (source is transmitting)} = \frac{\bar{p}}{\bar{p} + \bar{q}} \quad (4.3)$$

Where \bar{p} and \bar{q} are the average values of variables p and q respectively.

A two state Markov chain for a packet is shown in Figure [4.2b]. State-0 in this figure refers to not-transmitting source and state-1 refers to transmitting source. The transition rate from state-0 to state-1 is λ and transition rate from state-1 to state-0 is μ . Again

$$P_T = \text{Prob (source is transmitting)} = \frac{\lambda}{\lambda + \mu} \quad (4.4)$$

This probability is also related to the **Burstiness Factor** of the source. Burstiness Factor is defined as the ratio between the peak source rate to the average source rate. If R is the peak source rate and r is the average source rate, then

$$\begin{aligned} BF &= \frac{R}{r} \\ \text{Substituting } r &= R \frac{\bar{p}}{\bar{p} + \bar{q}}, \\ \text{we have} \quad BF &= \frac{\bar{p} + \bar{q}}{\bar{p}} = \frac{1}{P_T} = \frac{\lambda + \mu}{\lambda} \end{aligned} \quad (4.5)$$

Continuous Bit-Oriented traffic has a burstiness factor of 1, whereas very bursty sources have higher ($\gg 1$) burstiness factors. Asynchronous multiplexing exploits the burstiness of sources being multiplexed. It achieves better high speed trunk utilization by multiplexing more sources on the trunk than the Synchronous Time Division Multiplexing does.

3. Cell Level

When a source is transmitting a packet, it generates a sequence of fixed length (L bits) ATM cells as shown in Figure [4.1]. Cells coming from a source into the multiplexer, appear periodically

on the output trunk of the multiplexer as will be explained in section [4.2]. For a given cell length of L bits, the cell period $T (= L/R)$ and the cell duration $\Delta (= L/C)$ are determined by the source peak rate R and the high speed output trunk rate C .

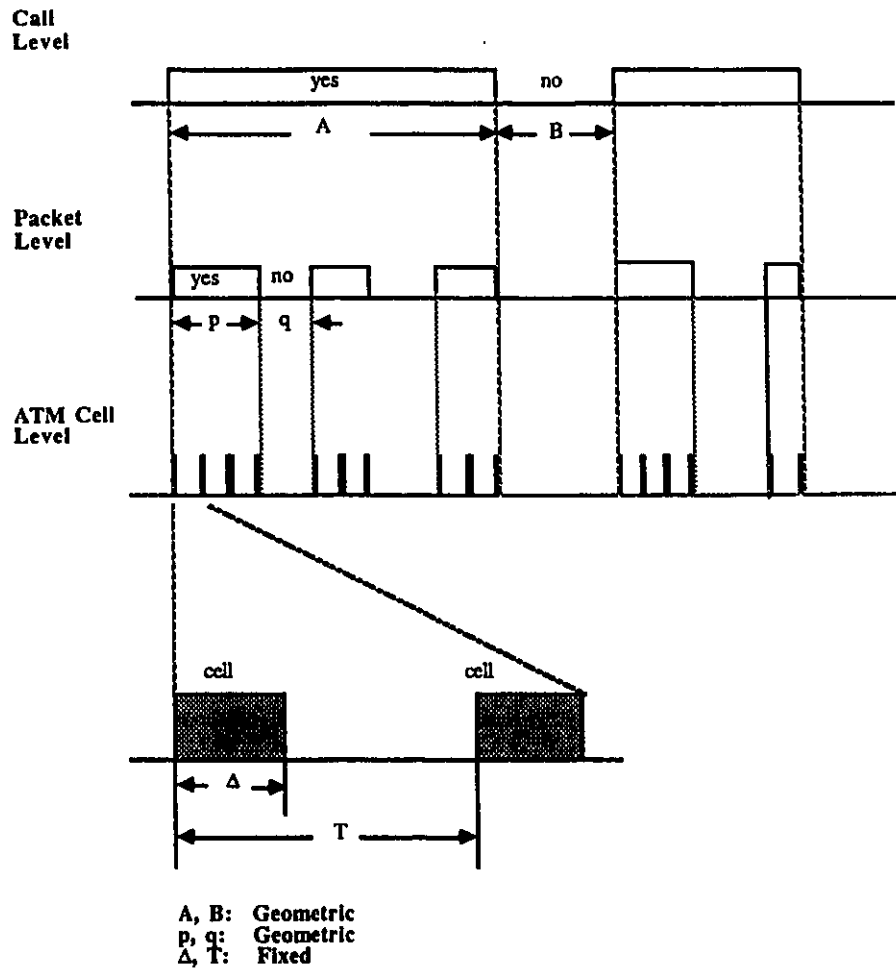


Figure [4.1]: Model of a Bursty ATM Traffic Source

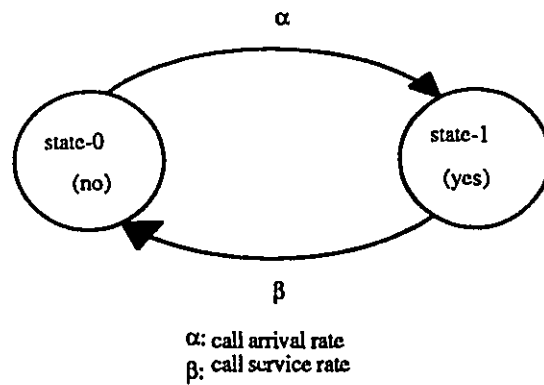


Figure [4.2a]: Call Level Markov Chain

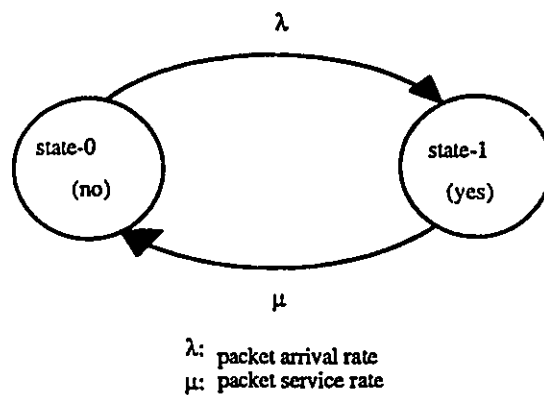


Figure [4.2b]. Packet Level Markov Chain

4.2 Queuing Model of an ATM Multiplexer

Figure [4.3] shows a simple model of an ATM multiplexer. N inputs with identical traffic characteristics (peak rate R , burstiness factor BF and average packet length p) are connected to the input of the MUX. The multiplexer employs shared cell buffering at the output. It collects cells from each input periodically and then transmits the cells on the output trunk. If L is the length of each cell in bits, each input (when transmitting) contributes one cell every L/R seconds where R is peak source rate in bits/second. This accounts for the periodic nature of cells from each input as they are transmitted on the high speed trunk. Each cell occupies $\Delta = L/C$ seconds on the output trunk where C is the trunk speed in bits per second. As all the input sources are assumed to be identical, it means that if $M = C/R = T/\Delta$ cell buffers are provided in the MUX, it will not lose any cells as long as the number of simultaneously transmitting input sources is less than or equal to M and the multiplexer samples inputs every T seconds.

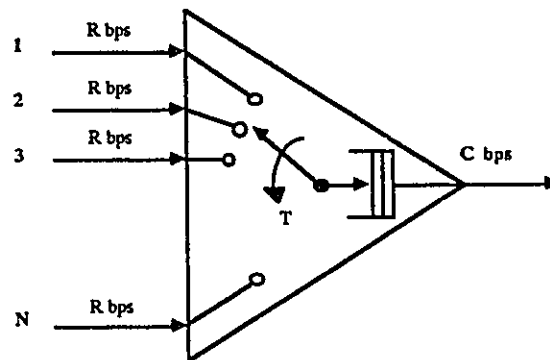
Each transmitting input generates a cell every T seconds with a probability of unity irrespective of the reference of synchronization of the inputs because the cell generation is deterministic in nature when the source is transmitting. If M or less input sources are transmitting simultaneously, they will contribute as many cells into the MUX in T seconds. Since the multiplexer has M buffers and the outgoing trunk can accommodate M cells in T seconds, no cell will be lost. But if more than M inputs lines were transmitting simultaneously, any additional cells (on top of M) will be lost.

$$\begin{aligned}
 P_{\text{loss}} &= \text{Prob (cell loss)} \\
 &= \text{Prob (more than } M \text{ input sources are transmitting)}
 \end{aligned}
 \tag{4.6}$$

It should be noted that as long as the multiplexer scan period is equal to or faster than the cell generation period of each source and the multiplexer buffers only the buffers that it can transmit out in one scan period, this model remains applicable and the results will be similar.

In asynchronous multiplexing, we try to multiplex as many sources as possible on the high speed trunk to achieve better bandwidth utilization. Unfortunately, the probability of cell loss increases with the increase in number of inputs N . Conversely, we can reduce the cell loss to a very small value by reducing the number of inputs but this defeats the purpose (of asynchronous multiplexing). The best compromise is to reduce the cell loss only to a value comparable to the cell loss due to other factors in the network (like bit-error during transmission). To reduce the cell loss any further will not contribute any gain to the overall cell loss probability.

In the above scenario, where only M cell buffers are provided in the MUX, the cell delay in the multiplexer is negligible since no queuing occurs. Only transmission delays are there. So the delay performance is as good as in **STDM** (Synchronous Time Division Multiplexing technique) and there is no delay fluctuation.



C: Output link rate
R: Input link rate
N: Number of inputs
 All inputs have same traffic characteristics:
 - Rate, Burstiness, Peak rate, Average rate

Figure [4.3]: Model Of An ATM Multiplexer

4.3 Solution

4.3.1 Markov Chain model

An inputs of the multiplexer may be connected or not-connected at a given time. When an input source is connected, it may be transmitting or silent (not-transmitting). The Markov chains given in Figure [4.2] can be expanded for N sources to generate S(i,j), a two dimensional state space given in Figure [4.4]. In this two dimensional chain, i refers to the number of sources connected simultaneously and j refers to the number of sources actually transmitting packets at a time. α , β , λ and μ have the same meaning as described in previous sections. When the system is in state (i,j), it can make following transitions (Figure [4.4b]):

1. It goes to state (i + 1, j) with a transition rate of (N - i) α because there are (N - i) sources that are not connected. A call may arrive on any of these lines with a rate α .
2. It goes to state (i - 1, j) with a transition rate of i β because any of the i connected sources servicing calls may finish service with a rate β .
3. It goes to state (i, j + 1) with a rate (i - j) λ because out of i sources that are connected and can transmit packets, j are already transmitting. There are (i - j) connected sources that may potentially start transmitting packet with a rate λ .
4. It goes to state (i, j - 1) with a rate j μ because any of the j sources transmitting packets may finish packet transmission with a rate μ .

This Markov chain gives the following equations [KLEI75]:

$$P_{i,j} = \frac{P_{i-1,j}(N-i+1)\alpha + P_{i,j-1}(i-j+1)\lambda + P_{i,j+1}(j+1)\mu + P_{i+1,j}(i+1)\beta}{i\beta + j\mu + (i-j)\lambda + (N-i)\alpha} \quad (4.7)$$

and the normalization equation:

$$\sum_{j=0}^N \sum_{i=j}^N P_{i,j} = 1 \quad \text{Where } i \geq j \quad (4.8)$$

The above equations have been solved using the Gauss-Seidel method [YOUN71] for solving large linear equations. A brief description of the Gauss-Seidel method and the program designed to solve it are given in Appendix C.

The probability that j sources are transmitting simultaneously is given by:

$$P_j = \sum_{i=j}^N P_{i,j} \quad (4.9)$$

The values of α , β , λ and μ are calculated as follows:

$$\alpha = \text{CCS}/(\text{call_holding_time}*(36\text{-CCS})) \quad \text{calls/second} \quad (4.10.1)$$

$$\beta = 1/\text{call_holding_time} \quad \text{seconds/call} \quad (4.10.2)$$

$$\lambda = 1/\text{interpacket_length} \quad \text{packets/second} \quad (4.10.3)$$

$$\mu = 1/\text{packet_length} \quad \text{seconds/packet} \quad (4.10.4)$$

Where CCS is number of 100 call seconds (per hour). The definitions of terms and derivation of the above relations is given in Appendix B.

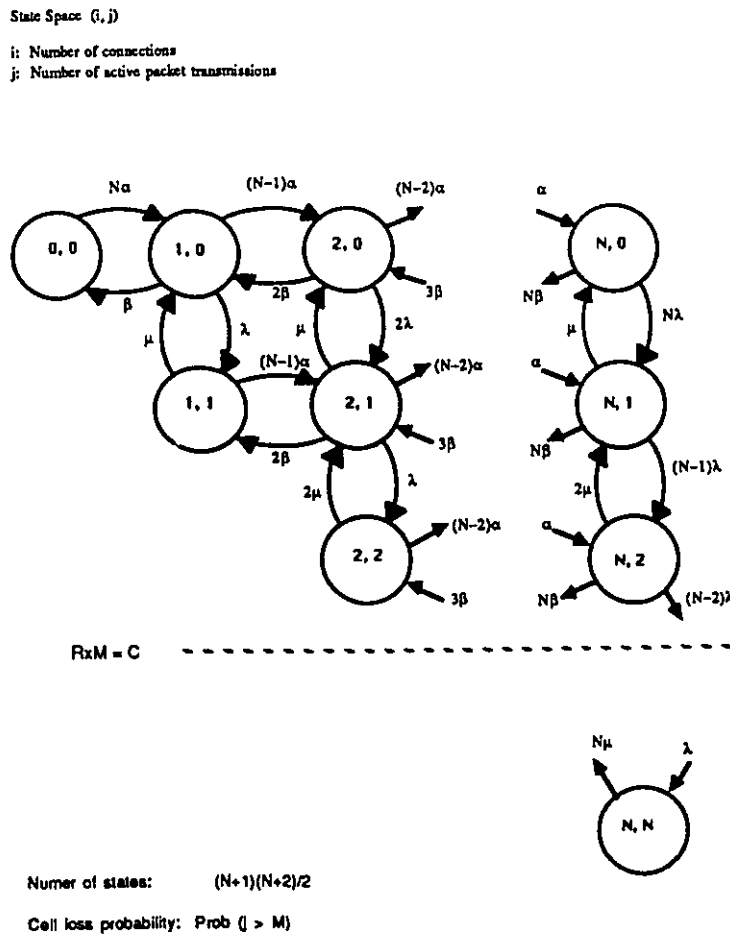


Figure (4.4a): Markov Chain For Multiplexer with N inputs.

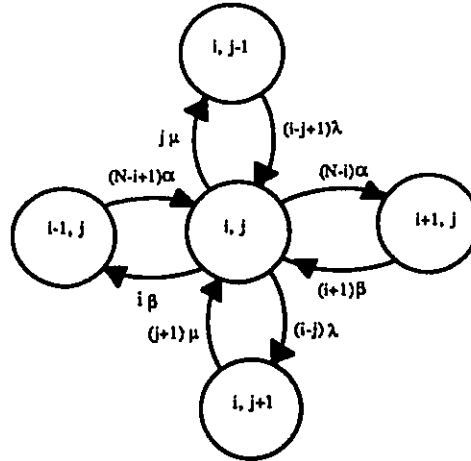


Figure [4.4b]: Transitions from state i, j in the Multiplexer Markov Chain

4.3.2 Petri Net Model

An equivalent Petri Net model [MARS86] for the multiplexer is shown in Figure [4.5]. Place-1 represents the not-connected sources. Initially N tokens are placed in this place to represent N inputs to the multiplexer. Place-2 represents the connected sources. Initially, no tokens are placed in this place. As the calls arrive on the inputs of multiplexer, the tokens move from place-1 to place-2 through transition-1 with a firing rate of $(N - i)\alpha$, where i is the number of tokens in place-2 or the number of sources already connected at that time and α is the call arrival rate for each input. The tokens move from place-2 to place-1 through transition-2 with a firing rate of $i\beta$, where β is the call service rate (inverse of call holding time). Place-3 in Figure [4.5] represents the maximum number of packets that could possibly be coming into the multiplexer simultaneously. Initially N tokens are placed in this place. Place-4 represents the actual number of packets simultaneously entering the multiplexer at a given time. Initially, there are no tokens in this place.

Tokens move from place-3 to place-4 through transition-3 with a firing rate of $(i - j)\lambda$, where j is the number of tokens in place-4 or the number of packets entering the multiplexer simultaneously at that time. Therefore, $(i - j)$ is the number of connected sources that are not sending packets at that time but could potentially do so. λ is the packet generation rate for each connected source. The tokens move from place-4 to place-3 through transition-4 with a firing rate of $j\mu$, where μ is the packet service rate. We are interested in finding the probability that there are x tokens in place 4.

The GSPN (Generalized Stochastic Petri Net) software package was used to solve the Petri Net shown in Figure [4.5] using the same parameters as given in the previous section.

Prob (x sources transmitting simultaneously) = Prob (x tokens in place-4)

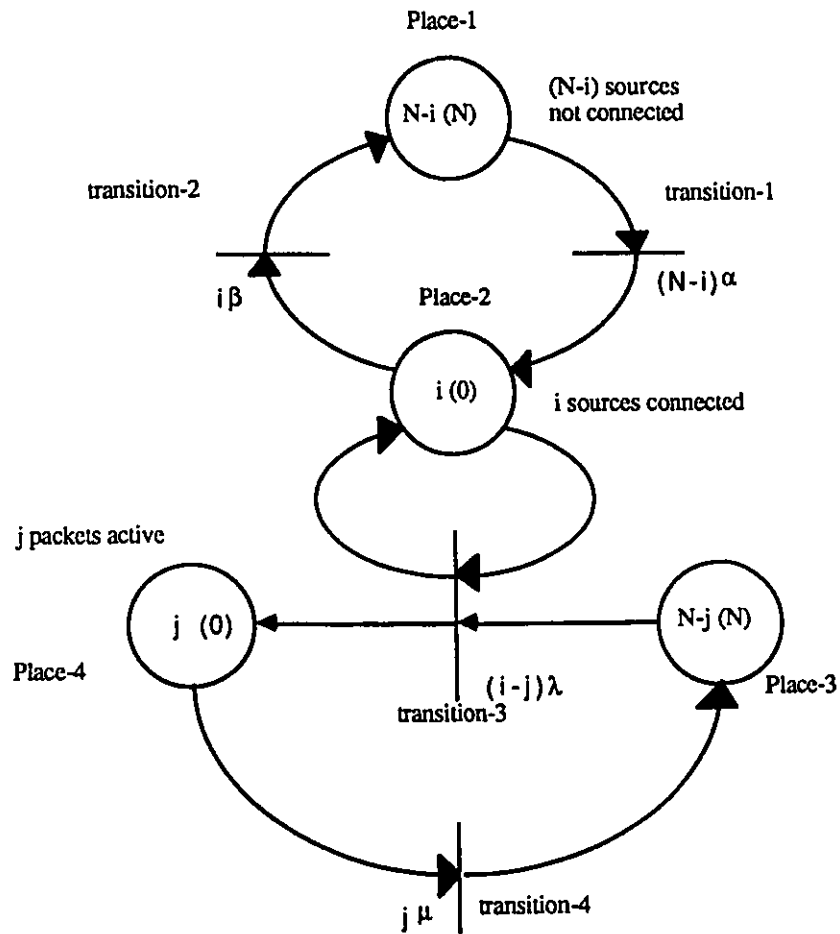


Figure [4.5]: Petri Net Model Of Multiplexer with N inputs

Number of Inputs Transmitting Simultaneously (x)	Probability of finding x multiplexer inputs active		
	Gauss-Seidel Method	Petri net Model	Closed form solution
0	0.041351	0.0437474	0.04392576
1	0.136227	0.1392782	0.1394468
2	0.2185075	0.2191810	0.21913077
3	0.2290537	0.2274351	0.2272467
4	0.1771881	0.1750912	0.174944
5	0.10881118337	0.1066776	0.10663248
6	0.05425939	0.0535812	0.0535983
7	0.02305416	0.0228193	0.02284918
8	0.008468432	0.0084116	0.0084324
9	0.00273247	0.0027262	0.00273645
10	0.00073422	0.0007865	0.00079053
11	0.00020222	0.0002040	0.00020533
12	0.00004724	0.0000480	0.0000483455
13	0.00001007	0.0000103	0.000010389
14	0.00000197	0.000002	0.00000205
15	0.000000355	0.0000004	0.000000373
16	0.00000006	0.0000001	0.0000000629
17	0.000000009		0.00000000987
18			0.00000000144

Table [4.1]: Probability of finding a given number of Multiplexer inputs active. Using Markov Chain model (solved by Gauss-Seidel method), Petri Net Model (solved by GSPN, accuracy 0.0004) and Closed Form solution. (Number of inputs=100, traffic intensity=2.77 CCS and burstiness factor=2.5)

4.3.3 Drawbacks with the above methods

We numerically solved the model using both methods. The values of the probability of finding x packets transmitting simultaneously are given in Table [4.1]. This table also gives results obtained using closed form model described in next section. Even though the results for the above two methods, namely the Markov chain (solved using the Gauss-Seidel method) and the Petri Net model are close, they are not suitable for solving large (with more than a hundred inputs to the multiplexer) models. The limitation is the memory and computation requirement.

- GSPN uses the Gaussian elimination method to get the probabilities for each place in the Petri Net. Consequently, the computation time and the memory requirement increases exponentially ($O(s^2)$) with increase in the number of states in the Markov Chain formed.
- The Gauss-Seidel method for solving linear equations for the Markov chain has better computation time as compared to the Gaussian elimination method [YOUN71]. Even though the Gauss-Seidel method was executed for shorter duration, it converges faster than the Petri Net method and hence gives more accurate results when compared with those from the closed form solution in Table [4.1].

In Gauss-Seidel method also, the time and memory requirements increase as the number of inputs is increased. $(N + 1)(N + 2)/2$ double precision variables are required to be stored.

4.3.4 Closed Form Solution

Bayes rule can be used to get the probability that x packets are entering simultaneously into the multiplexer.

$$\text{Prob (x packets entering)} =$$

$$\text{Prob (x sources transmitting)} =$$

$$\sum_{y=x}^N \text{Prob (x sources transmitting | y sources connected)} \cdot \text{Prob (y sources connected)} \quad (4.11)$$

If N , α , β , λ and μ have the usual meaning as described earlier and PC is the probability that a source is connected on a link, then

$$\text{Prob (y calls connected out of N links)}$$

$$\begin{aligned}
&= \binom{N}{y} P_C^y (1 - P_C)^{N-y} \\
&= \binom{N}{y} \left[\frac{\alpha}{\alpha + \beta} \right]^y \left[1 - \frac{\alpha}{\alpha + \beta} \right]^{N-y} \\
&= \frac{\binom{N}{y} \left[\frac{\alpha}{\beta} \right]^y}{\left[1 + \frac{\alpha}{\beta} \right]^N} \tag{4.12}
\end{aligned}$$

If P_T is defined as the probability that a connected source transmits a packet, then

Prob (x cells entering | y calls connected)

$$\begin{aligned}
&= \binom{y}{x} P_T^x (1 - P_T)^{y-x} \\
&= \binom{y}{x} \left[\frac{\lambda}{\lambda + \mu} \right]^x \left[1 - \frac{\lambda}{\lambda + \mu} \right]^{y-x} \\
&= \frac{\binom{y}{x} \left[\frac{\lambda}{\mu} \right]^x}{\left[1 + \frac{\lambda}{\mu} \right]^y} \tag{4.13}
\end{aligned}$$

Therefore, by substituting equations (4.12) and (4.13) into equation (4.11)

$$P(x) = \sum_{y=x}^N \frac{\binom{y}{x} \left[\frac{\lambda}{\mu} \right]^x}{\left[1 + \frac{\lambda}{\mu} \right]^y} \frac{\binom{N}{y} \left[\frac{\alpha}{\beta} \right]^y}{\left[1 + \frac{\alpha}{\beta} \right]^N} \tag{4.14}$$

This solution gives the probability of the event (that there are x packets entering simultaneously into the multiplexer) directly and therefore requires minimum storage and computation. It can be used to evaluate the probabilities for large values of N (of the order of a few thousand) quite easily by a computer program.

If λ and μ are replaced by the Burstiness Factor **BF** in the above equation, then from equation (4.5),

Substituting $BF = \frac{\lambda + \mu}{\lambda}$,

$$P(x) = \sum_{y=x}^N \frac{\binom{y}{x} [BF - 1]^{y-x} \binom{N}{y} \left[\frac{\alpha}{\beta}\right]^y}{BF^y \left[1 + \frac{\alpha}{\beta}\right]^N} \quad (4.15)$$

If the Burstiness Factor is unity, then $(BF - 1) \Rightarrow 0$. All the terms in the summation disappear except when $y = x$. When y is equal to x :

$$P(x) = \frac{\binom{N}{x} \left[\frac{\alpha}{\beta}\right]^x}{\left[1 + \frac{\alpha}{\beta}\right]^N} \quad (4.16)$$

This result applies if all the sources are CBO (Continuous Bit Oriented). Each source is continuously sending packets every T seconds. The above probability is nothing but the probability that there are x sources out of N connected at any time.

If the Burstiness Factor is large, $(BF - 1) \Rightarrow BF$.

$$P(x) = \frac{1}{BF^x} \left[\sum_{y=x}^N \frac{\binom{y}{x} \binom{N}{y} \left[\frac{\alpha}{\beta}\right]^y}{\left[1 + \frac{\alpha}{\beta}\right]^N} \right] \quad (4.17)$$

4.3.5 Results

The probability of cell loss is the probability that there are more than M inputs transmitting simultaneously.

$P_{\text{loss}} = \text{Probability of cell loss}$

$$= \frac{\text{Expected number of cells lost in T period}}{\text{Expected number of cells transmitted in T period}}$$

$$= \frac{\sum_{x=M+1}^N P(x) (x - M)}{\sum_{x=0}^N P(x) x} \quad (4.19)$$

If each input transmits one cell every T second period, the average cell loss rate ($R_{\text{cell loss}}$) for the input source is given by:

$$R_{\text{cell loss}} = \frac{1}{T} P_{\text{loss}} \text{ cells/second} \quad (4.20)$$

Cell loss probability for a 20 input multiplexer is evaluated by equation [4.19] using different values of trunk capacity (number of cell buffers provided). A traffic intensity of 12 CCS and a Burstiness Factor of 2.5 is assumed. The cell loss probabilities are shown in Figure [4.6]. The cell loss probabilities have also been confirmed by simulation on QNAP2 as given in Table [4.2]. It should be noticed that the cell loss probability is unity at trunk capacity of zero. The cell loss probability drops sharply with increase in trunk capacity and approaches 10^{-11} at a trunk capacity of about 15.

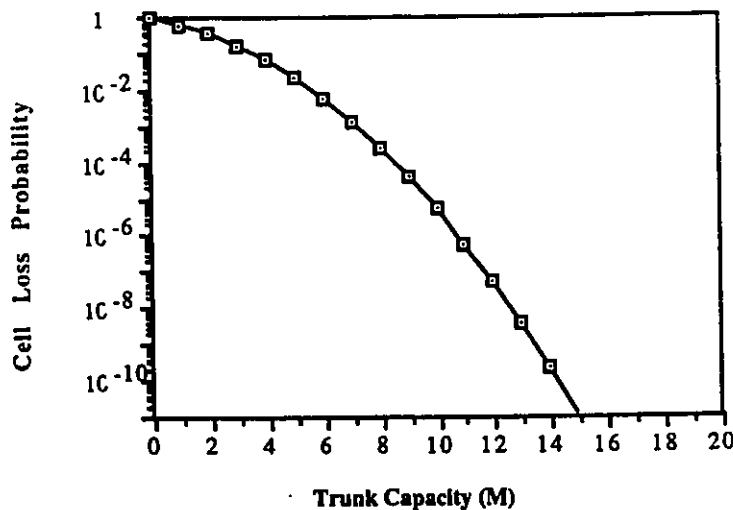


Figure [4.6]: Cell Loss Probabilities vs Trunk Capacity (Trunk Capacity is defined as the number of input sources that can be carried without information loss by the output trunk of the multiplexer). Number of inputs $N=20$ and burstiness factor= 2.5 .

Trunk Capacity (total inputs=20)	Probability of Cell Loss	
	Analytical Results	Simulation Results (confidence level=0.95)
0	1.0	1.0
1	0.6464	0.6434
2	0.3588	0.3550
3	0.1676	0.1646
4	0.0653	0.06432
5	0.0212	0.02116
6	0.005747	0.005875
7	0.001302	0.001385
8	0.000247	0.0002915
9	0.3926e-4	0.4467e-4
10	0.52128e-5	0.7192e-5
11	0.5758e-6	0.3785e-6
12	0.5253e-7	*
13	0.3914e-8	*
14	0.2343e-9	*
15	0.11e-10	*

Table [4.2]: Cell Loss Probabilities from Closed Form Solution and Simulation. Number of inputs N=2400 and burstiness factor=2.5. (* means that the values could not be obtained)

Variation of load

The probability of having x number of inputs transmitting simultaneously into the multiplexer, P(x), has been plotted in Figure [4.7] for traffic intensity varying from 2.77 CCS to 18 CCS. The multiplexer has 2400 inputs. All other parameters are kept constant (mean call holding time = 180 seconds, mean packet (or burst) length = 1.2 seconds, mean inter-packet length = 1.8 seconds). The values of burst length and inter-burst length have been studied for voice talk-spurts and pauses by Brady [BRAD68].

It should be noted that $P(x)$ peaks approximately at

$$N_{avg} = N \frac{\bar{A}}{\bar{A} + \bar{B}} \frac{\bar{p}}{\bar{p} + \bar{q}}$$

$$N_{avg} = N \frac{\alpha}{\alpha + \beta} \frac{\lambda}{\lambda + \mu} \quad (4.21)$$

The N_{avg} is nothing but the expected number of packets transmitting simultaneously. When the traffic intensity is increased, the N_{avg} increases as well. This is so because the variable A (probability of finding the line connected) increases.

If traffic from each input source is considered to be a stochastic variable, each curve in Figure [4.7] represents the sum of N (total number of inputs to the multiplexer) such identical independent variables. Therefore, some sort of Central Limit Theorem applied, so that we can approximate the distribution of the sum by a Gamma distribution. The Gamma distribution is chosen because it is close to Normal distribution and it has positive support [ECKB88]. With logarithmic $P(x)$ axis, each of the curves very closely resembles a Normal probability distribution. As the traffic intensity is increased, the mean and the variance of the distribution increases as well.

Cell loss probability P_{loss} is plotted in Figure [4.8] for different values of traffic intensity. At trunk capacity of zero, the cell loss probability is unity. The P_{loss} reduces as the trunk capacity is increased. For lower values of traffic intensity, the required trunk capacity to achieve a given cell loss probability is lower.

Figure [4.9] shows P_{loss} against source utilization ρ ($= CCS/36$) for BWG (bandwidth gain) of 20, 10, 6.67, 5 and 4. As the r is increased, so does the P_{loss} , as expected. The lower the trunk capacity, the lower is the source utilization (traffic intensity) at which a given cell loss probability is achieved.

In Figure [4.12] **Bandwidth Gain (BWG)** (defined as the ratio of the sum of input peak rates of all the sources to the multiplexer output trunk capacity) is plotted against source utilization for different values of cell loss probabilities. It is clear that the BWG is higher for low source utilization (low value of traffic intensity). It should also be noted that the BWG is higher if the acceptable cell loss probability is higher.

Variation in Burstiness Factor

Figure [4.10] gives $P(x)$ for various values of Burstiness Factor (BF) (1.015, 1.15, 2.5, 16.0, 150, 1500). The traffic intensity is maintained constant at 2.77 CCS. It is clear that if BF is increased, N_{avg} decreases because

$$N_{avg} = N \frac{\bar{A}}{\bar{A} + \bar{B}} \frac{\bar{p}}{\bar{p} + \bar{q}}$$

$$N_{avg} = N \frac{\alpha}{\alpha + \beta} \frac{1}{BF} \quad (4.22)$$

or N_{avg} is inversely proportional to the Burstiness Factor.

Figure [4.11] shows the P_{loss} curves for BFs of 1.015, 1.5, 2.5, 16.0, 150, 1500. The traffic intensity is maintained at 2.77 CCS. For more bursty sources, the required trunk capacity to achieve the same cell loss probability is low because, the packet lengths are smaller as compared to the inter-packet periods.

In Figure [4.13] Bandwidth Gain (BWG) is plotted against Burstiness Factor. The higher the Burstiness Factor, the larger the Bandwidth Gain. It should also be noticed that the BWG is higher if higher cell loss probability can be tolerated.

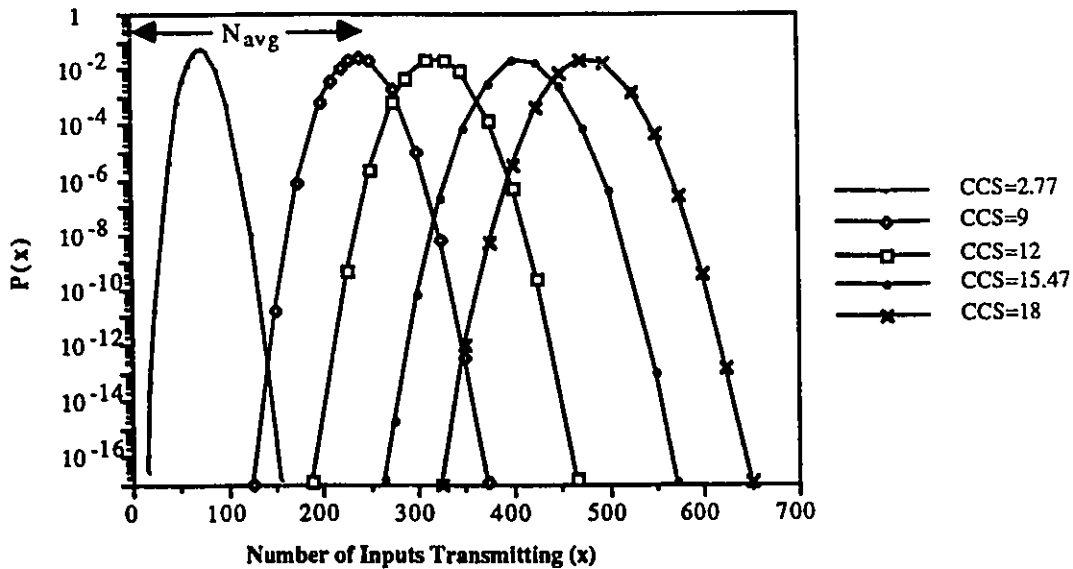


Figure [4.7]: Variation of $P(x)$ for different values of traffic intensity. Number of inputs $N=2400$ and burstiness factor=2.5.

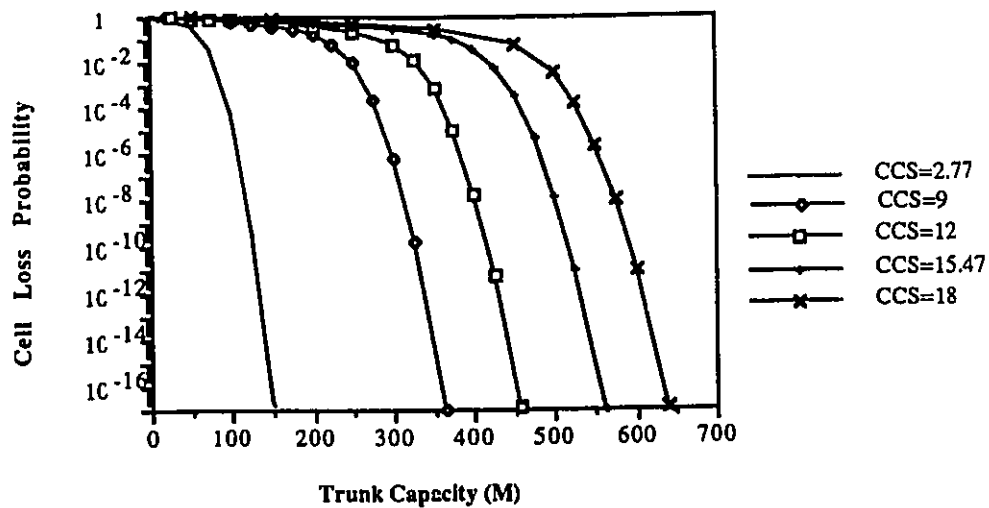


Figure [4.8]: Probability of cell loss against multiplexer trunk capacity (for different values of traffic intensity). Number of inputs $N=2400$ and burstiness factor=2.5.

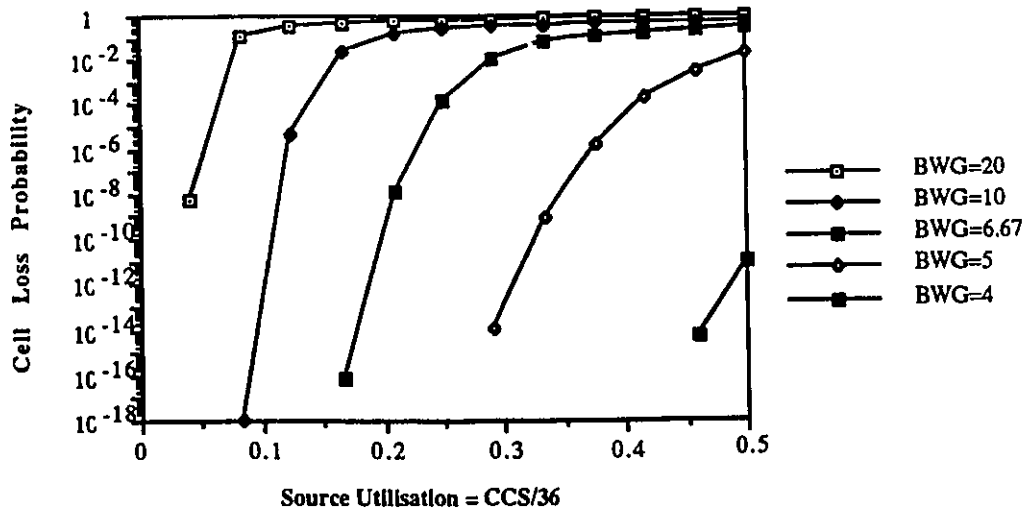


Figure [4.9]: Prob. of Cell Loss vs Source Utilization for different values of Bandwidth ratio provided. Number of inputs $N=2400$ and burstiness factor=2.5.

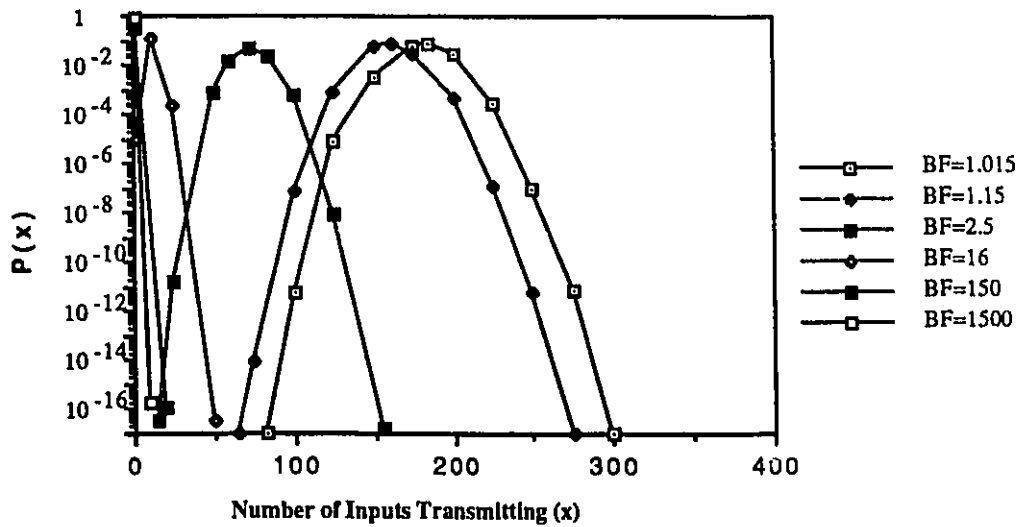


Figure [4.10]: Variation of $P(x)$ for different values of BF . Number of inputs $N=2400$ and traffic intensity = 2.77 CCS.

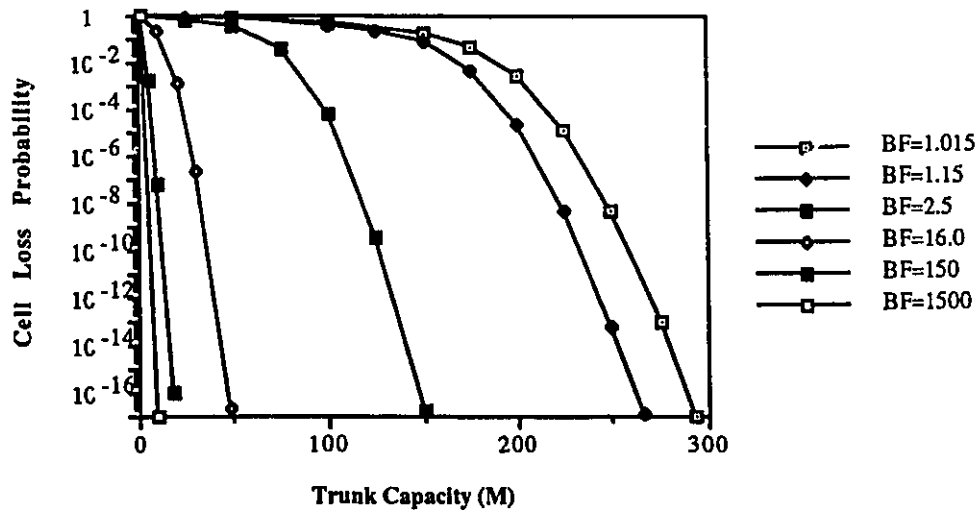


Figure [4.11]: Probability of cell loss against multiplexer trunk capacity (for different values of BF). Number of inputs $N=2400$ and traffic intensity = 2.77 CCS.

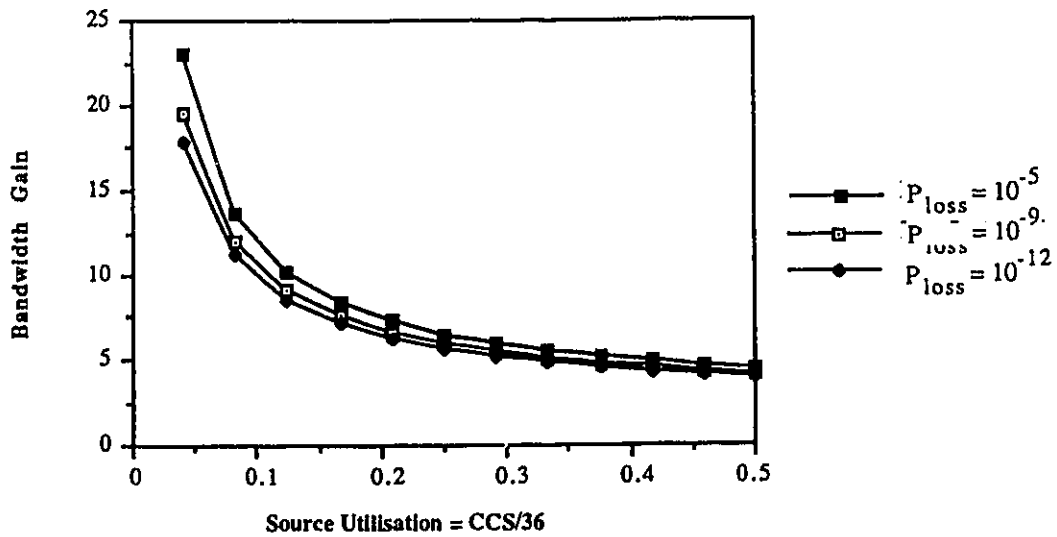


Figure [4.12]: Bandwidth Gain achieved for a given source utilization and cell loss probability. Number of inputs $N=2400$ and burstiness Factor=2.5.

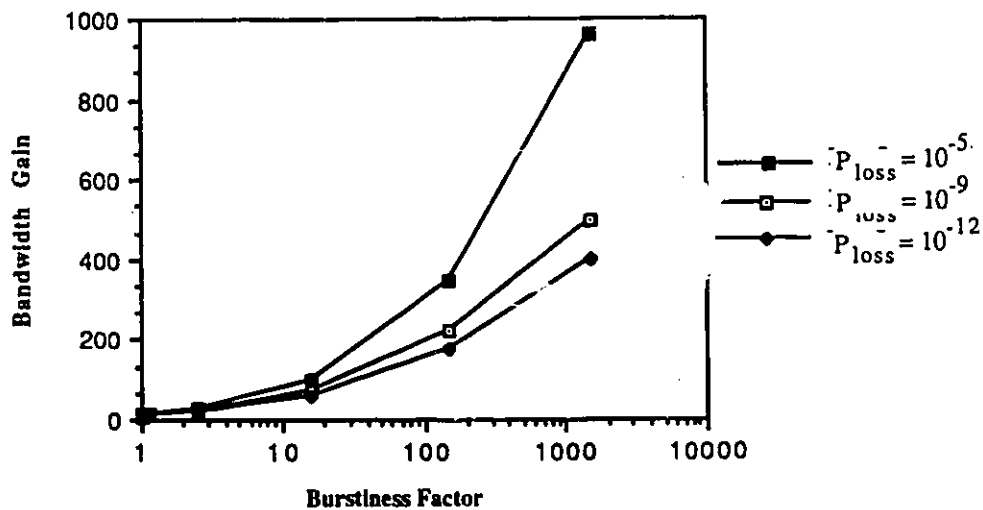


Figure [4.13]: Bandwidth Gain achieved for a given Burstiness Factor and cell loss probability. Number of inputs $N=2400$ and traffic intensity = 2.77 CCS.

Chapter 5

Extensions to the Model

Two extensions to the model are considered here. In the first case, it is assumed that z traffic type groups, group- i having N_i sources characterized by α_i , β_i , λ_i and μ_i , are multiplexed. Examples come naturally from integration of different services (ISDN). In the second case, it is assumed that each traffic source generates multiple types of traffic, although the traffic characteristics are identical among all the input sources. Each source generates z traffic types, traffic type- i defined by α_i , β_i , λ_i and μ_i . An example of this scenario is a multimedia workstation in which voice, data and video and other types of traffic are generated.

In order to resolve the resource contention among various sources, when sharing is involved, each traffic type has a priority level associated with it. If the contention is between two sources having different priorities, the higher priority source gets the resource. If the contention is between sources of the same priority, the allocation of resource depends upon the design of the multiplexer (e.g. random assignment). For simplicity, we consider two traffic groups type-1 and type-2 with the former having higher priority than the later. The cell loss for each traffic type is evaluated under the following buffer (assuming total number of buffers available is n) management schemes:

1. Complete Partitioning (CP)

In complete partitioning, shown in Figure [5.1a], the resource is divided up among the traffic groups on a static basis. In the figure, n_1 buffers have been allocated to type-1 traffic and n_2 buffers have been allocated to type-2 traffic. A source in one group does not have access to resources allocated to another group, even when the resource is unused.

2. Complete Sharing (CS)

In complete sharing, all the resources are shared among the users. As shown in Figure [5.1b], all n buffers are shared between the two traffic groups. The resource contention among various traffic groups may be resolved on a priority basis.

3. Sharing with Minimum Allocation (SMA)

In this scheme, a minimum amount of resources is allocated to each traffic group. In Figure [5.1c], n_1 buffers have been pre-allocated to type-1 traffic and n_2 buffers have been pre-allocated to type-2 traffic. The rest $(n-n_1-n_2)$ buffers are shared between the two traffic groups.

4. Sharing with Maximum Limit (SML)

In this scheme, all the resources are shared but there is a maximum limit on the amount of resources occupied by only one group (Figure [5.1d]). In Figure [5.1d], type-1 traffic can use n_1 buffers and type-2 traffic can use up to n_2 buffers at the most.

When only two traffic groups are considered, the last two sharing schemes (SMA and SML) are identical. If n_1 and n_2 in SMA is replaced with $(n-n_2)$ and $(n-n_1)$ respectively in SML, the performance results should be identical. The vice versa is also true.

Before we consider the above two cases separately, we define

$$P(x) = P(x_1, x_2, x_3, \dots, x_z) \quad (5.1)$$

where $P(x)$ is the probability that x_1 type-1, x_2 type-2, \dots , x_z type- z sources are simultaneously transmitting at a given time.

If the traffic groups are independent, then

$$P(x) = P_1(x_1) P_2(x_2) P_3(x_3) \dots P_z(x_z) \quad (5.2)$$

Where $P_i(x_i)$ is the probability that x_i sources of the i th group are transmitting simultaneously.

It is assumed that the multiplexer sample time T is the cell generation period of the highest peak bit rate source. For example, consider two types of sources s_1 and s_2 are multiplexed as shown in Figure [5.2]. If the source s_2 bit rate is twice that of s_1 , the multiplexer samples cells every cell period T and gets one cell in each period from each s_2 source while it gets one cell every alternate period from the s_1 source provided both are transmitting. If a number N_1 of s_1 sources and a number N_2 of s_2 sources are multiplexed, group-2 should be assigned proportionally $(2N_2/N_1)$ more cell buffers than those assigned to traffic group-1. For simplicity of calculations, it is assumed that the different traffic groups have the same traffic rate so that each source may transmit one cell in T seconds.

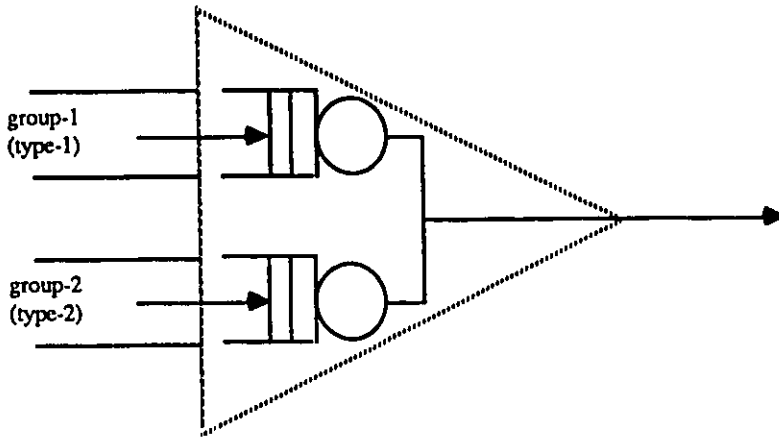
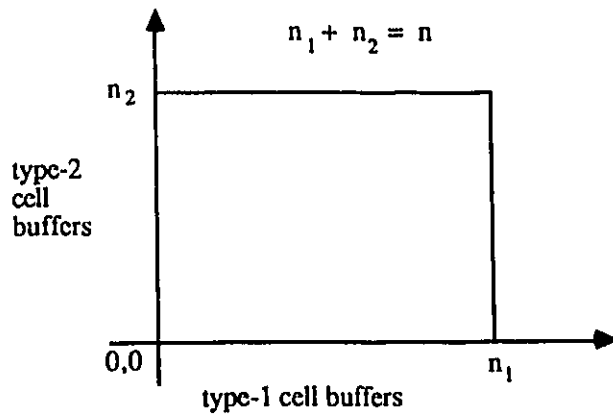


Figure [5.1a]: Complete Partitioning of Cell buffers

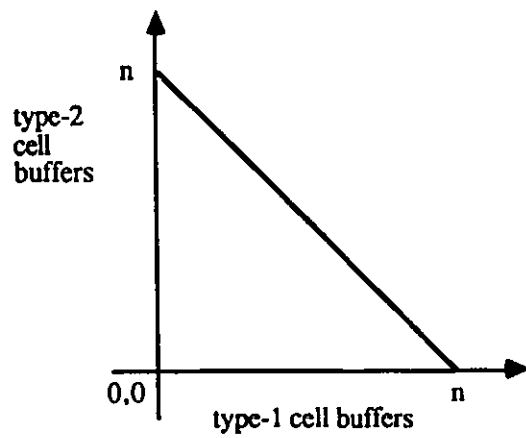


Figure [5.1b]: Complete Sharing of Cell Buffers

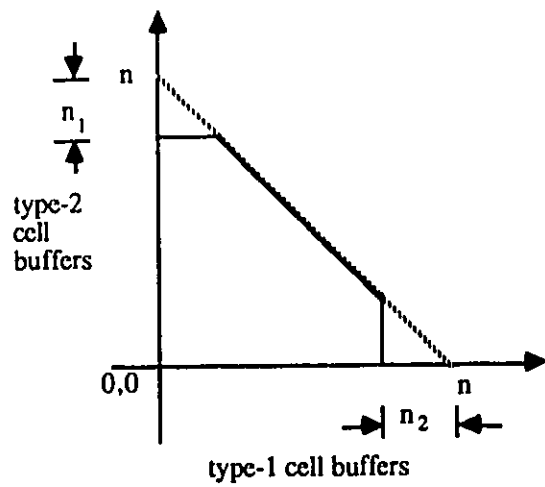


Figure [5.1c]: Sharing with Minimum Allocation

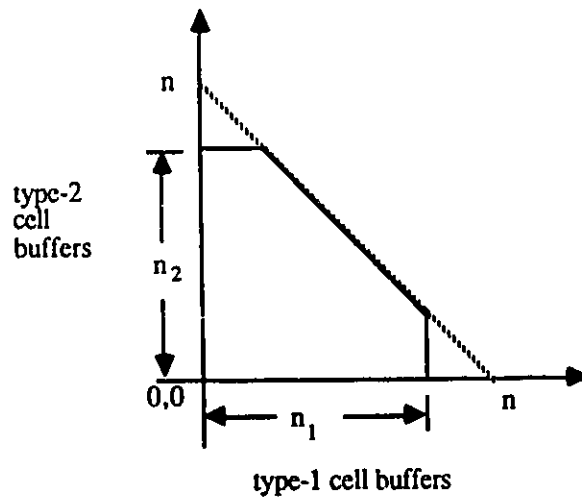


Figure [5.1d]: Sharing with Maximum Limit

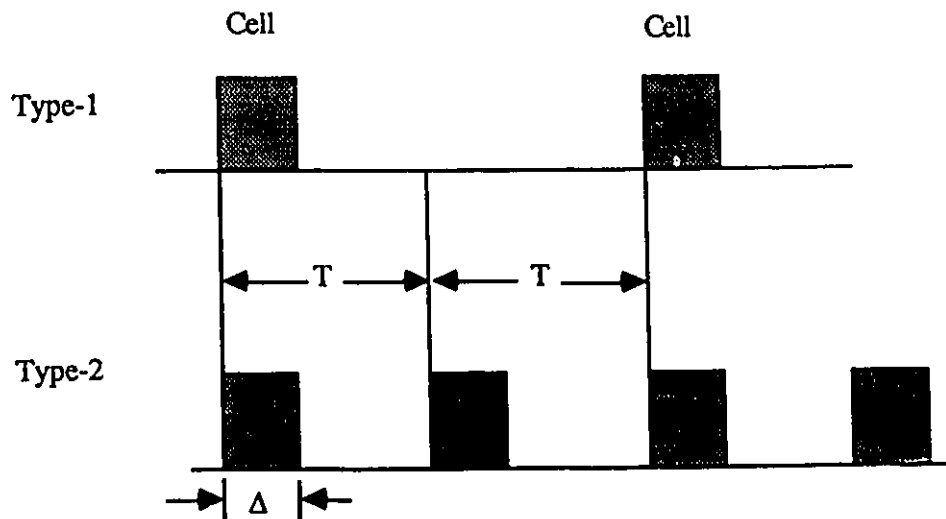


Figure [5.2]: Cells generated by type-1 and type-2 sources as they appear on the output trunk

5.1 Non-identical Traffic Sources

In our model, the ATM layer buffers the cells generated by the adaptation layer. It buffers the higher priority cells first and discards any cells that overflow the buffer capacity. The upper layers have no direct feedback from the ATM layer about the lost cells. In other words, the ATM provides a fixed throughput pipe to the traffic sent by the higher layers and in case of resource (cell buffer) contention, the lower priority cells are discarded without giving any direct feedback to the upper layers. Since the ATM layer and the higher layers operate independently, $p_i(n_i)$ on the right hand side of equation (5.2) can be simply written as (from equation (4.14)):

$$P_i(n_i) = \sum_{y=n_i}^{R_i} \frac{\binom{y}{n_i} \left[\frac{\lambda_i}{\mu_i} \right]^{n_i}}{\left[1 + \frac{\lambda_i}{\mu_i} \right]^y} \frac{\binom{R_i}{y} \left[\frac{\alpha_i}{\beta_i} \right]^y}{\left[1 + \frac{\alpha_i}{\beta_i} \right]^{R_i}} \quad (5.3)$$

Where $P_i(n_i)$ is the probability that n_i sources of i th group are transmitting simultaneously. R_i is the total number of sources in the i th group and α_i (call arrival rate), β_i (call service rate), λ_i (packet arrival rate) and μ_i (packet service rate) characterize the parameters for the i th traffic group as defined earlier.

For simplicity, let us assume that two groups of traffic sources are being multiplexed. Each group has different call arrival rate, call holding time, packet arrival rate and packet length.

Complete Partitioning:

In complete partitioning n_1 buffers are allocated to type-1 traffic and n_2 buffers are allocated to type-2 traffic. Since every group has its own resources to share independently, the analysis is simplified. Each group- i can be independently analyzed as if it were multiplexed by a smaller multiplexer (with n_i cell buffers). Figure [5.1a] (bottom) shows that each group sends traffic to a different queue. Figure [5.1b] (top) shows complete resource partitioning scheme. The blocking probability of type- i traffic is given by:

$$P_{\text{loss}}(i) = \frac{\sum_{x=n_i+1}^{R_i} P_i(x) (x - n_i)}{\sum_{x=0}^{R_i} P_i(x) x} \quad (5.4)$$

where $P_i(x)$ is the probability that there are x sources transmitting simultaneously in i th traffic group. And n_i , as described earlier, is the number of cell buffers reserved for type- i traffic group and R_i is total number of type- i sources in the multiplexer.

The results for cell loss probability for each group of traffic sources will be same as those given in Figure [4.6].

Complete Sharing:

Assuming that type-1 traffic has priority over type-2 traffic, the blocking probability of type-1 traffic will be independent of the amount of type-2 traffic being handled by the multiplexer.

$$P_{\text{loss}}(1) = \frac{\sum_{x=n+1}^{R_1} P_1(x) (x - n)}{\sum_{x=0}^{R_1} P_1(x) x} \quad (5.5)$$

Where $P_1(x)$ is the probability that there are x sources transmitting simultaneously in type-1 traffic group. And n is the total number of cell buffers and R_1 is the total number of type-1 traffic sources in the multiplexer.

The type-2 traffic is however influenced by the level of type-1 traffic being handled by the multiplexer. Assuming that n is the total number of cell buffers available, if x number of type-1 sources are active, then only $(n-x)$ number of type-2 traffic sources could be handled without cell loss by the multiplexer. It is further assumed that, each source occupies one cell buffer every T second when it is transmitting. Therefore, the probability of cell loss for type-2 traffic is:

$$P_{\text{loss}}(2) = \frac{\sum_{x=0}^n P_1(x) \sum_{y=n-x+1}^{R_2} P_2(y) (y - n + x) + \sum_{x=n+1}^{R_1} P_1(x) \sum_{y=1}^{R_2} P_2(y) y}{\sum_{x=0}^{R_2} P_2(x) x} \quad (5.6)$$

Where R_1 and R_2 are total number of type-1 and type-2 traffic sources into the multiplexer respectively. The first term on the right hand side of equation (5.6) represents type-2 cell loss probability when there are n or less type-1 sources transmitting. The second term gives the cell loss when there are more than n type-1 sources transmitting and all the cells transmitted by the type-2 traffic are lost. Figure [5.1b] shows the cell buffer allocation in the complete sharing scheme.

The cell loss probabilities for a 40 input multiplexer, having 20 high priority input sources and 20 low priority input sources, are evaluated by equation [5.5] and equation [5.6] respectively. The results obtained are confirmed by simulation on QNAP2 (the values for very low $\approx 10^{-5}$ probabilities are not reliable in simulation because of simulation time limitations). The cell loss probabilities obtained are listed in Table [5.1] and plotted in Figure [5.3] for high as well as low priority cells. It should be noted that the higher priority cells have same probability of loss as that for one type of traffic given in Figure [4.6]. The lower priority traffic does not affect the cell loss probability of the high priority traffic. The low priority cells, as expected, have higher cell loss probability for same trunk capacity.

Trunk Capacity (total inputs=20)	Probability of Cell Loss			
	Analytical Results		Simulation Results (confidence level=0.95)	
	High Priority	Low Priority	High Priority	Low Priority
0	1.0	1.0	1.0	1.0
1	0.6464	0.9798	0.6434	0.9797
2	0.3588	0.9011	0.3550	0.9001
3	0.1676	0.7488	0.1646	0.7465
4	0.0653	0.5515	0.06432	0.5479
5	0.0212	0.3588	0.02116	0.3558
6	0.005747	0.2068	0.005875	0.2053
7	0.001302	0.1063	0.001385	0.1058
8	0.000247	0.048874	0.0002915	0.04921
9	0.3926e-4	0.0202	0.4467e-4	0.02069
10	0.52128e-5	0.007524	0.7192e-5	0.007828
11	0.5758e-6	0.002532	0.3785e-6	0.002677
12	0.5253e-7	0.7708e-3	*	0.8224e-3
13	0.3914e-8	0.2128e-3	*	0.2253e-3
14	0.2343e-9	0.533e-4	*	0.533e-4
15	0.11e-10	0.1213e-4	*	0.9554e-5

Table [5.1]: Cell Loss Probabilities for High Priority and Low Priority Cells for two types of traffic sources in Complete Sharing scheme. (* means that the results could not be obtained)

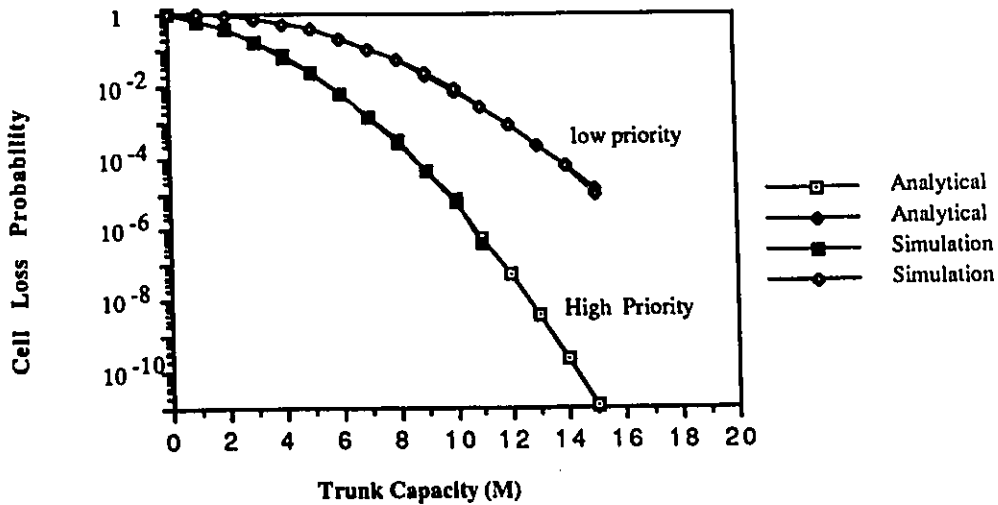


Figure [5.3]: Cell Loss Probabilities for High and Low Priority Cells Generated by Different Sources in Complete Sharing scheme.

Sharing With Minimum Allocation:

In this scheme, n_1 cell buffers are reserved for type-1 traffic and n_2 cell buffers are reserved for type-2 traffic. The remaining $(n - n_1 - n_2)$ cell buffers are shared according to the priority.

Since type-1 has higher priority, it can use up to $(n-n_2)$ cell buffers and thus

$$P_{\text{loss}}(1) = \frac{\sum_{x=n-n_2+1}^{R_1} P_1(x) (x - n + n_2)}{\sum_{x=0}^{R_1} P_1(x) x} \quad (5.7)$$

Type-2 traffic can use its n_2 cell buffers without fear of losing them. It can also use rest of $n-n_1-n_2$ cell buffers if they are not required by the type-1 traffic. So

$$P_{\text{loss}}(2) = \frac{\sum_{x=0}^{n_1} P_1(x) \sum_{y=n-n_1+1}^{R_2} P_2(y) (y-n+n_1) + \sum_{x=n_1+1}^{n-n_2} P_1(x) \sum_{y=n-x+1}^{R_2} P_2(y) (y-n+x) + \sum_{x=n-n_2+1}^{R_1} P_1(x) \sum_{y=n_2+1}^{R_2} P_2(y) (y-n_2)}{\sum_{x=0}^{R_2} P_2(x) x} \quad (5.8)$$

Figure [5.1c] shows the mapping of cell buffer allocation in this scenario.

The first term on the right hand side of equation (5.8) represents the cell loss probability for type-2 traffic when type-1 traffic is using less than or equal to its allocated n_1 cell buffers. Type-2 traffic can use the remaining $(n-n_1)$ buffers. The second term represents the case when type-1 traffic is using $(n-n_2)$ or less buffers so that the type-1 traffic can use the remaining cell buffers. The third term gives the situation in which more than $(n-n_2)$ type-1 sources are transmitting so that the type-2 traffic can use its minimum allocated n_2 buffers only.

The cell loss probabilities for SMA scheme are computed for a multiplexer with 40 inputs, half (20) of which have higher priority than the rest (20). A total of 10 cell buffers are provided in the multiplexer. A traffic intensity of 12 CCS and a burstiness factor of 2.5 is assumed for either traffic types. The results for cell loss probabilities, computed analytically, are shown in Figure [5.4]. The "High Priority" and "Low Priority 1" curves show cell loss probabilities for high and low priority traffic respectively when buffers are allocated only to the low priority traffic as per SMA scheme. It should be noted that the high priority cell loss increases gradually and reaches unity when all (ten) buffers have been allocated to the lower priority traffic.

The "Low Priority 2" curve shows the cell loss for low priority traffic when buffers are allocated only to the higher priority traffic according to the SMA scheme. Again note that the low priority cell loss probability keeps increasing as buffers allocation to high priority traffic is increased. The cell loss probability is unity when all (ten) the buffers have been assigned to the higher priority traffic.

Equation (5.8) may be used to derive the solutions for CP and CS cases. Other interesting scenarios (for example cell buffers may be reserved for only one source but not the other) may also be analyzed with this equation.

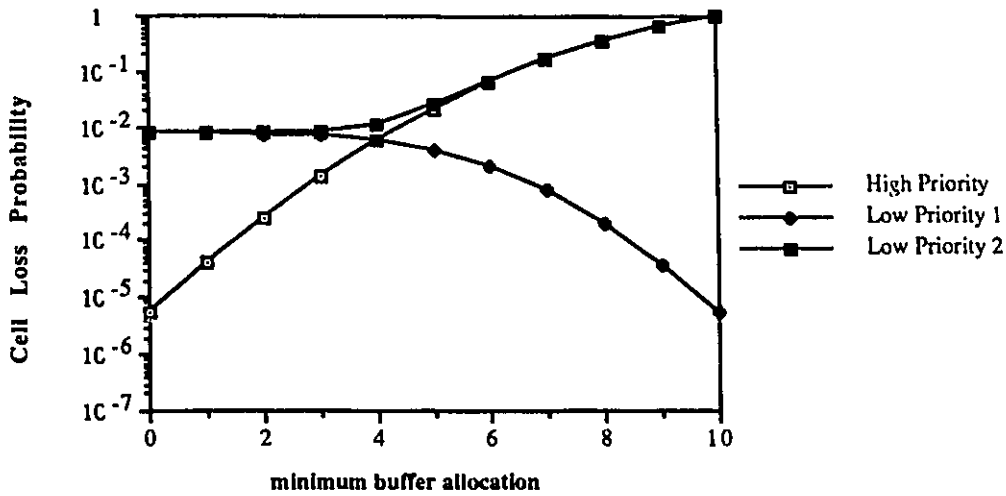


Figure [5.4]: Cell Loss Probabilities in Sharing with Minimum Allocation Scheme.

Sharing With Maximum Limit:

In this scheme, type-1 traffic can use only n_1 cell buffers at the most and type-2 traffic can use only n_2 cell buffers at the most. Otherwise all the buffers are shared.

Since type-1 has higher priority, it can use up to n_1 cell buffers. So

$$P_{\text{loss}}(1) = \frac{\sum_{x=n_1+1}^{R_1} P_1(x) (x - n_1)}{\sum_{x=0}^{R_1} P_1(x) x} \quad (5.9)$$

Type-2 traffic can use $(n-n_1)$ cell buffers without fear of losing them. It can also use the rest of $n-n_1-n_2$ cell buffers if they are not required by the type-1 traffic. Therefore

$$P_{\text{loss}}(2) = \frac{\sum_{x=0}^{n-n_2} P_1(x) \sum_{y=n_2+1}^{R_2} P_2(y) (y-n_2) + \sum_{x=n-n_2+1}^{n_1} P_1(x) \sum_{y=n-x+1}^{R_2} P_2(y) (y-n+x) + \sum_{x=n_1+1}^{R_1} P_1(x) \sum_{y=n-n_1+1}^{R_2} P_2(y) (y-n+n_1)}{\sum_{x=0}^{R_2} P_2(x) x} \quad (5.10)$$

Figure [5.1d] shows the mapping of cell buffer allocation in this scenario.

The first term right hand side of equation (5.10) represents cell loss probability for type-2 traffic when type-1 traffic is using $(n-n_2)$ or less buffers so that type-2 traffic can use n_2 buffers. The second term gives cell loss probability when type-1 traffic is using less than or equal to its maximum limit of n_1 but more than $(n-n_2)$ buffers. The type-2 traffic, with lower priority, can use the remaining buffers (out of a total of n). The third term represents the case when type-1 sources are transmitting more than n_1 cells so that type-2 traffic can only use $(n-n_1)$ buffers.

The cell loss probabilities for SML scheme are computed for a multiplexer with 40 inputs, half (20) of which have higher priority than the rest (20). A total of 10 cell buffers are provided in the multiplexer. A traffic intensity of 12 CCS and a burstiness factor of 2.5 is assumed for either traffic types. The results for cell loss probabilities, computed analytically, are shown in Figure [5.5]. The "High Priority" and "Low Priority 1" curves show cell loss probabilities for high and low priority traffic respectively when buffers are allocated only to the high priority traffic as per SML scheme. It should be noted that the cell loss probability for high priority traffic is unity when the its maximum buffer usage limit is zero. The cell loss probability reduces gradually when the maximum buffer limit is increased. The cell loss probability for the low priority traffic increases correspondingly.

The "Low Priority 2" curve shows the cell loss for low priority traffic when buffers are allocated only to the lower priority traffic according to the SML scheme. Again note that the low priority cell loss probability is unity when the maximum limit is zero. The cell loss probability for low priority traffic decreases as its maximum buffer usage limit is increased.

Equation (5.10) may also be used to derive the solutions for CP and CS cases. Other interesting scenarios (for example maximum limit on cell buffers used by one of the sources may be placed but not the other) may also be analyzed with this equation.

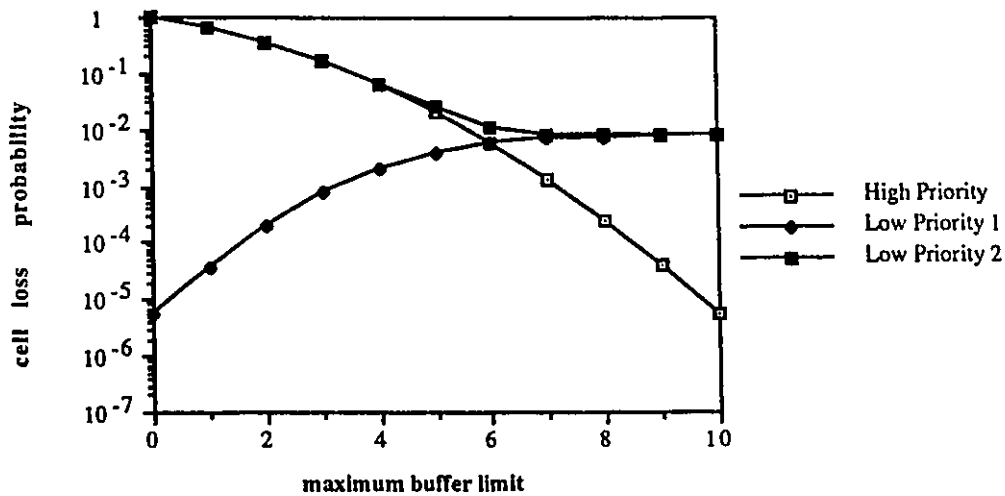


Figure [5.5]: Cell Loss Probabilities in Sharing with Maximum Limit Scheme.

5.2 Multimedia Sources

In this case, we analyze identical traffic sources generating more than one type of traffic. The traffic from each source is composed of super-imposed traffic types characterized by their packet generation rates and packet lengths. When the multiplexer scans the inputs every T second period, each input may send one cell per traffic type.

Here again, we consider a simplified scenario in which each source generates two types of traffic namely type-1 and type-2. A three dimensional Markov Chain can be constructed to describe the states of the multiplexer.

The state space $S(i, j, k)$ is constructed as shown in Figure [5.6] where i is the number of calls connected, j is the number of sources transmitting packets of type-1 and k is the number of sources transmitting packets of type-2.

In this case, the different traffic types depend on the establishment of a call. But, for a given number of calls connected, the number of sources transmitting traffic type-1 is independent of the number of sources transmitting type-2 traffic. The probability $P(j, k)$ of finding j number of

type-1 packets and k number of type-2 packets being transmitted simultaneously into the multiplexer is:

$$P(j, k) = \sum_{i=\max(j, k)}^N P(j, k | i) P(i)$$

$P(j, k | i)$ can also be split (as equation [5.1] was split into equation [5.2]) into two independent conditional probabilities because j and k are independent of each other, i.e.

$$P(j, k | i) = P(j | i) P(k | i)$$

or

$$P(j, k) = \sum_{i=\max(j, k)}^N P(j | i) P(k | i) P(i)$$

$$P(j, k) = \sum_{i=\max(j, k)}^N \frac{\binom{i}{j} \left[\frac{\lambda_1}{\mu_1} \right]^j}{\left[1 + \frac{\lambda_1}{\mu_1} \right]^i} \frac{\binom{i}{k} \left[\frac{\lambda_2}{\mu_2} \right]^k}{\left[1 + \frac{\lambda_2}{\mu_2} \right]^i} \frac{\binom{N}{i} \left[\frac{\alpha}{\beta} \right]^i}{\left[1 + \frac{\alpha}{\beta} \right]^N} \quad (5.11)$$

since the two traffic types are independent given that i calls are connected.

$S(i, j, k)$

i : number of calls connected
 j : number of type-1 packets tx
 k : number of type-2 packets tx

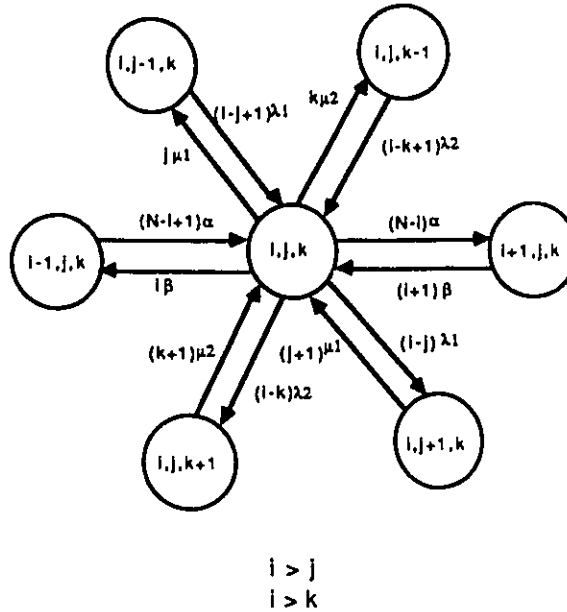


Figure [5.6]: Markov Chain Model of Multiplexer (Multimedia Sources - two types of traffic)

Complete Partitioning:

In the complete partitioning scheme, the two traffic types are considered completely independent like two smaller multiplexers each carrying different type of traffic.

$$P_{\text{loss}}(1) = \frac{\sum_{j=n_1+1}^N \sum_{k=0}^N P(j, k) (j - n_1)}{\sum_{j=0}^N \sum_{k=0}^N P(j, k) j} \quad (5.12)$$

Where $P_{\text{loss}}(1)$ is the probability of cell loss for type-1 traffic group. N is the total number of sources and n_1 is the number of cell buffers reserved for type-1 traffic.

Similarly

$$P_{\text{loss}}(2) = \frac{\sum_{k=n_2+1}^N \sum_{j=0}^N P(j,k) (k - n_2)}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) k} \quad (5.13)$$

The cell loss probabilities, computed numerically, for Complete Partitioning scheme are shown in Figure [5.7] for a multiplexer with 20 inputs. Again the traffic intensity for each source is 12 CCS and the burstiness factor is 2.5. The cell loss probability is unity when no cell buffers are allocated. As the buffer allocation is increased, the cell loss probability reduces.

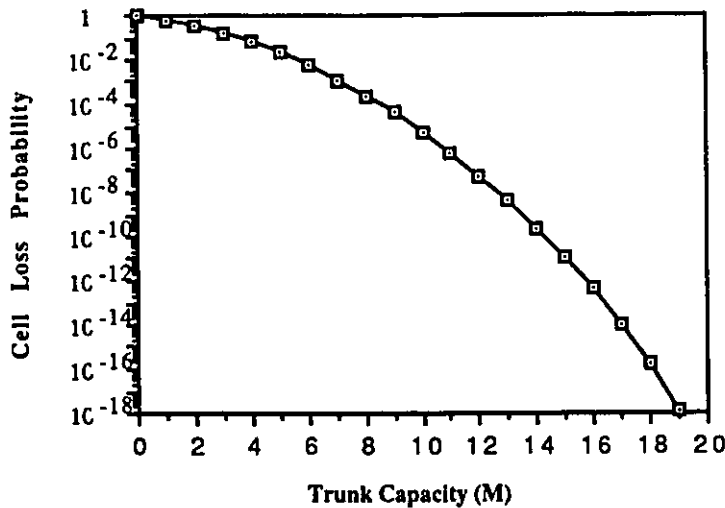


Figure [5.7]: Cell Loss Probability for Multimedia Sources using Complete Partitioning scheme.

Complete Sharing:

In complete sharing scheme, if type-1 has priority over type-2 traffic, then

$$P_{\text{loss}}(1) = \frac{\sum_{j=n+1}^N \sum_{k=0}^N P(j,k) (j-n)}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) j} \quad (5.14)$$

Since type-1 does not depend on type-2 traffic. In equation (5.14), N is the total number of sources into the multiplexer and n is the total number of cell buffers. Type-2 traffic is however affected by the level of type-1 traffic and hence

$$P_{\text{loss}}(2) = \frac{\sum_{j=0}^n \sum_{k=n-j+1}^N P(j,k) (k-n+j) + \sum_{j=n+1}^N \sum_{k=1}^N P(j,k) k}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) k} \quad (5.15)$$

The first term on right hand side of equation (5.15) represents the type-2 cell loss probability when type-1 traffic is using less than or equal to the total number of cell buffers (n) so that type-2 traffic can use the rest of buffers. The second term represents the case when more than n of type-1 sources are transmitting simultaneously. All the cell buffers are occupied by the type-1 traffic because it has the higher priority. Therefore, all the cells transmitted by the type-2 traffic are lost.

The cell loss probabilities for a 20 input multiplexer, each input source transmitting two high priority as well as low priority cells, are evaluated by equation [5.14] and equation [5.15] respectively. The results obtained are confirmed by simulation on QNAP2 (the values for very low $\approx 10^{-5}$ probabilities are not reliable in simulation because of simulation time limitations). The cell loss probabilities obtained are listed in Table [5.2] and plotted in Figure [5.8] for high as well as low priority cells. It should be noted that the higher priority cells have same probability of loss as that for one type of traffic given in Figure [4.6]. The lower priority traffic does not affect the cell loss probability of the high priority traffic. The low priority cells, as expected, have higher cell loss probability for same trunk capacity.

Trunk Capacity (total inputs=20)	Probability of Cell Loss			
	Analytical Results		Simulation Results (confidence level=0.95)	
	High Priority	Low Priority	High Priority	Low Priority
0	1.0	1.0	1.0	1.0
1	0.6464	0.9817	0.6367	0.9794
2	0.3588	0.9099	0.3457	0.9010
3	0.1676	0.7703	0.1576	0.7518
4	0.0653	0.5875	0.06022	0.5629
5	0.02120	0.4045	0.01937	0.3804
6	0.005747	0.2535	0.005348	0.2349
7	0.001302	0.1460	0.001282	0.1341
8	0.000247	0.07777	0.0002417	0.07144
9	0.3926e-4	0.03851	0.44028e-4	0.03570
10	0.52128e-5	0.01778	0.3873e-5	0.01677
11	0.5758e-6	0.007667	*	0.007457
12	0.5253e-7	0.003092	*	0.003144
13	0.3914e-8	0.001168	*	0.001268
14	0.2343e-9	0.4135e-3	*	0.4946e-3
15	0.11e-10	0.1373e-3	*	0.1718e-3
16	0.3897e-12	0.4278e-4	*	0.5186e-4
17	0.9804e-14	0.1251e-4	*	0.1006e-4
18	0.1561e-15	0.3436e-5	*	0.1161e-5

Table [5.2]: Cell Loss Probabilities for High and Low priority Cells for Multimedia sources with two traffic types in Complete Sharing Case. (* means that the results could not be obtained)

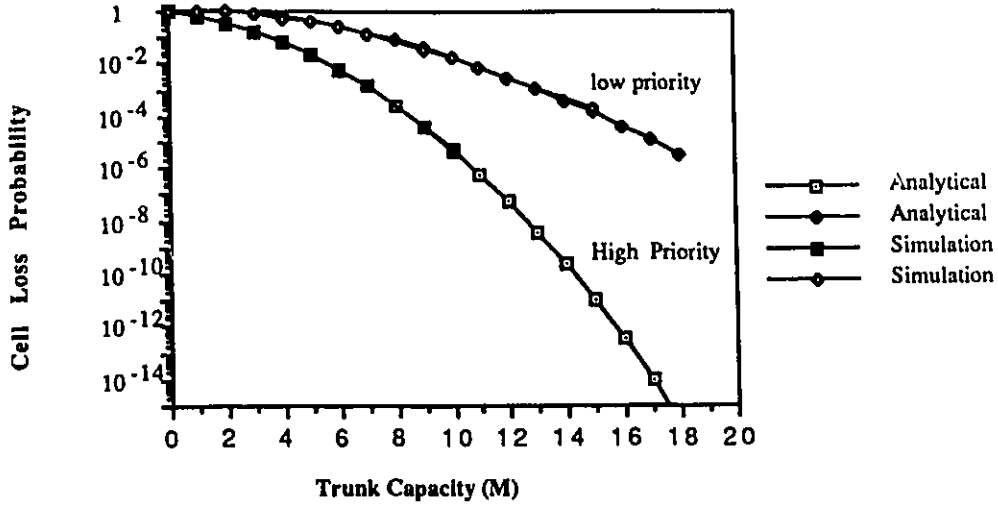


Figure [5.8]: Cell Loss Probabilities for High and Low Priority Cells in Multimedia Sources in Complete Sharing

Sharing With Minimum Allocation:

In sharing with minimum allocation scenario, n_1 cell buffers are reserved for type-1 traffic and n_2 cell buffers are reserved for type-2 traffic. The rest of buffers are shared according to allocation policy (priorities).

$$P_{\text{loss}}(1) = \frac{\sum_{j=n-n_2+1}^N \sum_{k=0}^N P(j,k) (j-n+n_2)}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) j} \quad (5.16)$$

and

$$P_{\text{loss}}(2) = \frac{\sum_{j=0}^{n_1} \sum_{k=n-n_1+1}^N P(j,k) (k-n+n_1) + \sum_{j=n_1+1}^{n-n_2} \sum_{k=n-j+1}^N P(j,k) (k-n+j) + \sum_{j=n-n_2+1}^N \sum_{k=n_2+1}^N P(j,k) (k-n_2)}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) k} \quad (5.17)$$

The first term on the right hand side of equation (5.17) represents cell loss probability for type-2 traffic when type one traffic is using less than or equal to its allocated n_1 cell buffers. Type-2 traffic can use the remaining $(n-n_1)$ buffers. The second term represents the case when type-1 traffic is using $(n-n_2)$ or less buffers so the type-2 traffic can use the remaining buffers. The third term gives the situation in which type-1 sources are using more than $(n-n_2)$ buffers so that type-2 traffic can only use its minimum allocated n_2 buffers.

The cell loss probabilities for SMA scheme are computed for a multiplexer with 20 inputs, transmitting two types of traffic. A total of 10 cell buffers are provided in the multiplexer. A traffic intensity of 12 CCS and a burstiness factor of 2.5 is assumed for either traffic type. The results for cell loss probabilities, computed analytically, are shown in Figure [5.9]. The "High Priority" and "Low Priority 1" curves show cell loss probabilities for high and low priority traffic respectively when buffers are allocated only to the low priority traffic as per SMA scheme. It should be noted that the high priority cell loss increases gradually and reaches unity when all (ten) buffers have been allocated to the lower priority traffic.

The "Low Priority 2" curve shows the cell loss for low priority traffic when buffers are allocated only to the higher priority traffic according to the SMA scheme. Again note that the low priority cell loss probability keeps increasing as buffers allocation to high priority traffic is increased. The cell loss probability is unity when all (ten) the buffers have been assigned to the higher priority traffic.

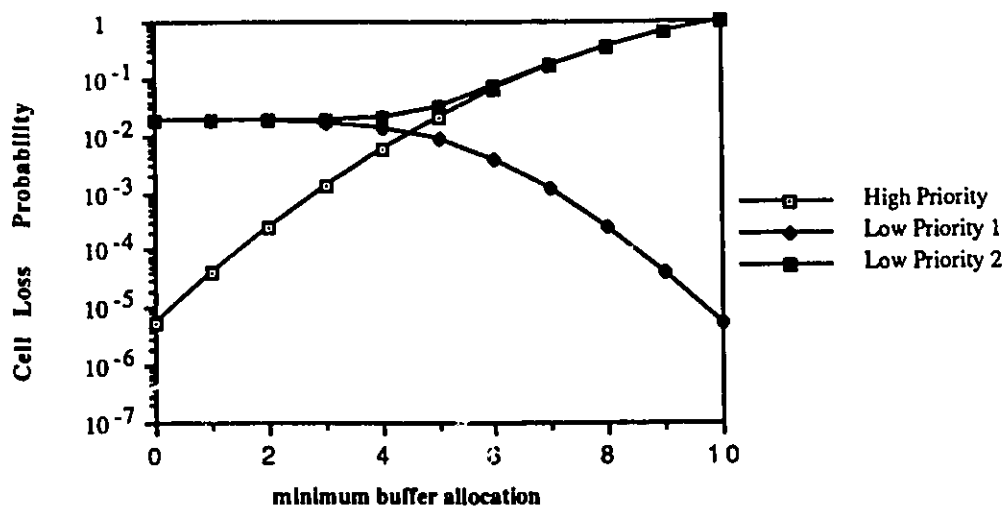


Figure [5.9]: Cell Loss Probabilities for Multimedia Sources using Sharing with Minimum Allocation scheme.

Sharing With Maximum Limit:

In sharing with maximum limit scenario, type-1 traffic can use up to n_1 buffers and type-2 traffic can use up to n_2 buffers. Otherwise all the buffers are shared.

$$P_{\text{loss}}(1) = \frac{\sum_{j=n_1+1}^N \sum_{k=0}^N P(j,k) (j - n_1)}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) j} \quad (5.18)$$

and

$$P_{\text{loss}}(2) = \frac{\sum_{j=0}^{n-n_2} \sum_{k=n_2+1}^N P(j,k) (k - n_2) + \sum_{j=n-n_2+1}^{n_1} \sum_{k=n-j+1}^N P(j,k) (k-n+j) + \sum_{j=n_1+1}^N \sum_{k=n-n_1+1}^N P(j,k) (k-n+n_1)}{\sum_{j=0}^N \sum_{k=0}^N P(j,k) k} \quad (5.19)$$

The first term right hand side of equation (5.19) represents cell loss probability for type-2 traffic when type-1 traffic is using $(n-n_2)$ or less buffers so that type-2 traffic can use n_2 buffers. The second term gives cell loss probability when type-1 traffic is using less than or equal to its maximum limit of n_1 but more than $(n-n_2)$ buffers. The type-2 traffic, with lower priority, can use the remaining buffers (out of a total of n). The third term represents the case when type-1 sources are transmitting more than n_1 cells so that type-2 traffic can only use $(n-n_1)$ buffers.

The cell loss probabilities for SML scheme are computed for a multiplexer with 20 inputs each transmitting two different types of traffic. A total of 10 cell buffers are provided in the multiplexer. A traffic intensity of 12 CCS and a burstiness factor of 2.5 is assumed for either traffic types. The results for cell loss probabilities, computed analytically, are shown in Figure [5.10]. The "High Priority" and "Low Priority 1" curves show cell loss probabilities for high and low priority traffic respectively when buffers are allocated only to the high priority traffic as per SML scheme. It should be noted that the cell loss probability for high priority traffic is unity when the its maximum buffer usage limit is zero. The cell loss probability for high priority traffic reduces gradually when the maximum buffer limit is increased. The cell loss probability for the low priority traffic increases correspondingly.

The "Low Priority 2" curve shows the cell loss for low priority traffic when buffers are allocated only to the lower priority traffic according to the SML scheme. Again note that the low priority cell loss probability is unity when the maximum limit is zero. The cell loss probability for low priority traffic decreases as its maximum buffer usage limit is increased.

Note that the cell loss probabilities are same in last two cases, namely SMA and SML, if n_1 and n_2 in SMA is replaced with $(n-n_2)$ and $(n-n_1)$ respectively in SML and vice versa.

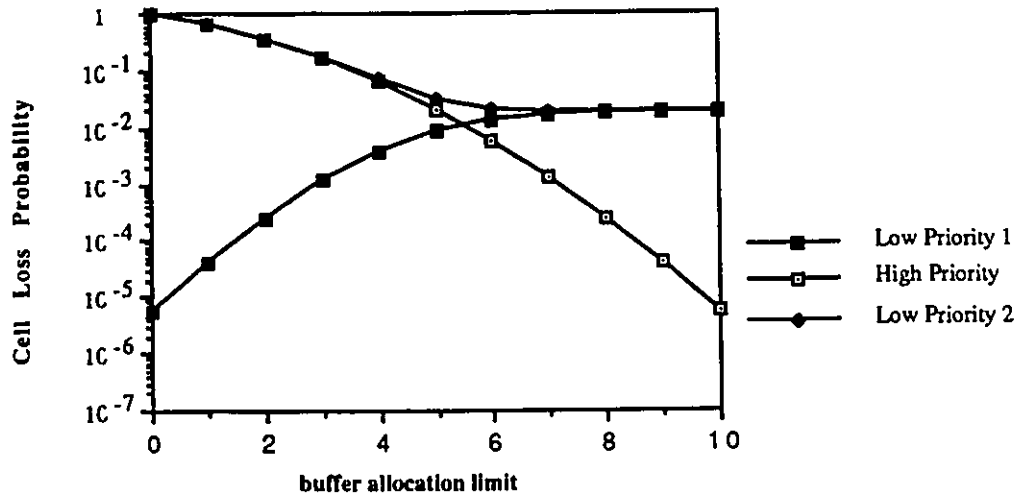


Figure [5.10]: Cell Loss Probabilities for Multimedia Sources using Sharing with Maximum Limit scheme.

Chapter 6

Conclusion

In this thesis, we have developed and solved a model for an ATM multiplexer. The input sources are assumed to be bursty in nature. The multiplexer samples each input periodically every T seconds. The period T is chosen such that each input source contributes either one cell or no cell to the multiplexer in T seconds. If the output trunk of the multiplexer has a capacity to transmit M cells in T seconds, the multiplexer buffers only M cells from the inputs in one scan (period). The rest of the cells received from the inputs are discarded. The cell discarding strategy should be fair to each input of the multiplexer. No cell queuing is done for the purpose of transmitting them when the traffic level subsides. Although queuing of cells may result in lower cell loss (cell blocking) probabilities, it introduces complexities in flow control, congestion control and buffer management. Queuing also introduces network delays which are variable from cell to cell. Such complexities must be avoided in ATM so that high throughput and better delay characteristics can be achieved.

A Markov chain model of the multiplexer is first developed and solved using the Gauss-Seidel method for solving large sparse linear equations. The results for state probabilities are then confirmed by using a Petri Net (GSPN) model for the multiplexer. Although the results matched, both of these methods are extremely slow and have large memory requirements. It is difficult to solve the model when the multiplexer inputs are increased beyond one hundred.

A closed form solution is then developed using Bayes Theorem. This closed form solution not only matches the results computed previously (Markov model and Petri Net model) for state probabilities but is quite fast and efficient. It can be used to solve a much larger model.

Cell loss (cell blocking) probabilities computed using the closed form solution are also confirmed using simulation (QNAP2). These cell loss probabilities are studied with variation in input traffic intensity (source occupancy) and burstiness factor. It is noted that the bandwidth gain decreases when traffic intensity is increased. The bandwidth gain increases when the burstiness factor is increased.

The model of the multiplexer developed earlier is then extended to include multiple traffic types. Inputs to the multiplexer are divided into traffic groups, each having different traffic characteristics. Different resource (cell buffer) sharing strategies are studied. The cell loss probabilities for Complete Sharing case computed from closed form solution and then they are

confirmed by simulation. One of the traffic type is assumed to have higher priority than the other. It should be noted that the cell loss probabilities for the higher priority sources are independent of the amount of lower priority traffic and are same as those obtained from the simple model with identical input sources transmitting only one type of traffic.

The model is also extended to include multimedia sources. Each source transmits more than one type of traffic. The results for cell loss probabilities for two traffic types is confirmed using simulation. Again, one type of traffic is assumed to have priority over the other. In this case also the results (cell loss probability) for higher priority traffic are independent of the amount of lower priority traffic.

From the results obtained, it is clear that the closed form model is quite accurate in computing the cell loss probabilities. For multimedia sources, a general equation for the state probability is:

$$P(x_1, x_2, \dots, x_n) = \sum_{i=\max(x_1, x_2, \dots, x_n)}^N \frac{\binom{N}{i} \left[\frac{\alpha}{\beta} \right]^i}{\left[1 + \frac{\alpha}{\beta} \right]^N} \prod_{x_j=x_1, x_2, \dots, x_n} \frac{\binom{i}{x_j} \left[\frac{\lambda_j}{\mu_j} \right]^{x_j}}{\left[1 + \frac{\lambda_j}{\mu_j} \right]^i} \quad (6.1)$$

$P(x_1, x_2, \dots, x_n)$ represents the probability that there are x_1 type-1 sources active, x_2 type-2 sources active and x_n type-n sources active. α and β are the call arrival rate and call service rate respectively. λ_j and μ_j are the packet arrival rate and packet service rate respectively for the j th traffic type.

It is found that the cell loss probability can be reduced by either increasing the trunk capacity or by reducing the number of input sources of the multiplexer. The bandwidth gain is reduced with either of these options. The cell loss probability becomes zero and the bandwidth gain becomes unity when the trunk capacity is equal to the sum of peak input source rates. This case is similar to the circuit switching. A compromise between the bandwidth gain and cell loss probability must be found according to the traffic requirements of the input sources. It is also noticed that the bandwidth gain is higher if the input sources have low traffic intensity or high burstiness.

If it is assumed that the performance of a network can be evaluated by evaluating the performance of individual network elements, the model developed here may be extended to compute the blocking probabilities of switching nodes and eventually the whole network. As shown in Figure [1.4b], a switching node may be modelled as an array of multiplexers. Each output of the switch consists of a multiplexer. All the inputs to the switch terminate at each of

these multiplexers. The output trunk of the multiplexer is connected to the corresponding output of the switch. A cell that arrives at an input of the switch goes into any of the multiplexers with certain probability (for homogeneous case, the probability is uniformly distributed). The cell loss (cell blocking) probability at the output of the switch is the same as the cell loss probability at that multiplexer.

Appendix A

SONET

A.1 Introduction

SONET (Synchronous Optical NETWORK) [AN0188, BALL89] is a newly adopted standard for optical interfaces. It was originally defined by Bellcore (Bell Communications Research) for inter-CO (Central Office) long haul network to transport information. The T1 committee of ECSA carried out most of the work towards definition of SONET as it stands today. The main goals of SONET were to provide:

- simple and flexible transport of existing North American digital hierarchy.
- compatibility with 64 Kbps digital voice channels.
- support for networks where signals pass through many jurisdictions that have different OAM procedures.
- adaptability to future broadband signals.
- ease of maintenance

In 1988, SONET was accepted by CCITT as a standard for optical fibre interfaces.

SONET defines a hierarchy of rates and formats for optical interfaces which are flexible enough to carry many different capacity signals. This is accomplished by defining a basic frame structure for standard optical signals, with a signal rate of 51.84 Mbps. The basic rate signal is called STS-1 (Synchronous Transport Signal - level 1). The STS-1 rate is chosen to accommodate the DS3 electrical signal at 44.736 Mbps. A byte interleave scheme has been defined at $N \times 51.84$ Mbps where $N (\leq 255)$ is an integer. Currently, values of 1, 3, 6, 12, 24 and 48 have been defined for N. The current maximum defined value of 48 for N gives a signal rate of 2480 Mbps. Higher rate signals of varying capacity can also be generated by linking the basic signals together, to carry signals of capacity greater than the basic signal.

The lowest internationally accepted signal rate for SONET is 150 Mbps. This signal is called Synchronous Transport Multiplex (STM). The STM interface resulted after a compromise between North American and European Carriers. It is suitable for carrying signals of both the T1 (1.544 Mbps) digital hierarchy used in North America and the CEPT (2.048 Mbps) digital hierarchy prevalent in Europe.

A.2 Synchronous Optical Equipment

The simplest optical network, based on SONET [CC0389, CC0489, CC0589, AN0289], is a point-to-point configuration shown in Figure [A.1a]. It consists of two Terminals, one on either side of an optical line. Each terminal in this configuration maps DSI (Digital Signal Interfaces e.g. DS1, DS3 etc.) signals into SONET format and vice versa. A terminal is always at the edge of the SONET network.

A single mode optical line can span about 50 km without the need of an optical signal regenerator. If the distance between the two Terminals is greater than that, one or more Thru-Repeaters (or Regenerators) may be used for signal grooming. A regenerator has two optical interfaces. It receives signal from one interface and transmits it on the other without altering the contents of the payload data. The regenerator only uses the section overhead (section A.6) of the SONET signal it regenerates. A simple network using a repeater is shown in Figure [A.1b].

An Add/Drop Multiplexer (ADM) interfaces with DSIs in addition to two optical interfaces of a regenerator. It maps DSI signals into the SONET signal that it regenerates. A network using ADM is shown in Figure [A.1c].

In a larger network, two or more low speed SONET signals may have to be multiplexed into a higher rate SONET signal. A switch that concentrates two or more lower speed SONET signals into a higher rate signal is called a concentrator or a HUB. A HUB, therefore, has more than two optical interfaces. Figure [A.1d] shows a network configuration using a HUB.

A Digital Cross-connect Switch (DCS) provides switching and cross-connection between two or more SONET signals. A DCS also terminates two or more optical lines. Figure [A.1e] shows a network configuration using DCS. The cross-connection may be done at STS-1 level or VT level (Virtual Tributary - described in section [A.8]).

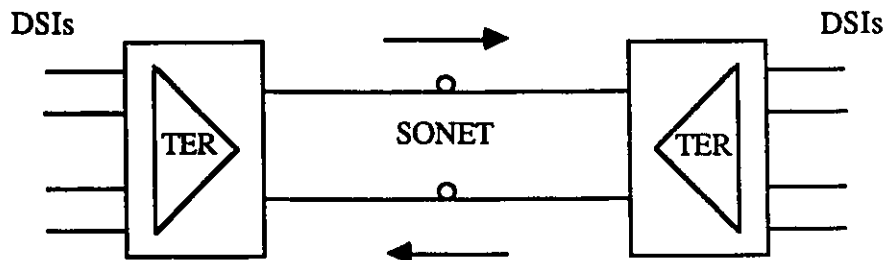


Figure [A.1a]: Point-to-point Configuration

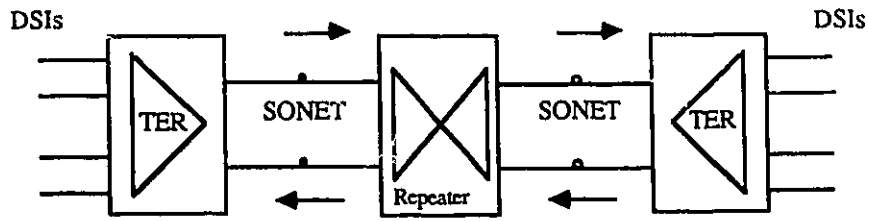


Figure [A.1b]: Repeater Configuration

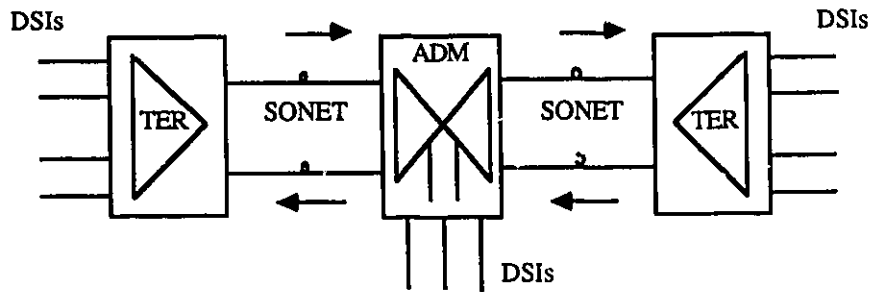


Figure [A.1c]: Add/Drop Multiplexer Configuration

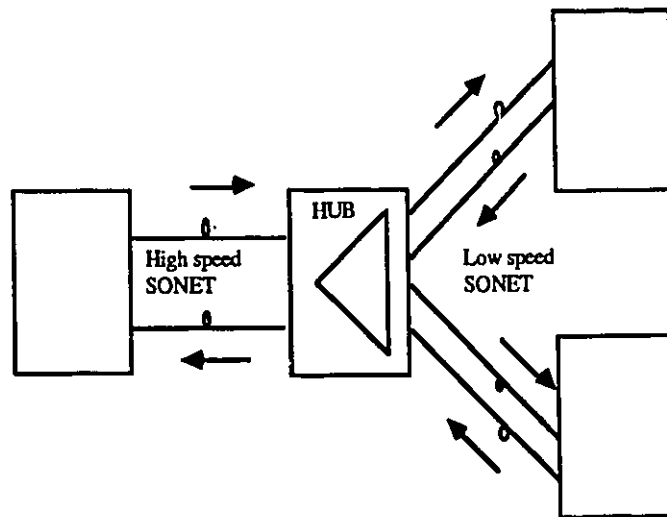


Figure [A.1d]: HUB Configuration

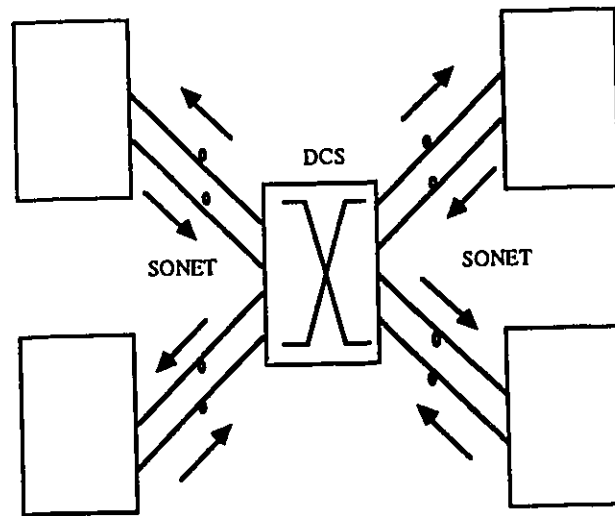


Figure [A.1e]: DCS Configuration

A.3 STS-1 frame

The basic frame for SONET signal hierarchy is called Synchronous Transport Signal - level 1 (STS-1). As shown in Figure [A.2], it can be drawn as 90 columns and 9 rows of 8 bit bytes. The STS-1 frame has a signal rate of 51.84 Mbps. The order of transmission of bytes is row by row, from left to right. This frame repeats itself every 125 μ sec. Therefore, each byte in an STS-1 frame corresponds to an equivalent channel of 64 Kbps and thus can transport a digital voice signal.

The First three columns of STS-1 frame contain Transport Overhead (TOH) and the remaining 87 contain the Synchronous Payload Envelop (SPE). Transport overhead includes Line Overhead (LOH) and Section Overhead (SOH). Out of $87 \times 9 (= 783)$ bytes of SPE, nine bytes are used by Path (or Payload) Overhead (POH) and the rest carries the SONET payload data. An STS-1 signal can carry one DS3 clear signal or other sub-DS3 signals such as DS1, DS1C, DS2 etc. An STS-1 can be converted, using electro-optical converters into an Optical Carrier - level 1 (OC-1) signal. The OC-1 is the basic optical signal used at SONET equipment and interfaces.

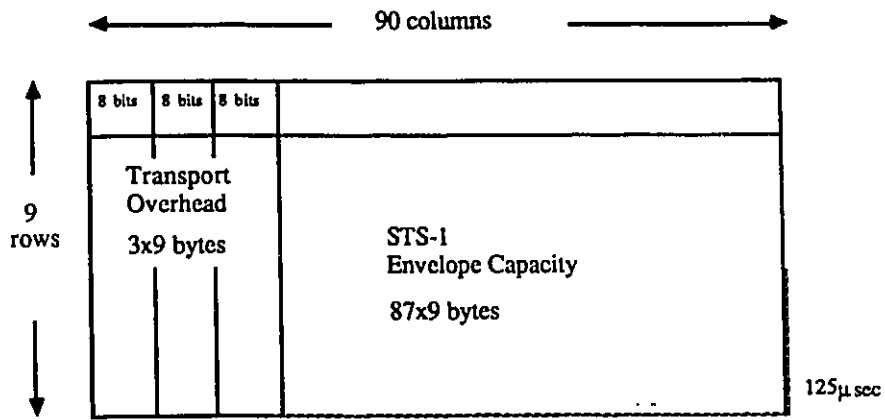


Figure [A.2]: STS-1 Frame

A.4 Frame Structure Of The STS-N Signal

Synchronous Transport Signal - level N (or equivalently OC-N signal) is formed by byte multiplexing (Figure [A.3]) of N frame aligned STS-1 signals. An STS-N signal can be used to carry higher rate payloads like broadband signals of 150 Mbps or 600 Mbps. All the section and line overhead channels of STS-1 #1 are used fully in an STS-N signal. Only BIP-8 channels and STS-1 id channels in TOH of other STS-1s are used.

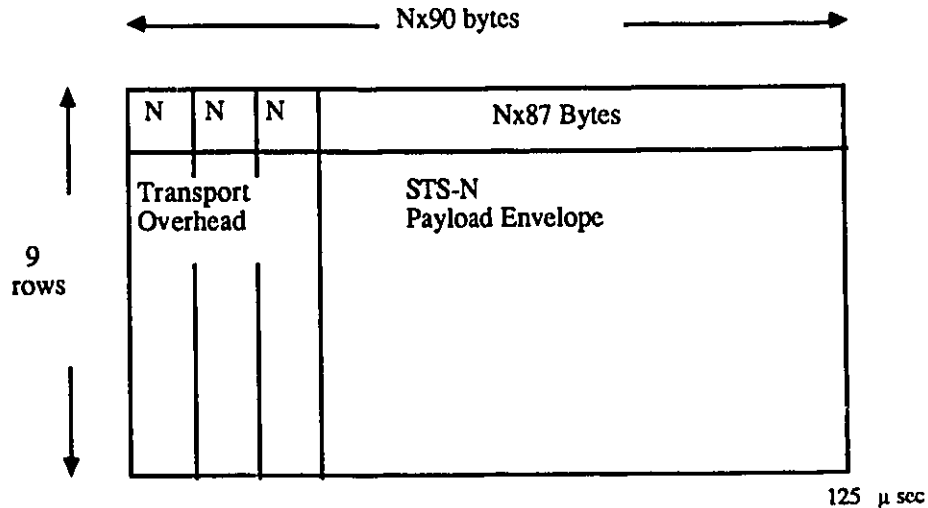


Figure [A.3]: STS-N Frame

A.5 STS-Nc Signal

If a broadband signal is to be carried in the SONET hierarchy, the phase and frequency of the STS-N signal must be locked so that the receiver at the far end can recover the broadband signal easily. The phase locking of the multiplexed frames is done by using concatenation indication in all the STS-1 pointers except the first one. When this concatenation indication is used, all the STS-1 pointers must maintain the same values. This is called concatenated STS-N signal or STS-Nc signal.

A.6 SONET Overhead Channels

Figure [A.4] shows the section, line and path overhead bytes and their relative positions. The first two bytes of the section overhead (SOH) are framing bytes that are used to synchronize the STS-1 frame. Then there is a STS-1 id byte followed by a Bit-Interleave Parity byte used for section error monitoring. An orderwire byte is reserved for craft communication. Then a byte is reserved for network/user applications. The last three bytes are used for section level data communications channel to carry maintenance and provisioning information. The complete protocol stack used for section data communication channels is shown in Figure [A.6]. The protocol stack is explained in section A.10. The section overhead is processed at all SONET equipment including repeaters.

Line overhead (LOH) is processed at all SONET equipment other than the repeaters. A repeater does not demultiplex the SPE and therefore does not terminate a line. Line overhead contains STS-1 pointer bytes H1 and H2. This pointer is used to achieve synchronization of signals in a plesiochronous environment. The pointer actually points to the start of the SONET payload with respect to the overall frame structure. This means that the SPE may start anywhere within the SONET frame structure (See Figure [A.5]).

Then there is a BIP-8 byte for line error monitoring, two bytes for Automatic Protection Switching (APS) message channel (both 1 + 1 and 1 by N protection is supported). The next 9 bytes are used for line data communications channel. After that, there are two bytes reserved for future growth and a byte for line orderwire. The protocol layers for line data communication channels are the same as those defined for the section data communications. The complete protocol stack for data communications is given in Figure [A.6].

The path overhead bytes are processed by the path (payload) terminating equipment. The path overhead includes a path BIP-8 for end-to-end error monitoring, a signal label byte to identify

the payload type, a path status byte to carry the maintenance signals, a multiframe alignment byte and three bytes for future growth. In a VT (Virtual Tributary) oriented SONET frame, there is overhead (VT pointer etc.) associated with VT transmission also.

Figure [A.7] shows the scope of SOH, LOH and POH in a simplified representation of a SONET network. The SOH terminates at all nodes, the LOH terminates at all nodes except the repeaters where as the POH terminates only at the Path Terminating Equipment (PTE).

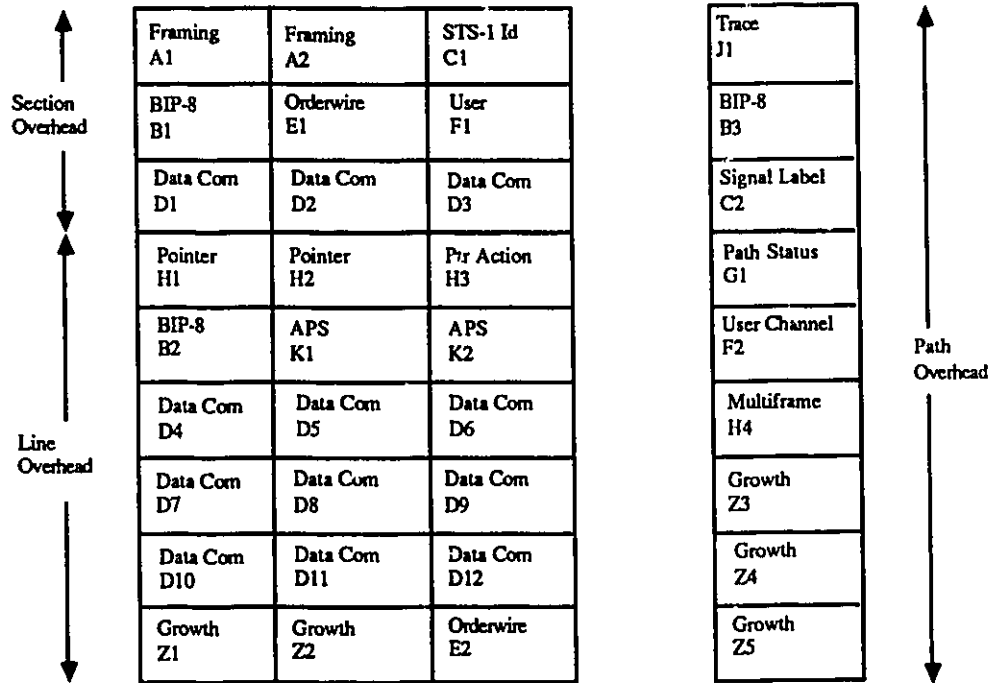


Figure [A.4]: Section, Line and Path Overhead Bytes

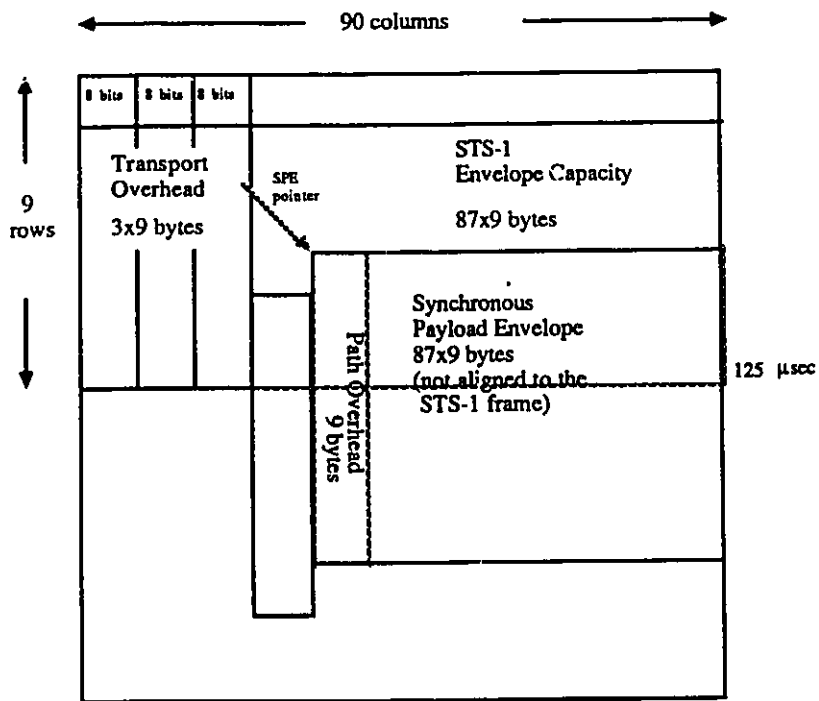


Figure [A.5]: SPE Pointer Function (STS-1 Frame)

Application Layer	CMISE ISO 9595-2, 9596-2 ROSE (X.219, X.229) ACSE (X.217, X.227)
Presentation Layer	X.216, X.226 ASN.1 (X.209)
Session Layer	X.215, X.225
Transport Layer	ISO 8073 (COTP)
Network Layer	ISO 8473 (CLNP)
Datalink Layer	LAPD (Q.920/21)
Physical Layer	Section/Line Data Com channels

Figure [A.6]: Section/Line Data Comm. Protocol Stack

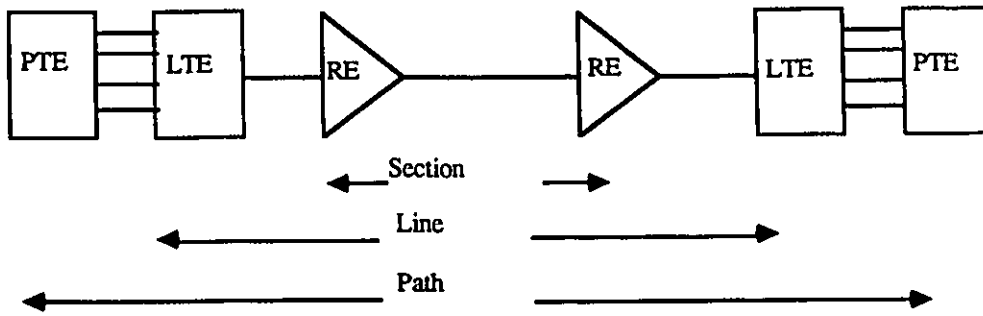


Figure [A.7]: SONET OverHead Boundaries

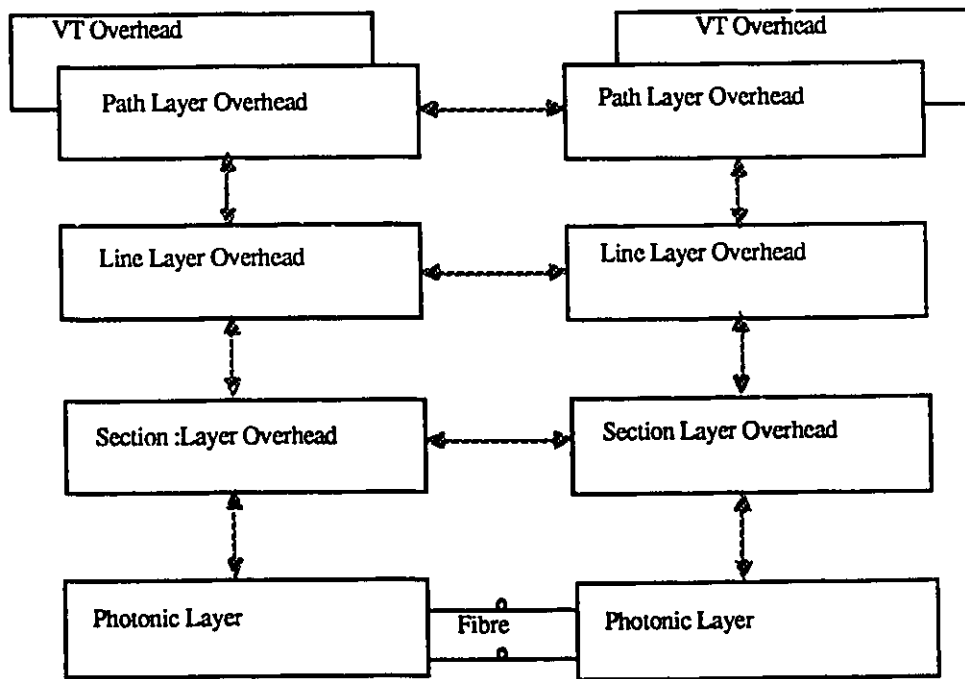


Figure [A.8]: SONET Overhead Hierarchy

A.7 Layered Overhead and Transport Functions

The overhead associated with SONET STS-1 frame structure follows a layered approach as shown in Figure [A.8]. At the bottom is the Photonic Layer which performs electro-optic conversion. The second layer is the Section Layer which provides framing, scrambling, section error monitoring and section data communication. The third layer is the Line Layer which multiplexes inputs from the path layer and adds line overhead to provide protection switching, line maintenance, line error detection etc. The line layer actually generates the STS frame. The top layer is the Path Layer which maps DSI signals and path overhead into STS-1 SPE.

Each layer communicates with peer equipment in that layer (horizontally) and hands over processed information to the next layer (vertically) as shown in Figure [A.8].

1. Photonic Layer

The photonic layer only deals with the transport of bits across the physical medium. It converts STS signal into OC signal. No overhead is associated with this layer. It transmits optical pulses to its peer entity.

2. Section Layer

It deals with transport of an STS-N frame across the physical medium using the photonic layer. It takes care of framing, section error monitoring and section data communication. Regenerators only use photonic and section layers.

It transmits STS-N frame and SOH (Section OverHead) to peer entities. It maps STS-Ns and SOH into pulses.

3. Line Layer

It provides synchronization and multiplexing for the path layer. Line overhead provides maintenance and protection. This overhead can only be accessed where the STS-N signal is formed.

It transmits SPE and LOH (Line OverHead) to peer entities. It maps SPEs and LOH into STS-N signal by multiplexing and payload justification and hands them to the section layer.

4. Path Layer

This layer transports services between path (payload) terminating equipment. It maps services (DSIs) into the format required by the line layer. It communicates over path overhead.

Path layer transmits services and POH (Path OverHead) to the peer entities. It maps services and POH into SPEs and hands them over to the line layer.

5. VT layer

This layer maps DSI services into VT oriented STS-1 frame. VT overhead (VOH) is associated with VT layer. Start of VT is indicated by VT pointer in VOH.

A.8 Multiplexing Synchronization Using Pointers

As mentioned before, the SONET payload is not locked to the frame structure. The payload floats with respect to the STS-1 frame. The start of the payload is indicated by STS-1 line overhead bytes (H1 and H2) shown in Figure [A.5].

If the payload and the STS-1 frame were locked, any frequency difference between the two would require to buffer 125 μ sec worth of data to phase align and slip (repeat a frame or discard a frame) frame.

In SONET, if the STS-1 payload data rate is higher, the payload pointer is decremented by one and the H3 overhead byte (pointer action) is used to carry data for one frame as shown in Figure [A.9a]. On the other hand, if the payload data is slow, the data byte immediately following the H3 byte is nulled for one frame and the pointer is incremented by one as shown in Figure [A.9b]. Thus, the synchronization can be maintained with a slight increase in complexity of processing the STS-1 signal. This processing can easily be implemented in hardware.

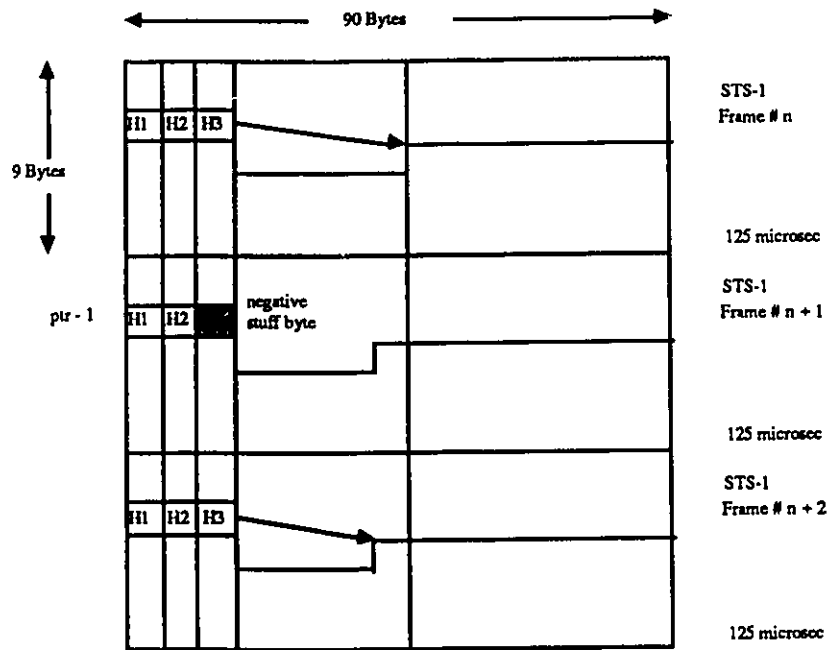


Figure [A.9a]: SPE Pointer Action (-ve Stuffing Byte)

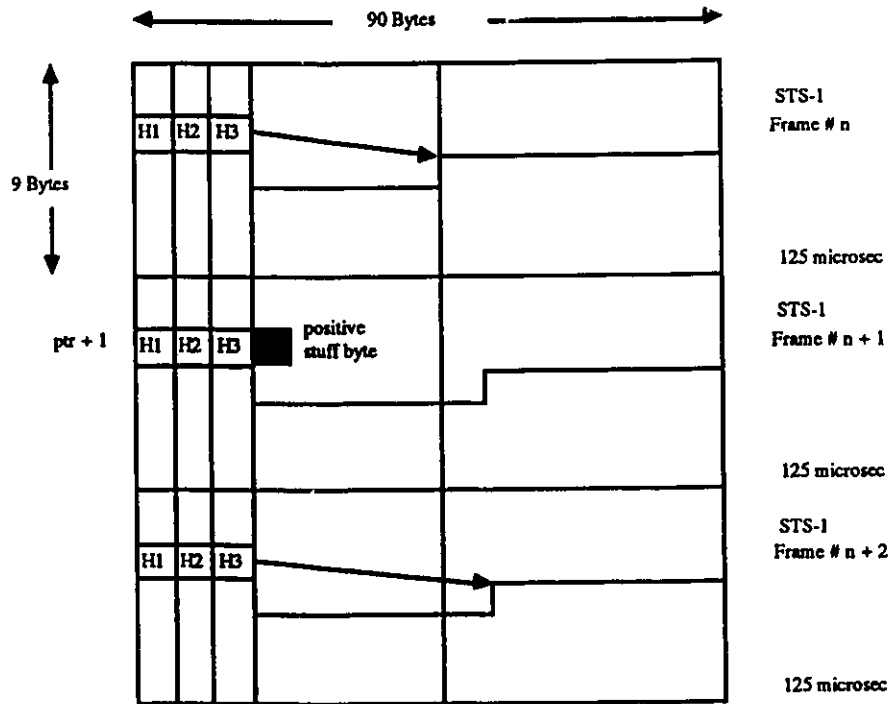


Figure [A.9b]: SPE Pointer Action (+ve Stuffing)

A.9 Sub-STS-1 multiplexing

In order to transport payloads whose rate is less than that of a DS3 signal, a VT (Virtual Tributary) structure is defined for an STS-1 SPE. Four different types (sizes) of VTs have been defined. A VT1.5 is 1.728 Mbps and it can carry a DS1 payload, a VT2 is 2.304 Mbps and it can carry a CEPT-1 payload, a VT3 is 3.456 Mbps and it can carry a DS1C payload where as a VT6 is 6.912 Mbps and it can carry a DS2 payload. A VT1.5 occupies 3 columns in an STS-1 SPE, a VT2 occupies 4 columns, a VT3 occupies 6 columns and a VT6 occupies twelve columns.

A VT structured STS-1 SPE is divided into 7 VT groups (Figure [A.10]). Each VT group occupies 7 columns and may contain 4 VT1.5s, 3 VT2s, 2 VT3s or 1 VT6. One column is used for path overhead and two columns are wasted in a VT-oriented SPE. A VT group may contain only one type of VTs but groups within same SPE may contain different types of VTs.

Two different modes have been defined for transporting payloads within a VT. A floating mode VT improves the transport and cross-connection of VT payloads. A floating VT works within a VT structure very much like a floating SPE in a STS-1 structure (Figure [A.11]). A VT pointer is used to point to the first byte of the VT. A locked mode VT improves the transport and cross-connection of DS0 signals by maintaining the phase and frequency of DS0 signals. An STS-1 SPE either carries all locked or all floating VTs.

A.10 Section/Line Data Comm. Channel Protocol Stack

The complete protocol stack for section and line data communications is shown in Figure [A.5]. It can be easily mapped to the ISO OSI model protocol layers. At the very bottom of the stack is the physical layer. The physical layer is provided by the section and line data communication overhead bytes. LAPD protocol is used as the second or the datalink layer. ISO 8473 ConnectionLess Network Protocol (CLNP) has been standardized as the third or the network layer. ISO 8073 Connection Oriented Transport Protocol (COTP) has been standardized as the fourth or the transport layer in the protocol stack. Session (X.215, X.225) layer, presentation (X.216, X.226, ASN.1 or X.209) layer and application (SMISE, ROSE, ACSE) have also been standardized. The details of network to application layer are still to be worked out. The protocol layers are flexible enough to support any type of OAM messaging.

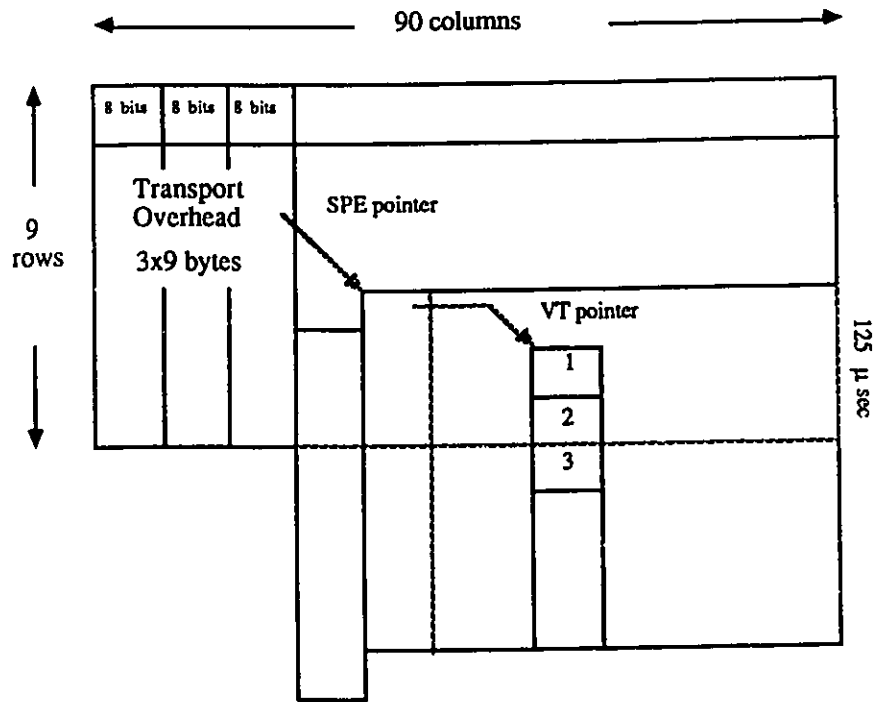


Figure [A.10]: VT Oriented STS-1 Frame

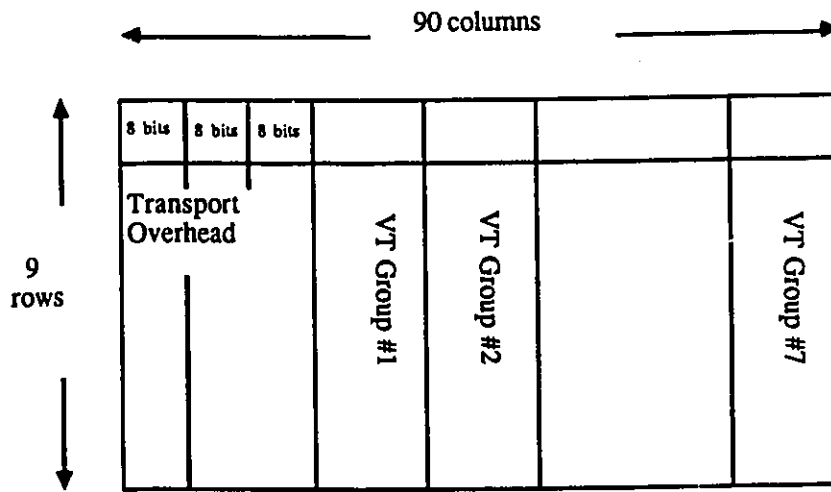


Figure [A.11]: Virtual Tributary Groups

Appendix B

Evaluation Of α , β , λ and μ

B.1 Call Arrival Rate α and Call Service Rate β

CCS is defined as number of hundred call seconds per hour, a source remains active. This means that the source remains off-hook for:

$$\begin{array}{ll} \text{or} & 100 \times \text{CCS} & \text{seconds/hour} \\ & 100 \times \text{CCS} / 3600 & \text{seconds/second} \\ \text{or} & \text{CCS} / 36 & \text{seconds/second} \end{array}$$

If a call stays off-hook for `holding_time` seconds and the off-hook time is assumed to be exponentially distributed stochastic variable, then the call service rate β may be evaluated as:

$$\beta = 1 / \text{holding_time} \quad (\text{B.1})$$

If α is the rate of call arrival per second and β is the call service rate, then from Figure [4.1]:

$$\alpha / (\alpha + \beta) = \text{CCS} / 36 \quad (\text{B.2})$$

Therefore, from equations (B.1) and (B.2):

$$\alpha = \text{CCS} / (\text{holding_time} \times (\text{CCS} - 36)) \quad (\text{B.3})$$

Note that this equation is different from the standard equation for α (i.e. $\alpha = \text{CCS} / (\text{holding_time} \times 36)$) used in telephony. This is because of the fact that we are considering finite number of sources at the input where as in telephony, infinite number of sources are assumed. In other words, in our model, if a input source is connected at a given time, another call can not arrive at that input until the current call has been serviced. In telephony, it is assumed that a source is receiving calls at the rate of α calls per second irrespective of the state of the connection at that time.

B.2 Packet Arrival Rate λ and Packet Service Rate μ

If an off-hook source transmits packets with an average length of `packet_length` seconds and an inter-packet length of `inter_packet_length` seconds and both if these parameters are assumed to be exponentially distributed, then λ and μ may be evaluated as:

$$\begin{array}{ll} & \lambda & = 1 / \text{inter_packet_length} \\ \text{and} & \mu & = 1 / \text{packet_length} \end{array}$$

Appendix C

Gauss-Seidel Method

C.1 Gauss-Seidel Method

We will not go into detail of describing the Gauss-Seidel method of solving large linear equations. Please refer to "Iterative Solution of Large Linear Equations" by DM Young [YOUN71]. It suffices here to say that equation [4.7] is iteratively used to compute new values of $P_{i,j}$ till the results converge. Initially, all states are assigned equal probabilities ($2.0/(N+1)(N+2)$). The new value is taken as the mean of the old value and the result of equation [4.7].

C.2 Program Developed for Gauss-Seidel Method

```
#include <stdio.h>

/* total number of inputs to the Multiplexer */
#define NUM_INPUTS      10

/* number of iterations */
#define ITERATIONS      1000000

/* accuracy of result required */
#define ACCURACY        0.000000000000001

/* CCS */
#define CCS              12.0

/* holding time in seconds */
#define HOLDING_TIME    180.0

/* packet length in seconds */
#define PACKET_LEN      1.2

/* inter-packet length in seconds */
#define INTER_PACKET_LEN 1.8

/* two dimensional markov chain */
double p[NUM_INPUTS+1][NUM_INPUTS+1];

main ()
{
    int          i, j;    /* index variables */

    double       a,      /* packet generation rate */
               b,      /* packet service rate */
               c,      /* rate of voice calls generated */
               d,      /* rate at which calls are serviced */
               sum;    /* sum of probabilities */
```

```

long          m;      /* iteration index */

/* Initialize all the probabilities to be equal. They add up to 1 */
for (j=0; j <= NUM_INPUTS; j++)
    for (i=j; i <= NUM_INPUTS; i++)
        p[i][j]=2.0/((NUM_INPUTS + 1)*(NUM_INPUTS + 2));

/* Initialize all variables */
a=CCS/(HOLDING_TIME*(36.0-CCS));
b=1.0/HOLDING_TIME;
c=1.0/INTER_PACKET_LEN;
d=1.0/PACKET_LEN;

/* update the probabilities ITERATION times */
for (m=0; m<ITERATIONS; m++) {
    for (i=0,sum=0; i <= NUM_INPUTS; i++) {
        for (j=0; j <= i; j++) {
            if(i==j) {
                if (i==0)
                    p[i][j]=((p[i+1][j]*(i+1)*b)/
                    (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                    p[i][j])/2.0;
                else if (i==NUM_INPUTS)
                    p[i][j]=((p[i][j-1]*(i-j+1)*c)/
                    (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                    p[i][j])/2.0;
                else
                    p[i][j]=((p[i][j-1]*(i-j+1)*c +
                    p[i+1][j]*(i+1)*b)/
                    (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                    p[i][j])/2.0;
            }
            else
                if (j==0 && i==NUM_INPUTS)
                    p[i][j]=((p[i-1][j]*(NUM_INPUTS-i+1)*a +
                    p[i][j+1]*(j+1)*d)/
                    (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                    p[i][j])/2.0;
                else if (j==0)
                    p[i][j]=((p[i-1][j]*(NUM_INPUTS-i+1)*a +
                    p[i][j+1]*(j+1)*d +
                    p[i+1][j]*(i+1)*b)/
                    (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                    p[i][j])/2.0;
                else if (i==NUM_INPUTS)
                    p[i][j]=((p[i-1][j]*(NUM_INPUTS-i+1)*a +
                    p[i][j-1]*(i-j+1)*c +
                    p[i][j+1]*(j+1)*d)/
                    (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                    p[i][j])/2.0;
                else
                    p[i][j]=((p[i-1][j]*(NUM_INPUTS-i+1)*a +
                    p[i][j-1]*(i-j+1)*c +
                    p[i][j+1]*(j+1)*d +
                    p[i+1][j]*(i+1)*b)/

```

```

                (i*b + j*d + (i-j)*c + (NUM_INPUTS-i)*a) +
                p[i][j])/2.0;
                sum += p[i][j];
            }
        }
    }

    /* Normalize the Probability Matrix */
    for (i=0; i <= NUM_INPUTS; i++)
        for (j=0; j <= i; j++)
            p[i][j] /= sum;
}

/* print parameters and probabilities */
printf ("a=%lf, b=%lf, c=%lf, d=%lf, iterations = %ld\n",
        a, b, c, d, m);
for (j=0, sum=0.0; j <= NUM_INPUTS; j++)
    for (i=j; i <= NUM_INPUTS; i++) {
        sum += p[i][j];
        if (p[i][j] > ACCURACY)
            printf ("%2.2d:%2.2d:%13.12le:%13.12le:%13.12le\n",
                    i, j, p[i][j], sum, 1.0-sum);
    }
}

```

Appendix D

Closed Form Solution

D.1 One Type Of Traffic Sources

```
#include <stdio.h>
#include <math.h>

#define NUM_INPUTS          40    /* Number of inputs */
#define ACCURACY            0.0000000000000001
#define CCS                 12.0  /* Hundred call seconds */
#define HOLDING_TIME       180.0 /* Call holding time in seconds */
#define PACKET_LEN         1.2   /* Packet length in seconds */
#define INTER_PACKET_LEN   1.8   /* Inter-packet length in seconds */

double    a,      /* Call arrival rate */
          b,      /* Call service rate */
          c,      /* Packet arrival rate */
          d;      /* Packet service rate */

double arry[NUM_INPUTS + 1]; /* Cell loss probability table */
double look[NUM_INPUTS + 1]; /* factorial table */
double abyb[NUM_INPUTS + 1]; /* power of a/b table */
double cbyd[NUM_INPUTS + 1]; /* power of c/d table */
double pabyd[NUM_INPUTS + 1]; /* power of (1 + c/d) table */
double pabyb; /* (1 + a/b) to the power of N */

/* function to compute the combinations of n taken r at a time */
double comb(n, r)
unsigned n;
unsigned r;
{
    return (look[n] - look[r] - look[n - r]);
}

/* function to compute yth power of a/b or c/d */
double pw (tab, y)
double *tab;
unsigned y;
{

    return(tab[y]);
}
```

```

/* Procedure to generate tables. All tables use log of the values */
generate_lookup()
{
    unsigned x;

    look[0] = 0;

    for (x=1; x <= NUM_INPUTS; x++)
        look[x] = look[x - 1] + log((double)x);

    for (x=0; x <= NUM_INPUTS; x++)
        abyb[x] = x*log(a/b);

    for (x=0; x <= NUM_INPUTS; x++)
        cbyd[x] = x*log(c/d);

    for (x=0; x <= NUM_INPUTS; x++)
        pcpyd[x] = x*log(1.0 + c/d);

    pabyb = NUM_INPUTS*log(1.0 + a/b);
}

/* Program to compute the cell loss probabilities */
main ()
{
    unsigned    k, x, m;    /* index variables */
    double      s,
               total,     /* expected cells sent by inputs per period */
               loss,      /* expected cell loss per period */
               sum;       /* sum of probabilities of no-loss states */

    a= CCS/(HOLDING_TIME*(36.0-CCS));
    b=1.0/HOLDING_TIME;
    c=1/INTER_PACKET_LEN;
    d=1/PACKET_LEN;

    printf("a=%le, b=%le, c=%le, nd=%le, ccs=%le, holding_time=%le\n",
           a, b, c, d, CCS, HOLDING_TIME);

    /* generate lookup tables */
    generate_lookup ();

    /* compute and print values of state probabilities */
    for (x=0,sum=0,total=0; x <= NUM_INPUTS; x++)
    {
        for (s=0, k=x; k <= NUM_INPUTS; k++)
            s += exp (comb(NUM_INPUTS, k) - pw(pcpyd, k) +
                    comb(k, x) - pabyb +
                    pw(abyb, k) + pw(cbyd, x));

        arry[x] = s;
        sum += s;
        total += s*x;
    }
}

```

```

        if (arry[x] > ACCURACY)
            printf("%3.3d:%15.14le:%15.14le:%15.14le\n",
                x, arry[x], sum, 1.0 - sum);
    }

    /* compute and print cell loss for different trunk capacities */
    for (m=0; m <= NUM_INPUTS; m++)
    {
        for (x=m+1,loss=0; x <= NUM_INPUTS; x++)
            loss += arry[x]*(x - m);

        printf("loss for m = %3.3d is %15.14le\n", m, loss/total);
    }
}

```

D.2 Two Types Of Traffic Sources

```
#include <stdio.h>
#include <math.h>

#define NUM_INPUTS_1      40    /* Number of type-1 inputs */
#define NUM_INPUTS_2      40    /* Number of type-2 inputs */

/* CCS for both types of sources */
#define CCS1               12.0
#define CCS2               12.0

/* holding time in seconds */
#define HOLDING_TIME1     180.0
#define HOLDING_TIME2     180.0

/* packet length in seconds */
#define PACKET_LEN1       1.2
#define PACKET_LEN2       1.2

/* inter-packet length in seconds */
#define INTER_PACKET_LEN1 1.8
#define INTER_PACKET_LEN2 1.8

double look[NUM_INPUTS_1 + NUM_INPUTS_2]; /* factorial table */

/* call arrival rate, call service rate, packet
   arrival rate and packet service rate for
   type_1 and type_2 sources */
double a1, b1, c1, d1;
double a2, b2, c2, d2;

/* tables for type-1 */
double arry1[NUM_INPUTS_1 + 1];
double abyb1[NUM_INPUTS_1 + 1];
double cbyd1[NUM_INPUTS_1 + 1];
double pabyd1[NUM_INPUTS_1 + 1];
double pabyb1;

/* tables for type-2 */
double arry2[NUM_INPUTS_2 + 1];
double abyb2[NUM_INPUTS_2 + 1];
double cbyd2[NUM_INPUTS_2 + 1];
double pabyd2[NUM_INPUTS_2 + 1];
double pabyb2;

/* function to compute combinations of n taken r at a time */
double comb(n, r)
unsigned n;
unsigned r;
{
```

```

    return (look[n] - look[r] - look[n - r]);
}

/* function to compute yth power of a variable */
double pw (tab, y)
double *tab;
unsigned y;
{
    return(tab[y]);
}

/* this procedure generates tables for type-1 sources */
generate_lookup1 ()
{
    unsigned x;

    for (x=0; x <= NUM_INPUTS_1; x++)
        abyb1[x] = x*log(a1/b1);

    for (x=0; x <= NUM_INPUTS_1; x++)
        cbyd1[x] = x*log(c1/d1);

    for (x=0; x <= NUM_INPUTS_1; x++)
        pabyd1[x] = x*log(1.0 + c1/d1);

    pabyb1 = NUM_INPUTS_1*log(1.0 + a1/b1);
}

/* this procedure generates table for type-2 sources */
generate_lookup2 ()
{
    unsigned x;

    for (x=0; x <= NUM_INPUTS_2; x++)
        abyb2[x] = x*log(a2/b2);

    for (x=0; x <= NUM_INPUTS_2; x++)
        cbyd2[x] = x*log(c2/d2);

    for (x=0; x <= NUM_INPUTS_2; x++)
        pabyd2[x] = x*log(1.0 + c2/d2);

    pabyb2 = NUM_INPUTS_2*log(1.0 + a2/b2);
}

/* Program to compute cell loss probabilities for
resource (buffer) complete sharing case */
main ()
{
    unsigned    k, x, y, n;    /* index variables */
    double      s,

```

```

        loss_l,      /* expected cells lost per period - low priority */
        loss_h,      /* expected cells lost per period - high priority */
        total_l,     /* expected cells sent per period - low priority */
        total_h,     /* expected cells sent per period - high priority */
        sum;

/* Initialize variables and tables for type-2 traffic */
a2= CCS2/(HOLDING_TIME2*(36.0-CCS2));
b2=1.0/HOLDING_TIME2;
c2=1.0/INTER_PACKET_LEN2;
d2=1.0/PACKET_LEN2;

for (x=1,look[0]=0; (x <= NUM_INPUTS_1) || (x <= NUM_INPUTS_2); x++)
    look[x] = look[x - 1] + log((double)x);

printf("a2=%le, b2=%le, c2=%le,\nd2=%le, CCS2=%le, holding_time2=%le\n",
        a2,  b2,  c2,  d2,  CCS2,  HOLDING_TIME2);

generate_lookup2 ();

for (x=0, total_l=0; x <= NUM_INPUTS_2; x++) {
    for (s=0, k=x; k <= NUM_INPUTS_2; k++)
        s += exp (comb(NUM_INPUTS_2, k) - pw(pcbyd2, k) +
                comb(k, x) - pabyb2 +
                pw(abyb2, k) + pw(cbyd2, x));

    arry2[x] = s;
    total_l += s*x;
}

/* Initialize variables and tables for type-1 traffic */
a1= CCS1/(HOLDING_TIME1*(36.0-CCS1));
b1=1.0/HOLDING_TIME1;
c1=1.0/INTER_PACKET_LEN1;
d1=1.0/PACKET_LEN1;

printf("a1=%le, b1=%le, c1=%le,\nd1=%le, CCS1=%le, holding_time1=%le\n",
        a1,  b1,  c1,  d1,  CCS1,  HOLDING_TIME1);

generate_lookup1 ();

for (x=0, total_h=0; x <= NUM_INPUTS_1; x++) {
    for (s=0, k=x; k <= NUM_INPUTS_1; k++)
        s += exp (comb(NUM_INPUTS_1, k) - pw(pcbyd1, k) +
                comb(k, x) - pabyb1 +
                pw(abyb1, k) + pw(cbyd1, x));

    arry1[x] = s;
    total_h += s*x;
}

/* Compute cell loss probabilities for both type of sources */
for(n=1;n<=NUM_INPUTS_1;n++) {
    for (x=0, loss_l=0; x<=n; x++) {
        for (y=n-x+1, sum=0; y<=NUM_INPUTS_2; y++)
            sum+=arry2[y]*(y-n+x);
    }
}

```

```

        loss_l+=arry1[x]*sum;
    }
    for (x=n+1; x<=NUM_INPUTS_1; x++) {
        for (y=1, sum=0; y<=NUM_INPUTS_2; y++)
            sum+=arry2[y]*y;

        loss_l+=arry1[x]*sum;
    }

    printf("Low Priority cell loss for m = %3.3d is %15.14le\n",
           n, loss_l/total_l);

    for (x=n+1, loss_h=0; x<=NUM_INPUTS_1; x++)
        loss_h += arry1[x]*(x-n);

    printf("High Priority cell loss for m = %3.3d is %15.14le\n",
           n, loss_h/total_h);
}
}

```

D.3 Multimedia Sources With Two Traffic Types

```
#include <stdio.h>
#include <math.h>

#define NUM_INPUTS          20    /* Number of inputs */

/* CCS for both types of sources */
#define CCS                  12.0

/* holding time in seconds */
#define HOLDING_TIME        180.0

/* packet length in seconds */
#define PACKET_LEN1         1.2
#define PACKET_LEN2         1.2

/* inter-packet length in seconds */
#define INTER_PACKET_LEN1   1.8
#define INTER_PACKET_LEN2   1.8

double look[NUM_INPUTS + 1]; /* factorial table */

/* call arrival rate, call service rate, packet
   arrival rate and packet service rate for
   type_1 and type_2 sources */
double a, b, c1, d1, c2, d2;

/* tables for type-2 */
double arry[NUM_INPUTS + 1][NUM_INPUTS + 1];
double abyb[NUM_INPUTS + 1];
double cbyd1[NUM_INPUTS + 1];
double pabyd1[NUM_INPUTS + 1];
double cbyd2[NUM_INPUTS + 1];
double pabyd2[NUM_INPUTS + 1];
double pabyb;

/* function to compute combinations of n taken r at a time */
double comb(n, r)
unsigned n;
unsigned r;
{
    return (look[n] - look[r] - look[n - r]);
}

/* function to compute yth power of a variable */
double pw (tab, y)
double *tab;
unsigned y;
{
    return(tab[y]);
}
```

```

}

/* this procedure generates tables for type-1 sources */
generate_lookup ()
{
    unsigned x;

    for (x=0; x <= NUM_INPUTS; x++)
        abyb[x] = x*log(a/b);

    for (x=0; x <= NUM_INPUTS; x++)
        cbyd1[x] = x*log(c1/d1);

    for (x=0; x <= NUM_INPUTS; x++)
        pcbyd1[x] = x*log(1.0 + c1/d1);

    for (x=0; x <= NUM_INPUTS; x++)
        cbyd2[x] = x*log(c2/d2);

    for (x=0; x <= NUM_INPUTS; x++)
        pcbyd2[x] = x*log(1.0 + c2/d2);

    pabyb = NUM_INPUTS*log(1.0 + a/b);
}

/* Program to compute cell loss probabilities for
resource (buffer) complete sharing case */
main ()
{
    unsigned    i, j, k, x, y, n;    /* index variables */
    double      s,
               loss_l,             /* expected cells lost per period - low priority */
               loss_h,             /* expected cells lost per period - high priority */
               total_l,            /* expected cells sent per period - low priority */
               total_h,            /* expected cells sent per period - high priority */
               sum;

    /* Initialize variables and tables for type-2 traffic */
    a= CCS/(HOLDING_TIME*(36.0-CCS));
    b=1.0/HOLDING_TIME;
    c1=1.0/INTER_PACKET_LEN1;
    d1=1.0/PACKET_LEN1;
    c2=1.0/INTER_PACKET_LEN2;
    d2=1.0/PACKET_LEN2;

    for (x=1,look[0]=0; x <= NUM_INPUTS; x++)
        look[x] = look[x - 1] + log((double)x);

    printf("a=%le, b=%le, c2=%le,\nd2=%le, CCS=%le, holding_time=%le\n",
           a,  b,  c2,  d2,  CCS,  HOLDING_TIME);

    generate_lookup ();

    for (j=0, total_l=0, total_h=0; j <= NUM_INPUTS; j++)
        for (k=0; k <= NUM_INPUTS; k++) {
            if (j > k) i=j; else i=k;

```

```

for (s=0; i <= NUM_INPUTS; i++)
    s += exp (comb(NUM_INPUTS, i) - pw(pcbzd1, i) - pw(pcbzd2, i) +
        comb(i, j) - pabyb + comb(i, k) +
        pw(abyb, i) + pw(cbyd1, j) + pw(cbyd2, k));

array[j][k] = s;
total_h += s*j;
total_l += s*k;
}

/* Compute cell loss probabilities for both type of sources */
for(n=0;n<=NUM_INPUTS;n++) {
    for (j=0, loss_l=0; j<=n; j++) {
        for (k=n-j+1, sum=0; k<=NUM_INPUTS; k++)
            sum+=array[j][k]*(k-n+j);

        loss_l+=sum;
    }

    for (j=n+1; j<=NUM_INPUTS; j++) {
        for (k=1, sum=0; k<=NUM_INPUTS; k++)
            sum+=array[j][k]*k;

        loss_l+=sum;
    }

    printf("Low Priority cell loss for m = %3.3d is %15.14le\n",
        n, loss_l/total_l);

    for (j=n+1, loss_h=0; j<=NUM_INPUTS; j++) {
        for (k=0,sum=0; k<=NUM_INPUTS; k++)
            sum+=array[j][k]*(j-n);

        loss_h+=sum;
    }

    printf("High Priority cell loss for m = %3.3d is %15.14le\n",
        n, loss_h/total_h);
}
}

```

Appendix E

QNAP2 Programs

E.1 One Type Of Traffic Sources

```
/CONTROL/ OPTION=NSOURCE;
/DECLARE/
  INTEGER    N = 40,           & number of inputs to the Mux
             m = 4,           & trunk capacity
             csema = 0,       & counting semaphore for number of cells queued
             ON = 1,
             OFF = 0,
             i,
             off_hook(N),     & flags indicating on_hook/off_hook
             pkt_tx(N);       & flags indicating packet transmission

  REAL loss = 0,              & loss for high priority
     total_cnt = 0,          & total number of cells sent
     c_tx=1.0,               & trunk service time for a cell
     t_sim=1000000.0,        & simulation interval
     tc_arr=150.0/0.002777778, & mean call arrival time
     tc_srv=150.0/0.005555556, & mean call holding time
     tp_arr=150.0/0.55555556, & mean packet arrival time
     tp_srv=150.0/0.83333333; & mean packet service time

  QUEUE     INTEGER    number;  & number refers to Mux input number

  QUEUE     mux,         & Multiplexer queue
            cell_src (N), & cell transmitting sources
            pkt_ctl (N),  & packet level control
            conn_ctl (N); & connection control

& only sets and resets the off_hook flags
/STATION/ NAME = conn_ctl (1 STEP 1 UNTIL N);
  TYPE      = SOURCE;
  SERVICE   =
  BEGIN
    & if source is off-hook, wait for call holding
    & time and put it on-hook.
    IF off_hook(number) = ON THEN BEGIN
      EXP (tc_srv);
      off_hook(number) := OFF;
    END

    & if source is on-hook, wait for call arrival
    & time and put it off-hook.
    ELSE BEGIN
      EXP (tc_arr);
      off_hook(number) := ON;
    END;

  TRANSIT (OUT);
END;
```

```

& Only controls the packet tx on flags
/STATION/ NAME = pkt_ctl (1 STEP 1 UNTIL N);
  TYPE          = SOURCE;
  SERVICE       =
  BEGIN
    & if source is on-hook, turn the packet tx off.
    & if source is off-hook and packet tx is on, wait
    & for packet service time and turn the packet
    & tx off.
    IF off_hook(number) = ON THEN BEGIN
      IF pkt_tx(number) = ON THEN BEGIN
        EXP (tp_srv);
        pkt_tx(number) := OFF;
      END
    END

    & if source is off-hook but packet tx is off,
    & wait for the packet arrival time and
    & turn the packet tx on.
    ELSE BEGIN
      EXP (tp_arr);
      pkt_tx(number) := ON;
    END;
  END
ELSE BEGIN
  pkt_tx(number) := OFF;
  CST (c_tx);
END;

TRANSIT (OUT);
END;

```

```

& This is real cell transmitting source. The cells are transmitted
& only when the packet tx flag is on.
/STATION/ NAME = cell_src (1 STEP 1 UNTIL N);
  TYPE          = SOURCE;
  SERVICE       =
  BEGIN
    & if packet tx is on, wait for cell period and generate
    & a cell. The cell is sent to the Mux only if there
    & are less than m cells in the Mux queue.
    CST (c_tx);
    IF pkt_tx(number) = ON THEN BEGIN
      total_cnt := total_cnt + 1;

      IF csema < m THEN
        csema := csema + 1
      ELSE
        loss := loss + 1;
      END;
    END;

    TRANSIT (OUT);
  END;

```

```

& MUX clears its queue periodically every c_tx time
/STATION/ NAME = mux;

```

```

TYPE          = SOURCE;
SERVICE      =
  BEGIN
    CST (c_tx);
    csema := 0;
    TRANSIT (OUT);
  END;

/CONTROL/
TMAX          = t_sim;
ACCURACY     = mux;
EXIT         =
  BEGIN
    PRINT("Cell loss probability = ", loss/total_cnt);
  END;

/EXEC/
BEGIN
  FOR i := 1 STEP 1 UNTIL N DO BEGIN
    pkt_tx (i) := OFF;

    IF i <= N/2 THEN BEGIN
      IF i > N/4 THEN pkt_tx (i) := ON;

      off_hook (i) := ON;
    END
    ELSE
      off_hook (i) := OFF;

    pkt_ctl (i).number := i;
    conn_ctl (i).number := i;
    cell_src (i).number := i;
  END;
  SIMUL;
END;
/END/

```

E.2 Two Types Of Traffic Sources

```

/CONTROL/ OPTION=NSOURCE;
/DECLARE/
  INTEGER    N = 20,           & number of inputs to the Mux
             m = 2,           & trunk capacity
             csema_l = 0,     & counting semaphore for low priority cells
             csema_h = 0,     & counting semaphore for high priority cells
             ON = 1,
             OFF = 0,
             i,
             off_hook(N),     & source off-hook or call active flag
             pkt_tx(N,2);     & packet transmitting flag - transmitting cells

  REAL loss_h = 0,           & loss for high priority
             loss_l = 0,     & loss for high priority
             tot_hcnt = 0,   & total number of cells sent
             tot_lcnt = 0,   & total number of cells sent
             c_tx = 1.0,     & trunk service time for a cell
             t_sim = 1000000.0, & simulation interval

             tc_arr=150.0/0.002777778, & mean call arrival time
             tc_srv=150.0/0.005555556, & mean call holding time
             thp_arr=150.0/0.555555556, & mean packet arrival time
             thp_srv=150.0/0.833333333, & mean packet service time
             tlp_arr=150.0/0.555555556, & mean packet arrival time
             tlp_srv=150.0/0.833333333; & mean packet service time

  QUEUE     INTEGER    number; & number refers to Mux input number

  QUEUE     mux,
             cell_src (N,2),   & Multiplexer queue
             pkt_ctl (N,2),   & cell transmitting sources
             conn_ctl (N);    & packet level control
                               & connection control

& only sets and resets the off-hook flags
/STATION/ NAME = conn_ctl (1 STEP 1 UNTIL N);
  TYPE      = SOURCE;
  SERVICE   =
    BEGIN
      & if source is off-hook, wait for call holding
      & time and put it on-hook.
      IF off_hook(number) = ON THEN BEGIN
        EXP (tc_srv);
        off_hook(number) := OFF;
      END

      & if source is on-hook, wait for call arrival
      & time and put it off-hook.
      ELSE BEGIN
        EXP (tc_arr);
        off_hook(number) := ON;
      END;

    TRANSIT (OUT);
  END;

```

```

& Only controls the packet tx on flags
/STATION/ NAME = pkt_ctl (1 STEP 1 UNTIL N, 1);
  TYPE          = SOURCE;
  SERVICE       =
  BEGIN
    & if source is on-hook, turn the packet tx off.
    & if source is off-hook and packet tx is on, wait
    & for packet service time and turn the packet
    & tx off.
    IF off_hook(number) = ON THEN BEGIN
      IF pkt_tx(number, 1) = ON THEN BEGIN
        EXP (thp_srv);
        pkt_tx(number, 1) := OFF;
      END

      & if source is off-hook and packet tx is off,
      & wait for the packet arrival time and
      & turn the packet tx on.
      ELSE BEGIN
        EXP (thp_arr);
        pkt_tx(number, 1) := ON;
      END;
    END
  ELSE BEGIN
    pkt_tx(number, 1) := OFF;
    CST (c_tx);
  END;

  TRANSIT (OUT);
END;

```

```

& This is real cell transmitting source. The cells are transmitted
& only when the packet tx flag is on.
/STATION/ NAME = cell_src (1 STEP 1 UNTIL N, 1);
  TYPE          = SOURCE;
  SERVICE       =
  BEGIN
    & if packet tx is on, wait for cell period and generate
    & a cell. The cell is sent to the Mux only if there
    & are less than m cells in the Mux queue.
    CST (c_tx);
    IF pkt_tx(number, 1) = ON THEN BEGIN
      tot_hcnt := tot_hcnt + 1;

      IF (csema_l + csema_h) < m THEN
        csema_h := csema_h + 1
      ELSE
        IF csema_l > 0 THEN BEGIN
          csema_l := csema_l - 1;
          csema_h := csema_h + 1;
          loss_l := loss_l + 1;
        END
        ELSE
          loss_h := loss_h + 1;
      END;
    END;
  END;

```

```

        TRANSIT (OUT);
    END;

    & Only controls the packet tx on flags
    /STATION/ NAME = pkt_ctl (1 STEP 1 UNTIL N, 2);
    TYPE           = SOURCE;
    SERVICE        =
    BEGIN
        & if source is on-hook, turn the packet tx off.
        & if source is off-hook and packet tx is on, wait
        & for packet service time and turn the packet
        & tx off.
        IF off_hook(number) = ON THEN BEGIN
            IF pkt_tx(number, 2) = ON THEN BEGIN
                EXP (tlp_srv);
                pkt_tx(number, 2) := OFF;
            END

            & if source is off-hook and packet tx is off,
            & wait for the packet arrival time and
            & turn the packet tx on.
            ELSE BEGIN
                EXP (tlp_arr);
                pkt_tx(number, 2) := ON;
            END;
        END
    ELSE BEGIN
        pkt_tx(number, 2) := OFF;
        CST (c_tx);
    END;

    TRANSIT (OUT);
END;

```

```

    & This is real cell transmitting source. The cells are transmitted
    & only when the packet tx flag is on.
    /STATION/ NAME = cell_src (1 STEP 1 UNTIL N, 2);
    TYPE           = SOURCE;
    SERVICE        =
    BEGIN
        & if packet tx is on, wait for cell period and generate
        & a cell. The cell is sent to the Mux only if there
        & are less than m cells in the Mux queue.
        CST (c_tx);
        IF pkt_tx(number, 2) = ON THEN BEGIN
            tot_lcnt := tot_lcnt + 1;
            IF (csema_l + csema_h) < m THEN
                csema_l := csema_l + 1
            ELSE
                loss_l := loss_l + 1;
            END;
        END;

        TRANSIT (OUT);
    END;

```

```

    & mux clears its queue periodically every c_tx time
    /STATION/ NAME = mux;

```

```

TYPE          = SOURCE;
SERVICE      =
  BEGIN
    CST (c_tx);
    csema_l := 0;
    csema_h := 0;
    TRANSIT (OUT);
  END;

/CONTROL/
TMAX          = t_sim;
ACCURACY     = mux;
EXIT         =
  BEGIN
    PRINT("High priority cell loss probability = ", loss_h/tot_hcnt);
    PRINT("Low priority cell loss probability = ", loss_l/tot_lcnt);
  END;

/EXEC/
  BEGIN
    FOR i := 1 STEP 1 UNTIL N DO BEGIN
      pkt_tx (i, 1) := OFF;
      pkt_tx (i, 2) := OFF;

      IF i <= N/2 THEN BEGIN
        IF i > N/4 THEN pkt_tx (i, 1) := ON
        ELSE          pkt_tx (i, 2) := ON;

        off_hook (i) := ON;
        END
      ELSE
        off_hook (i) := OFF;

      pkt_ctl (i, 1).number := i;
      conn_ctl (i).number := i;
      cell_src (i, 1).number := i;
      pkt_ctl (i, 2).number := i;
      cell_src (i, 2).number := i;
    END;
  SIMUL;
  END;
/END/

```

E.3 Multimedia Sources With Two Traffic Types

```

/CONTROL/ OPTION=NSOURCE;
/DECLARE/
  INTEGER    N = 20,           & number of inputs to the Mux
             m = 2,           & trunk capacity
             csema_l = 0,     & counting semaphore for low priority cells
             csema_h = 0,     & counting semaphore for high priority cells
             ON = 1,
             OFF = 0,
             i,
             off_hook(N),    & source off-hook or call active flag
             pkt_tx(N,2);    & packet transmitting flag - transmitting cells

  REAL loss_h = 0,           & loss for high priority
             loss_l = 0,     & loss for high priority
             tot_hcnt = 0,   & total number of cells sent
             tot_lcnt = 0,   & total number of cells sent
             c_tx = 1.0,     & trunk service time for a cell
             t_sim = 1000000.0, & simulation interval

             tc_arr=150.0/0.002777778, & mean call arrival time
             tc_srv=150.0/0.005555556, & mean call holding time
             thp_arr=150.0/0.555555556, & mean packet arrival time
             thp_srv=150.0/0.833333333, & mean packet service time
             tlp_arr=150.0/0.555555556, & mean packet arrival time
             tlp_srv=150.0/0.833333333; & mean packet service time

  QUEUE     INTEGER    number;    & number refers to Mux input number

  QUEUE     mux,           & Multiplexer queue
             cell_src (N,2), & cell transmitting sources
             pkt_ctl (N,2),  & packet level control
             conn_ctl (N);   & connection control

& only sets and resets the off-hook flags
/STATION/ NAME = conn_ctl (1 STEP 1 UNTIL N);
  TYPE      = SOURCE;
  SERVICE   =
  BEGIN
    & if source is off-hook, wait for call holding
    & time and put it on-hook.
    IF off_hook(number) = ON THEN BEGIN
      EXP (tc_srv);
      off_hook(number) := OFF;
    END

    & if source is on-hook, wait for call arrival
    & time and put it off-hook.
    ELSE BEGIN
      EXP (tc_arr);
      off_hook(number) := ON;
    END;

  TRANSIT (OUT);

```

```

END;

& Only controls the packet tx on flags
/STATION/ NAME = pkt_ctl (1 STEP 1 UNTIL N, 1);
  TYPE          = SOURCE;
  SERVICE      =
  BEGIN
    & if source is on-hook, turn the packet tx off.
    & if source is off-hook and packet tx is on, wait
    & for packet service time and turn the packet
    & tx off.
    IF off_hook(number) = ON THEN BEGIN
      IF pkt_tx(number, 1) = ON THEN BEGIN
        EXP (thp_srv);
        pkt_tx(number, 1) := OFF;
      END

      & if source is off-hook and packet tx is off,
      & wait for the packet arrival time and
      & turn the packet tx on.
      ELSE BEGIN
        EXP (thp_arr);
        pkt_tx(number, 1) := ON;
      END;
    END
  ELSE BEGIN
    pkt_tx(number, 1) := OFF;
    CST (c_tx);
  END;

  TRANSIT (OUT);
END;

```

```

& This is real cell transmitting source. The cells are transmitted
& only when the packet tx flag is on.
/STATION/ NAME = cell_src (1 STEP 1 UNTIL N, 1);
  TYPE          = SOURCE;
  SERVICE      =
  BEGIN
    & if packet tx is on, wait for cell period and generate
    & a cell. The cell is sent to the Mux only if there
    & are less than m cells in the Mux queue.
    CST (c_tx);
    IF pkt_tx(number, 1) = ON THEN BEGIN
      tot_hcnt := tot_hcnt + 1;

      IF (csema_l + csema_h) < m THEN
        csema_h := csema_h + 1
      ELSE
        IF csema_l > 0 THEN BEGIN
          csema_l := csema_l - 1;
          csema_h := csema_h + 1;
          loss_l := loss_l + 1;
        END
      ELSE
        loss_h := loss_h + 1;
    END;
  END;

```

```

        TRANSIT (OUT);
    END;

    & Only controls the packet tx on flags
    /STATION/ NAME = pkt_ctl (1 STEP 1 UNTIL N, 2);
    TYPE           = SOURCE;
    SERVICE        =
    BEGIN
        & if source is on-hook, turn the packet tx off.
        & if source is off-hook and packet tx is on, wait
        & for packet service time and turn the packet
        & tx off.
        IF off_hook(number) = ON THEN BEGIN
            IF pkt_tx(number, 2) = ON THEN BEGIN
                EXP (tlp_srv);
                pkt_tx(number, 2) := OFF;
            END

            & if source is off-hook and packet tx is off,
            & wait for the packet arrival time and
            & turn the packet tx on.
            ELSE BEGIN
                EXP (tlp_arr);
                pkt_tx(number, 2) := ON;
            END;
        END
    ELSE BEGIN
        pkt_tx(number, 2) := OFF;
        CST (c_tx);
    END;

    TRANSIT (OUT);
END;

    & This is real cell transmitting source. The cells are transmitted
    & only when the packet tx flag is on.
    /STATION/ NAME = cell_src (1 STEP 1 UNTIL N, 2);
    TYPE           = SOURCE;
    SERVICE        =
    BEGIN
        & if packet tx is on, wait for cell period and generate
        & a cell. The cell is sent to the Mux only if there
        & are less than m cells in the Mux queue.
        CST (c_tx);
        IF pkt_tx(number, 2) = ON THEN BEGIN
            tot_lcnt := tot_lcnt + 1;

            IF (csema_l + csema_h) < m THEN
                csema_l := csema_l + 1
            ELSE
                loss_l := loss_l + 1;
            END;
        END;

        TRANSIT (OUT);
    END;

```

```

& mux clears its queue periodically every c_tx time
/STATION/ NAME      = mux;
  TYPE              = SOURCE;
  SERVICE           =
    BEGIN
      CST (c_tx);
      csema_l := 0;
      csema_h := 0;
      TRANSIT (OUT);
    END;

/CONTROL/
  TMAX              = t_sim;
  ACCURACY          = mux;
  EXIT              =
    BEGIN
      PRINT("High priority cell loss probability = ", loss_h/tot_hcnt);
      PRINT("Low priority cell loss probability = ", loss_l/tot_lcnc);
    END;

/EXEC/
  BEGIN
    FOR i := 1 STEP 1 UNTIL N DO BEGIN
      pkt_tx (i, 1) := OFF;
      pkt_tx (i, 2) := OFF;

      IF i <= N/2 THEN BEGIN
        IF i > N/4 THEN pkt_tx (i, 1) := ON
        ELSE          pkt_tx (i, 2) := ON;
        off_hook (i) := ON;
      END
      ELSE
        off_hook (i) := OFF;

      pkt_ctl (i, 1).number := i;
      conn_ctl (i).number := i;
      cell_src (i, 1).number := i;
      pkt_ctl (i, 2).number := i;
      cell_src (i, 2).number := i;
    END;
  SIMUL;
END;
/END/

```

Bibliography

- [AN0188] ANSI T1.105-1988, "American National Standard for Telecommunications - Digital Hierarchy - Optical Rates and Formats Specification".
- [AN0289] T1X1.5/89-190 Supplement to ANSI T1.105-1988, 2 Oct.1988.
- [ARMB86] Armbruster H, "Applications of Future Broadband Services in the Office and Home", IEEE J. Sel Areas Comm., 1986, Vol.4(4), pp. 429-437.
- [BALL89] Ballart R, Ching YC, "SONET: Now It's the Standard Optical Network", IEEE Communications Magazine, Mar. 1989, pp. 73-91.
- [BRAD68] Brady PT, "A Statistical Analysis of On-Off Patterns in 16 Conversations", Bell Syst. Tech J., Jan. 1968, pp. 8-15.
- [BULI87] Bulick S, Irven J, "Broadband Information Services: Concepts and Prototypes", IEEE Globecom '87, pp. 50.2.1-50.2.8.
- [CC0189] Meeting Report of Sub-working Party 8/1 ATM, CCITT Study Group VIII, Geneva, 19-30 June, 1989.
- [CC0289] CCITT Recommendation I.121, "Broadband Aspects of ISDN", Blue Book, Geneva, Switzerland, 1989.
- [CC0389] CCITT Recommendation G707, "Synchronous Digital Hierarchy Bit Rates", Blue Book, Geneva, Switzerland, 1989.
- [CC0489] CCITT Recommendation G708, "Network Node Interface for the Synchronous Digital Hierarchy", Blue Book, Geneva, Switzerland, 1989.
- [CC0589] CCITT Recommendation G709, "Synchronous Multiplexing Structure", Blue Book, Geneva, Switzerland, 1989.
- [CHEN89] Chen TM, Walrand J, Messerschmitt DG, "Dynamic Priority Protocols for Packet Voice", IEEE Journal on Selected Areas in Communications, June 1989, Vol.7, No.5, pp. 632-643.
- [COUD87] Coudreuse JP, Serval M, "Prelude: An Asynchronous Time Division Switched Network", Proc. IEEE Int'l Conf. on Comm. '87, Vol.2, pp. 769-773.
- [COUD89] Coudreuse JP, "ATM: Status of Definition and Discussion of Some Open Issues", Multimedia '89, Montebello, Quebec, 20-23 Apr. 1989.
- [ECKB88] Eckberg AE and Hou TC, "Effects of output buffer sharing on buffer requirements in an ATDM packet switch", Proceedings, IEEE INFOCOM '88, pp.. 5A.4.1-5A.4.8.
- [FITZ89] Fitzpatrick GJ, Munter EA, "Input-Buffered ATM Switch Traffic Performance", Multimedia '89, Montebello, Quebec, 20-23 Apr. 1989, paper 4.2.
- [FLAN88] Flanagan TPJ, Allen BE, Copley GC, "Application Benefits of SONET Networking", IEEE Globecom '88, pp. 30.3.1-30.3.5.

- [GIML85] Gimlett JL, Stern M, Vodhanel RS, Cheung NK, "Transmission Experiments at 560 Mb/s and 140 Mb/s Using Single Mode Fiber and 1300 nm LEDs", *Electronic Letters*, 1985, Vol.21, pp. 1.198-1.200.
- [GITM81] Gitman I, Hsieh WN, Occhiogrosso BJ, "Analysis and Design of Hybrid Switching Networks", *IEEE Trans. on Comm.*, Sept. 1981, Vol Com-29, No.9, pp. 1290-1300.
- [GONE87] Gonet P, Coudreuse JP, Serval M, "Implementing Asynchronous Transfer Mode Concepts: Main Results of the Prelude Experiment", *IEEE Globecom '87*, pp. 47.4.1-47.4.5.
- [GRUB81] Gruber JG, "Delay Related Issues in Integrated Voice and Data Networks", *IEEE Transactions on Comm.*, 1981, Vol Com-29, No.6, pp. 786-800.
- [HAYW87] Hayward G, Linnel L, Mahoney D, Smoot L, "A Broadband ISDN System Using Emerging-Technology Components", *Proc. Int'l Switching Symp '87*, Vol.3, pp. 597-601.
- [HLUC88] Hluchi MG, Karol MJ, "Queueing in High Performance Packet Switching", *IEEE J. on Sel. Areas in Comm.*, Vol.6, No.9, Dec, 1988, pp. 1587-1597.
- [KLEI75] Kleinrock L, "Queueing Systems, Volume 1: Theory", John Wiley & Sons, 1975.
- [KRAI85] Kraimeche B, Schwartz, "Analysis of Traffic Access Control Strategies in Integrated Service Networks", *IEEE Transactions on Communications*, Oct. 1985, Vol Com-33, No 10, pp. 1085-1093.
- [KRAI86] Kraimeche B, Schwartz, "Bandwidth Allocation Strategies in Wide-Band Integrated Networks", *IEEE Journal on Sel. Areas in Communications*, Sept. 1986, Vol Sac-4, No.6, pp. 869-878.
- [KRAI87] Kraimeche B, Schwartz, "A Channel Access Structure for Wideband ISDN", *IEEE J. on Sel Areas in Comm.*, Oct. 1987, Vol SAC-5, No.8, pp. 1327-1335.
- [LIPP87] Lippman A, Bender W, "News and Movies in the 50 Megabit Living Room", *IEEE Globecom '87*, pp. 50.1.1-50.1.6.
- [LUTZ87] Lutz KA, "Broadband ISDN, a Hybrid Network with Continuous and Packetized Information", *IEEE Globecom '87*, pp. 50.8.1-50.8.5.
- [MARK89] Mark JW, "User Access to Broadband ISDN", *Proc. Canadian Conference on Electrical and Computer Engineering*, 17-20 Sept. 1989, pp. 689-692.
- [MARS86] Marsan MA, Balbo G, Conte G, "Performance Models of Multiprocessor Systems", *Computer Systems Series*, The MIT press, Cambridge, Massachusetts, 1986.
- [MINZ89] Minzer SE, "Broadband ISDN and Asynchronous Transfer Mode (ATM)", *IEEE Communications Magazine*, Sept. 1989, pp. 17-24.
- [OREI88] Oreilly P, "The Case For Circuit Switching in Future Wide Bandwidth Networks", *ICC '88*, pp. 29.1.1-29.1.6.

- [PRYC89] Prycker MD, "Evolution from ISDN to BISDN: a logical step towards ATM", *Computer Communications*, Vol.12, No.3, June 1989, pp. 141-146.
- [RICH88] Richards PS, Wernik MR, Woodruff GM, "Congestion Control for Asynchronous Transfer Mode Networks", *IEEE Globecom '88*, pp. 203-207.
- [RIDE89] Rider MJ, "Protocols for ATM Access Networks", *IEEE Network*, Jan. 1989, Vol 3, No 1, pp. 17-22.
- [SEGU87] Seguin H, Flichy P, "Interactive Services in French Videocommunication Networks", *IEEE Globecom '87*, pp. 50.5.1-50.5.5.
- [SPEA88] Spears DR, "Broadband ISDN - Service Visions and Technology Realities", *Int'l J. for Analog and Digital Cabled Syst.*, Vol.1, No.1, pp. 3-18.
- [SPEA89] Spears DR, Minzer SE, "New Directions in Signaling for Broadband ISDN", *IEEE Communications Magazine*, Feb 1989, pp. 6-14.
- [TIS188] NTI, TIS1 Contribution, July 1988, "Performance and Technology Consideration in the selection of ATM Switch and Multiplex port size".
- [TENE81] Tenenbaum AS, "Computer Networks", Prentice-Hall, Inc., 1981.
- [TURN83] Turner JS, Wyatt LF, "A Packet Network Architecture for Integrated Services", *IEEE Globecom '83*, pp. 2.1.1-2.1.6.
- [TURN86] Turner JS, "New Directions in Communications", *IEEE Communications Magazine*, Oct. 1986, Vol.24, No.10, pp. 8-15.
- [WEIL88] Wei Li, You-Wei Yang, Zheng Hu, "Synchronous Composite Circuit Switching for Integrated Voice-Data Networks", *Proc. IEEE*, 1988, pp. 47.7.1-5.
- [WERN88] Wernik MR, "Architecture and Technology Considerations for Multimedia Broadband Communications", *IEEE Globecom '88*, pp. 21.2.1-21.2.5.
- [WERN89] Wernik MR, Fitzpatrick G, McNamara R, Seguin, M, "Supporting Multimedia Applications in Asynchronous Transfer Mode Networks", *Multimedia '89*, Montebello, Quebec, 20-23 Apr. 1989, paper 6.2.
- [WONG86] Wong K, Georganas ND, "Strategies for Optimal Capacity allocations in DAMA satellite communication system", *Computer Communications*, 1986, pp. 26-32.
- [WOOD88] Woodruff GM, Rogers RGH, Richards PS, "A Congestion Control Framework for High-Speed Integrated Packetized Transport", *IEEE Globecom '88*, pp. 7.1.1-7.1.5.
- [WOOD89] Woodruff GM, Kositpaiboon R, "Evaluation of ATM Network Performance", *Multimedia '89*, Montebello, Quebec, 20-23 April. 1989, paper 2.2.
- [WRIG89] Write DJ, To M, "A Characterization of Telecommunication Services in the 1990s", *IEEE Computer Society*, Apr. 1989, pp. 624-631.
- [YOUN71] Young DM, "Iterative Solution of Large Linear Systems", Academic Press, 1971.