Report for the degree of
Master of Engineering

Simulation and Performance of a Statistical Multiplexer in an ATM Network

Author Diarmuid Corry, B Eng
Supervisor Dr T Curran

Dublin City University

School of Electronic Engineering

23 September, 1993

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed ______________
Diarmuid Corry

Date 23/08/93

CONTENTS

Simulation and Performance of a Statistical Multiplexer in an ATM Network

Author Diarmuid Corry

Abstract

This report examines some of the issues arising in the implementation of statistical multiplexing in a broadband Integrated digital services network (B-ISDN) by analysis and simulation The B-ISDN concept is introduced and described A review of the current areas of research is given along with some of the important issues as they relate to telephone traffic The report then focuses on the problem of multiplexing voice traffic

A typical voice source is analysed and the traffic characteristics which result are described The concept of statistical multiplexing is introduced A review of the current literature studies relating to the problems of analysing multiplexed sources is given, with particular reference to the concept of cell level and burst level queues being separate and disparate components requiring different analytical approaches Several models are introduced including the 3-state model not previously described in the literature

The queue behaviour resulting from a large number of superposed lines is analysed as a simplified Markov process and the results are used to argue that it is not feasible to provide buffers for nodes which multiplex a large number of low intensity sources The problem of scaling small models up to realistic situations is discussed

An approach to simulating the problem is described along with algorithms for implementing the basic elements A series of results derived from the described simulation are presented and analysed The report concludes that statistical multiplexing is feasible, but with certain limits as to the type of traffic which can be supported

Acknowledgements

This report would not have been possible without the support and contributions of many I would like to thank Dr Tommy Curran for his guidance and support, Dmitri Botvich for introducing the 3-State Model and focusing the direction of the simulations Dr Nick Duffield gave insights into the scaling problem and Dr E Buffet provided support on Markov Theory Adrian MacFearraigh provided useful discussions John Murphy, although a pain to live with, was invaluable in bouncing ideas off and outlining flaws in my reasoning He also does an excellent job of proof-reading

For personal reasons, undertaking this project was far more than just an academic challange Apart altogether from his technical advice, I would like to thank Dr Curran for his support and confidence Stephen Neville showed me that life's boundaries are only in the mind - his going to Australia probably helped get this thesis finished more than anything else! Everybody at DCU, especially "the gang" have been more supportive than they can know

I thank Aaron, Fergal and Niall - my colleagues in ACRA CONTROL (Innovation in Telemetry, tel 2951264 for state of the art data acquisition links) - for their support I would also like to thank my mother Anne for her constant reminders not to overwork Never fear!

This project has been funded by the European Programme for Advanced Technology

1. Introduction and Scope

1.1 Introduction to the ISDN

The Integrated services digital network (ISDN) is the current generation of the traditional telephony network. It replaces the old analogue form of wide area network with an entirely digital implementation. Data is transmitted along the network in ones and zeros representing the original data form. Because all data can be represented in a digital form the ISDN offers more than just voice carrying capabilities. It is called an integrated network because data, regardless of it source, can travel along the same link, Thus voice, video and fax transmissions can all share the same network. This opens up a whole range of possibilities. Apart from the large range of services which can be carried, the digital implementation enables some intelligence to be built into the network which broadens the range of services which the network providers can offer the customer. These include call diversion, call waiting alert, free phone numbers and other services previously impossible. Mobile calls can integrate seamlessly since the data is transmitted digitally (albeit over a different type of link).

With the ISDN the aim of utilising the (almost) world-wide telephony network to provide everything from news, video, data and voice communications becomes realisable. The limit of today's network is the physical link. As technology improves the bandwidth of the link increases opening the range of services that can be carried. The new network is known as the Broadband Integrated Services Digital Network (B-ISDN).

The B-ISDN presents new challenges in network management, resourcing, control and cost/performance ratio. Work has been going on to address these problems over the last ten years but much research remains to be done. In particular the areas of error control, network policing, network resourcing and protocol management are under active consideration. A related problem is that of source characterisation, whereby efforts are being made to determine what parameters of the wide range of possible data source presented to the network, are important with respect to network performance.

The work detailed in this report has been carried out under the auspices of the European Sponsored Programme for Advanced Technology. It concentrates on the area of network resourcing and in particular on the problem of statistical multiplexing.

1.2 Statistical Multiplexing

A statistical multiplexer aims to maximise network utilisation by using all the available bandwidth all the time. This is possible because some sources do not utilise all their bandwidth all the time allowing other sources to share the same channel. A typical example is voice which is usually encoded with built in silence detection This means that no data is sent while the person is silent (Silent in this regard includes the space between syllables, words and sentences as well as the longer speech pauses and listening silences) Typically, a person is only speaking for 35-40% of the time that he/she is on a call This means that if we have 100 active lines coming into a multiplexer, only 35-40 of them will be active at any one time If we want to fully utilise the output channel of the multiplexer we could increase the number of inputs to 250 (100 x (100/40)) and on average only 100 of them will be active at any one time However, we still need to be able to deal with the situation when more than the average are active without noticeable loss of network quality The problem is to balance network efficiency against quality of service by varying the number of inputs or providing buffers in the multiplexer

The problem is complicated by the fact that the sources may not be homogeneous, that some of the sources may require differing bandwidth at different times and that there is a limit to how long a single node can delay a packet of information by buffering it These problems have not been overcome and the current set of CCITT standards for the B-ISDN do not permit statistical multiplexing (the total input capacity to a node must not be greater than the output capacity) Nevertheless, statistical multiplexing offers strong cost/performance advantages in the future and will almost certainly be implemented in some form

1.3 Objectives of Thesis

This report looks at a particular implementation of a statistical multiplexer by simulation and draws some conclusions with respect to the performance which can be achieved and the source parameters which most affect this performance

1.4 Report Structure

The report is divided into seven chapters This chapter provides an introduction and overview of the report

Chapter 2 looks at the broad picture The method of data transmission on a B-ISDN is described in detail An overview of all areas of research is given with a brief description of the solutions which are being investigated From this broad overview the rest of the report focuses on the issue of multiplexing

Chapter 3 provides a detailed analysis of a reference source (voice) which will be used to analyse multiplexer performance The method of encoding voice is discussed and a detailed picture of the resulting traffic is provided The multiplexing problem is defined, and a modelling concept is presented A Markov model for a source is introduced which forms the basis for discussions in chapter 4 A survey of current literature on the topic is presented and the range of analytic models being examined is discussed Two models in particular are discussed in some detail because they both address the two-state nature of the multiplexer system performance Finally a three state Markov model is described This chapter contains all the definitions for the terminology and variables which are used throughout the rest of the report

Chapter 4 is a detailed description of the simulation approach used It describes the source code in some detail in terms of an object oriented model

Chapter 5 analyses multiplexer queuing behaviour in detail The two components of multiplexer queues are described and the issue of large scale systems is discussed Utilising some of the models described in chapter 3 a simple equation is derived which gives an insight into burst mode queue behaviour Utilising simulation results, sub-burst mode queuing is discussed

Chapter 6 presents a series of results from simulation runs Some of the implications and explanations for the results are given The results show the system sensitivity to traffic load, buffer size, source characteristics and source mixtures The performance of the 3-state Markov model as compared to the "direct" simulation is analysed and is found to be similar

Finally, chapter 7 summarises the results and concludes with suggestions for further study

2. Transmission over Broadband Networks

2.1. Introduction

The focus of much recent work in telecommunications has been on Broadband Integrated Services Digital networks This chapter is a brief introduction to the B-ISDN and outlines some of the work that is taking place It provides an overview of the area within which the work of this project takes place

2.2. Transmission modes in B-ISDN

2 2 1 Broadband Integrated Services Digital Network

The evolution of telecommunications networks continues to take place Up to twenty years ago the network was analogue, with large electro-mechanical exchanges providing the switching functions Traffic carried on the network was exclusively voice, in the form of analogue signals with a maximum bandwidth of 4kHz Developments in digital coding techniques and the availability of technology to support it led to the gradual integration of digital transmission of data which worked in tandem with the analogue network As technology improved the scope of the new network was expanded to include data as well as voice transmission and the concept of an Integrated Services Digital network was developed (ISDN) [S]

Coupled to the evolution of the network is a global standardisation effort controlled by the Consultative Committee on Telegraphy and Telephony (CCITT) This has seen the gradual elimination of disparate networks and the provision of a set of global standards facilitating the linking of networks across the world

While ISDN is being implemented throughout the world, advances in technology have set the stage for the next generation of telecommunications networks The introduction of fibre-optic transmission media and improvements in VLSI switching technology has presented the capability of increasing the bandwidth of telecommunications networks dramatically, allowing a wider range of services to be provided Current digital network capabilities are typically about 64kbit/s which is adequate for speech, some electronic data and very low resolution visual signals The new technology allows transmission speeds of 155Mbit/s and beyond opening the door to video, HDTV and high speed data signals being transmitted, along with the traditional services [K] This opens a whole new market to the network providers and radically alters the face of telecommunications The new network is referred to as the Broadband Integrated Services Digital Network (B-ISDN)

The CCITT is the world-wide body responsible for defining and maintaining standards for telecommunications It is part of the International Telecommunications Union which is a UN treaty organisation The CCITT generally releases a new set of standards every four years, and has been working on digital telecommunications and ISDN since 1968 The relatively recent developments in B-ISDN mean that the standardisation procedure is still very much in progress

2 2 2 Transmission Modes

The range of services that B-ISDN can carry is potentially enormous Some services (such as data) have a clearly defined bandwidth requirement which is quite low while others (such as video) are "bandwidth hungry" The situation is further complicated by the fact that some sources have a variable bit rate and the peak bit rate requirements may be an order of magnitude or more above the average This presents several problems in terms of implementation Since all of these services share the same network it is desirable that bandwidth be allocated in an efficient manner It is also desirable that such allocation can be altered dynamically as traffic characteristics change For example, the cessation of a high-bandwidth call should allow the newly released capacity to be allocated to several low-bandwidth calls This benefits the network in improving cost-to-performance ratios and benefits the customer in that he can be charged only for the bandwidth he actually uses

All transmission requires a standard for dividing the available transmission bandwidth between the various services requiring that bandwidth Originally an exchange simply hard-wired one pair of low bandwidth copper wires to another pair for the duration of the call With increasing bandwidth and sophistication in technology various ways of multiplexing several calls onto one channel were devised A popular paradigm in telecommunications networks is known as Synchronous Transfer Mode (STM)

STM transmits data by allotting time slots within a larger frame to a particular channel Every subsequent frame therefore carries data for that channel in the same time slot, a channel is identified by its time slot Higher bandwidth requirements are met by allotting the appropriate number of time slots per frame to the channel This method suits digital transmission technology and fixed demand services (i e services with known and invariable bandwidth requirements) Using STM current ISDNs carry data in several predefined channels (normally a 16kbit/s channel and two 64kbit/s channels known as 2B + D)

The problem with this method is a lack of flexibility When extended to broadband networks the system would require an expansion of the number of channels and range of bandwidths available Because the overall available bandwidth is carved up into certain pre-defined channels a high bandwidth service could be "locked out" simply because the high-bandwidth allocation was already used, even though the lower bandwidth allocation was under-utilised and enough spare bandwidth was available This problem could be overcome by dynamic allocation but the processing overhead required at each node of the network would be very large

Further difficulties arise when considering some of the highly-bursty services which are predicted for broadband networks STM would cope with this by allocating a channel capable of handling the maximum bit rate possible from the source However, the bursty nature means that the channel would be under-utilised for a relatively large amount of time resulting in inefficient network usage To address these problems a new mode of transmission has been developed known as Asynchronous Transfer Mode (ATM)

In ATM usable transmission capacity is divided into fixed size cells Each cell consists of a header containing system information and a body containing data The header (5 bytes) contains channel and path identifiers as well as error checking codes and other system information The body contains 48 bytes of data The rate at which a source generates cells can be constant or variable The allocation of bandwidth to a particular source is dependent on the number of cells from a source accepted onto a channel Further, cells from the same source can occupy different physical channels allowing great flexibility in network utilisation This is permissible since each ATM cell has a header which could be used for routing information However, the addition of a header field increases the total overhead for network transmission

	STM	ATM/Broadband
Bandwidth	64kb/s	150-620Mb/s
Bandwidth Allocation	Fixed	Dynamic
Switching Speed	50-100ms	10ms
Bottle-necks	Link Bandwidth Transmission delay	Switch bandwidth Switching delay
Protocols	Up to layer 3	Layer 1

Table 2 1 *Comparison of STM and ATM*

There is some discussion about the most suitable transmission structure for ATM The two structures under current discussion are the Synchronous Optical Network (SONET) and the Asynchronous Time Division (ATD) approach [JS] Both systems propose carrying ATM cells in frames but the structure of the frame is different in each case

The SONET frame is proposed by the US as a transmission structure It would have a bit rate of 51 84 Mbit/s and repeat every 125μs The frame carries system information bits (the transport overhead) necessary for the secure transmission of the frame and the data payload, consisting of a number of ATM cells SONET has the following advantages

- It can carry ATM or STM payloads, making the transition to broadband easier

- Some high-bandwidth services could be switched into their own exclusive SONET frames making "high-level" circuit switching easier The desirability of this is still under debate

- SONET synchronous multiplexing schemes allow several ATM streams to be combined, building interfaces with higher bit rates than normally supported by ATM

SONET does not require a framing pattern since the overhead acts as a framer

An ATD frame consists solely of ATM cells Therefore a framing pattern is required in order to differentiate cells Current proposals are to fill unused cell slots with a pre-defined pattern for framing, or to force a framing pattern in the case of all cells being full This framing pattern makes 100% utilisation impossible All data switching information must be carried in the cell headers The Asynchronous nature of ATD simplifies the switching interface and provides a conceptual unity with ATM

Standards have been developed for both SONET and ATD and it remains to be seen whether one or the other will become the *de facto* standard through implementation An important single recent development has been the standardisation of the ATM cell format (described above) which was released in 1990 **[CI 361]** Coupled with this is the work addressing SONET and ATD **[CI 432]** The B-ISDN network protocol model has been defined **[CI 321]** Thus a framework has been defined within which the details of ATM implementation can be worked out

2.3. Areas of Research

2 3 1 Active Topics

B-ISDN presents many new problems for research The areas of prime interest are

- Error Control

- Policing/Flow Control

- Source Modelling

- Network Resourcing

These areas are briefly described in the following sections

2 3 2 Error Control

The speed of Broadband networks requires that a new approach be taken to error control

Traditionally error control has taken place on a link-to-link basis, i e each node performs error checking on received data If an error is detected that node then requests a re-transmission This may mean re-transmitting all packets beginning with the erroneous one, or it may just require re-transmission of the bad packet The latter requires extra processing at the node to re-order the received packets, the former takes time

As the speed of networks grow the number of packets "en route" when an error is detected increases which means that the number of packets to be re-transmitted grows too large to be economical Similarly, the extra processing required at each node for link-by-link checking slows down transmission

Research is taking place into which method of error control is appropriate for the high transmission speeds encountered in broadband nets **[BKTV]** It has been suggested that an edge-to-edge scheme with "block" error checking (as opposed to cell checking) could be the most efficient for ATM networks

2 3 3 Policing/Flow Control

The speed of broadband networks has increased the problems associated with congestion management and flow control The fact that many packets may be in transit between nodes at a given time has the effect of lessening the control feedback frequency, reducing feedback effectiveness and increasing instability

The classic policing method suggested is known as the "leaky bucket" approach In this method cells which arrive faster than the stated rate are buffered (the buffer may be 0 size) and released at a rate which agrees with the maximum stated rate If the buffer is full, arriving cells are lost **[T]** Other policing methods have been proposed but all share the same basic traits, unruly cells are dropped The effectiveness of the policing method is dependent on the control algorithm for the "bucket" and the chosen cut-off point for cell loss If the stated maximum rate is close to the peak then cell loss is low but so is network efficiency, if it is too low then network efficiency improves (more calls are allowed on channels of less bandwidth) but cell loss probability increases

A compromise has been proposed (**[GRV]**), coined the "virtual leaky bucket" which utilises the fact that an unruly source may not necessarily overload the system, i e excess calls may be carried if other bursty sources are below average rate at that time In this approach unruly cells are marked but passed into the network Later, the congestion management operation of the network will discard marked cells if and only if the network is congested This improves network efficiency while decreasing cell loss probability, but requires a more complex congestion management algorithm

There is some confusion over how to define congestion in an ATM network The use of statistical multiplexing increases the risk of transient congestion (and associated cell loss) This is because, with this type of multiplexing, the sum of the peak transfer rates of the transmitting services is generally (or permitted to be) greater than the total available bandwidth There is a real probability that enough of these sources may transmit a burst simultaneously to temporarily overload the system The effects of this are under review One critical aspect is determining the parameters that describe the burstiness of a source

Solutions to the problem of congestion control are being considered Most suggestions remove the need for packet feedback by assuming implicit feedback (i e no feedback implies congestion, or using some other parameter to determine that congestion is occurring) The favoured approach at present is service denial as part of a congestion avoidance scheme In congestion avoidance an attempt is made to operate the network at optimum efficiency (throughput = load) and avoid any queuing (throughput < load) (By contrast, a congestion control scheme allows queuing but attempts to prevent packet loss) A new customer is accepted based on a knowledge of the present, and predicted, load on the network over the requested route, and service is refused if congestion is thought likely By operating at maximum efficiency there is a certain congestion leeway which should be able to absorb transient overloads due to simultaneous bursts from statistically multiplexed sources

The area of traffic control is under active consideration by the standards committees There are many questions which remain to be resolved This area affects allocation of bandwidth to users based on stated traffic characteristics The exact format of these traffic characteristics has not yet been defined For the present bandwidth is being allocated on a peak requirement basis This is wasteful of bandwidth but is the best that can be done until a better understanding of traffic behaviour is achieved One implication of this is that it shall not be possible to implement statistical multiplexing in the current format of B-ISDN However, as a clearer picture emerges it is expected that some form of statistical multiplexing will be introduced

2 3 4 Source Modelling

In classical network modelling the number of customers N(t) using the network at a time t_0 has been modelled as a stochastic random variable Customers are assumed to arrive randomly, in a Poisson process, and call duration is modelled as a random variable with exponential distribution This leads to the Erlang formulae which are used to calculate the probability of blocking for network resourcing problems

This classical model cannot be extended to broadband sources Cells arriving from bursty sources do not behave as a Poisson process Video sources are difficult to model since the bit-rate, burst-length and cell correlation are dependent on the type of video source Small variations in a scene generate a different type of traffic from sudden scene changes due to the video-coding algorithms normally used Various flavours of Markov Processes have been proposed to deal with these

In general no satisfactory model has been developed to cope with variable bit rate sources

Studies have shown that network performance is very sensitive to alterations in the burstiness of the source so adequate modelling is essential Long bursts reduce the efficiency of statistical multiplexing The need to define a parameter (or two) which accurately reflects the bursty behaviour of the source is critical to the success of congestion control and network resourcing techniques

2 3 5 Network Resourcing

One of the main problems with implementing Broadband networks is trying to predict a traffic profile Many of the expected sources are not yet clearly defined, and those that are have not been adequately modelled The problem of network resourcing therefore requires new techniques to be developed

In ATM networks the processing at each node is greatly simplified (relative to STM) reducmg the node cost However, the introduction of high-capacity fibre-optic links with low error rates, high-speed and good reliability means that the transmission cost has decreased drastically The node cost therefore becomes relatively high and the aim of low cost network design is to reduce the node cost without affecting too adversely the network utilisation efficiency

It is expected that the bursty nature of sources will permit statistical multiplexing of ATM cells Various approaches have been taken to determine an algorithm which utilises this statistical approach In general, most success has been gained by dividing traffic into classes according to its characteristics and considering each class to have it's own, separate virtual network Research has shown that by taking account of this burstiness capacity requirements of a network can be significantly reduced relative to peak-bit-rate allocation methods A similar improvement in network efficiency can be obtained **[GRV], [SS]**

The main problem with managing the capacity is determining how bandwidth is allocated To do this efficiently certain parameters which provide the network with information about the source must be supplied The more parameters there are, the better the resources can be managed, but the more complex control becomes

The nature of ATM lends itself well to a simple manifestation of a virtual network using the concept of the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) m the ATM cell header **[SOT], [GMP]** There is a proposal that the VCI, VPI information could be place in the overhead section of the SONET frame but this leads to increased processing requirements at nodes Alternatively the information is left in the ATM cell and is processed at the ATM layer

The advantages of using the ATM header is that node processing is simplified The fact that the path information is contained in the header reduces the need for route table management at each node, improving throughput speed Dynamic reconfiguring of the network in event of congestion or failure of a link is relatively easy, and network efficiency is improved since virtual paths can have their bandwidth varied dynamically

Current standards work in this area is involved with defining the signalling systems and requirements for signalling in B-ISDN An important aspect of this is the concept of the virtual circuit (VC) and virtual path (VP) A virtual circuit is a connection between users, a virtual path is a bundle of VCs which are switched and routed together The protocol for switching and managing these is intended for release in 1992

2.4. Conclusion

There is a world-wide drive towards integrating a range of services on the telecommunications network To achieve this the B-ISDN network is being defined and refined Already there is a clear picture of how this network will carry traffic in terms of ATM cells and network protocols Many areas, such as statistical multiplexing, are still undefined however, and much work remains to implement these proposals

This report falls into the area of network resourcing and performance Eventually it is hoped to achieve greater network utilisation by taking advantage of statistical multiplexing Within the context of the standardisation procedure this is some way down the road The following chapters will focus in on this problem of statistical multiplexing and examine some of the issues which affect it

---

[S] M F Slana, Integrated Service Digital Network *Fundamentals of Digital Switching*, edited by J McDonald Plenum Press, 1990 New York

[K] L Kleinrock, "ISDN - The Path to Broadband Networks *Proc of the IEEE Vol 79 No* 2 Feb 1991

[JS] J Jungog and T Suda, Survey of traffic Control Schemes and protocols in ATM Networks",

[CI 361] CCITT Recommendations I 361, B ISDN ATM Layer Specification, 1990

[CI 432] CCITT Recommendations I 432, "B ISDN User Network Interface Physical Layer Specification", 1990

[CI 321] CCITT Recommendations I 321, B-ISDN Protocol Reference Model and its Applications" 1990

[BKTV] A Bhargava, J F Kurose, D Towsley, and G Vanleemput, "Performance Comparison of Error Control Schemes in High Speed Computer networks", *IEEE J Select Areas Commun* vol 6, Dec 1988

[T] J S Turner, "New Directions in Communications (or which way to the information age?), *IEEE Commun Mag* , vol 24, Oct 1986

[GRV] G Gallassi, G Rigolio and L Verri, Resouirce management and Dimensioning in ATM Networks", *IEEE Network Magazine,* May 1990

[SS] Y Sato and K Sato, Virtual Path and Link Capacity Design for ATM networks , *IEEE J Select Areas Commun* , Vol 9, No 1, Jan 1991

[SOT] K Sato, S Ohta, I Tokizawa, "Broab Band ATM Network Architecture Based on Virtual Paths , *IEEE Trans on Commun* , Vol 38, NO 8, August 1990

[GMP] M Gerla, J A S Monteiro, and R Pazos, Topology Design and Bandwidth Allocation in ATM Nets", *IEEE Jour on Sel Areas Commun* , Vol 7, No 8, Oct 1989

3. Statistical Multiplexing of bursty ATM Traffic

3.1. Introduction

In this chapter the central problems associated with multiplexing ATM traffic are described First, the type of traffic that the B-ISDN network will carry is described Then a detailed description of a particular class of traffic (voice) is given along with definitions for some of the basic terms used throughout this report This traffic is used as a reference in all the simulation work carried out in this project The concept of statistical multiplexing is introduced and the effects of combining numerous traffic sources Finally, the queueing process that arises in the multiplexer as a result of the source arrivals is described and some of the approaches taken to modelling this process

3.2. Source Characterisation

3 2 1 Types of Sources

B-ISDN networks must deal with a large variety of sources Apart from the standard voice sources there are data sources, video sources and combinations of these The behaviour of these sources falls into one of three possible categories

- **Constant Bit Rate (CBR)** These sources generate data at a fixed rate Since this data is being collected and packetised it is presented to the network in the form of ATM cells with a deterministic interarrival time These cells are delivered for as long as the source call is held, which is usually several minutes but could be in the order of hours or even days Examples are modem communications between computers and fax

- **On/Off Source** These sources alternate between an active and an inactive state While active they generate data at a constant rate but generate no data while inactive Thus cells arrive at the network in clusters with a deterministic interarrival time between cells in a cluster and a random interarrival time between clusters Examples of these kind of sources are interactive computers and voice with silence detection

- **Variable Bit Rate (VBR)** These sources generate data at a rate which varies over time The rate may vary continuously or between fixed quanta Cells from these sources arrive at the network with an interarrival time which varies according to the state of the source Video is an example of this kind of source where the data rate is directly proportional to the degree of change from frame to frame

The important category of **bursty sources** is used to describe any source where the data rate tends to vary in bursts Obviously the On/Off sources described above fall into this category

VBR sources whose data rate varies only between a few fixed quanta also fall into this category However, when the term bursty is used in this report it refers to on/off sources only, i e the bit rate in the off state is zero

The aim of statistical multiplexing is to utilise the silence periods in bursty sources to support another active source

3 2 2 Characterising Voice

A very important example of bursty traffic is given by voice with silence detection This is the common form of voice transmission used today and a detailed examination of the characteristics of this traffic illustrates much of the important features of bursty traffic

Human speech consists of bursts of activity (sounds) interspersed by silences Because of its importance to telephony the characteristics of speech have been closely examined and are well understood **[B1],[B2],[G],[Y]** The following terms have particular meaning in voice analysis

- **Talkspurt** An uninterrupted burst of speech containing no silences longer than the hangover time
- **Silence** A period of time in which no voice activity is detected
- **Hangover** The period of time after a talkspurt has ended required to determine whether a silence has begun
- **Fill-in** The minimum length of a silence between talkspurts Silences of duration less than the fill-in time are considered to be part of the talkspurt
- **Threshold** The level (in dBm) used to decide whether speech is present or not Sounds below this level are deemed to be silence, sounds above this level are deemed to be speech

The choice of hangover time in particular has a dramatic effect on the silence distribution Silences are of two types Firstly there is the inter-word or inter-sentence silence which is normally quite short and occurs often Then there is the silence which results from the conversant listening to the other person speaking These silences are normally very long but are not so frequent as the first kind There can be no silences shorter than the hangover time, therefore the longer this hangover time the less bursty the voice data becomes and the more important the listening silences become to the overall average silence length

The effects of fill-in and hangover mean that it is not possible to obtain an absolute measurement for the statistical distribution of speech It will vary according to the hangover, fill-in and threshold chosen, as well as from language to language It also differs between the sexes This means that when examining multiplexer behaviour it is important to understand its sensitivity or otherwise to the underlying traffic characteristics (Some of these issues are addressed in chapter 6)

The length of a talkspurt varies randomly Brady [B1,2] establishes a mean length of 1 34s while Gruber [G] settles on a mean of *169 7ms* Other research [Y] has come up with a mean of 436ms The voice model used in [SW] is based on unpublished work by Bell labs and uses a mean of 352ms This model has been adopted as a standard by others and is also used in this project as the reference source However all sources agree that the distribution can be modelled as a geometric distribution

As stated earlier the distribution of silence is composed of two elements, conversational silences and listening silences The former are short and plentiful, the latter long and infrequent Again the actual mean for silence duration varies widely according to the experimental approach, Brady measured 1 67s and Gruber 123 9ms (it should be noted that Gruber used no hangover time, thus his results showed a large number of very short pauses) The private work mentioned earlier uses a mean silence length of 650ms To account for the twin effects of pauses and listening silence a hypergeometric distribution can be used

Measurements on voice activity (percentage of time that a single conversant actually spends speaking) do not show the same variance as the other measurements In general it is reasonable to assume that the average rate will lie in the band 35 - 42%

3 2 3 Voice Packetisation

The arrival process of voice cells is dependant on the ATM cell characteristics, the sampling rate and network control factors

There are various techniques for encoding speech The most common approach is based on Pulse Code Modulation (PCM) In this system the speech time signal is sampled and the resulting discrete samples are coded into an 8-bit representation of the amplitude Thus the entire range of speech is quantised mto 256 discrete values (normally non-linearly to improve the range) With this system each sample takes up 8 bits, at 8kHz resulting in an encoding rate of 64kbit/s

However, speech samples show a strong autocorrelation and this has been used to develop adaptive systems In these systems the speech is sampled at a higher rate (normally 16kHz) and the actual magnitude is not recorded Instead the difference in magnitude, or the rate of change or some combination based on these parameters is recorded In the limit the required information can be sent in 1 bit (delta-modulation) and the encoding rate is 16 kbit/s Another variation of this technique is known as Adaptive Differential PCM (ADPCM) and various algorithms exist for this which result in anything from 8 to 2 bits per sample

There is no consensus on which technique will be used in ATM networks The CCITT has recommendations on a 64kbit/s and a 32 kbit/s ADPCM system **[CG 722], [CG 721]** The main effect that differing encoding techniques has is that the inter-arrival time between packets in a talkspurt varies The effect of this will be discussed below

Certain congestion control techniques may affect the packet arrival rate In particular one technique know as Cell Dropping (CD) may be implemented In this mechanism the most significant bits and least significant bits of each sample are split up and put in different packets This packet pair is then sent together If there is congestion the packet containing the LSBs can be identified and dropped by the congested node This allows a doubling of the throughput speed with a negligible effect on signal quality **[SKS]**

The effect of CD on cell arrivals is twofold Firstly the packets are transmitted in pairs Secondly the interarrival time of each pair is twice that of individual pairs where non-CD is used since it takes twice as long to "pack" the two cells

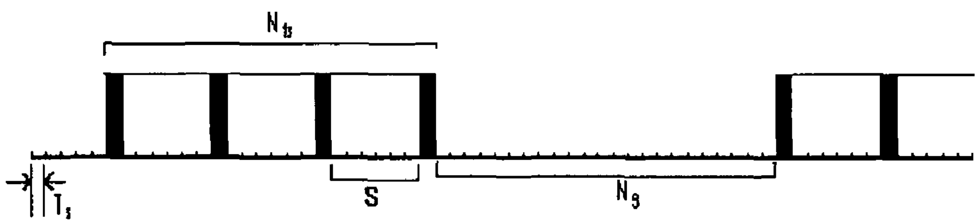


Fig 3 3 Cell arrivals during talkspurt

Fig 3 3 shows the cell arrivals during a talkspurt In this case there is a finite number of "ticks" in which a cell may arrive Each tick is equivalent to the transmission time of one cell which in turn depends on the bandwidth of the transmission medium A tick is also the smallest quantum of information that can be transmitted, the act of multiplexing comes from interlacing cells from different sources into the available ticks on the multiplexer output line The central problem is to determine how the statistical nature of the cell arrivals affect the queuing inside the multiplexer of cells for which there is no room at each discrete slot time Each tick is 424/C seconds long, where C is the bandwidth of the medium in bits/s (There are 424 bits in an ATM cell)

There is an assumption implicit in using a geometric distribution for talkspurt length that the arrivals over N lines are synchronised along ticks However, this assumption is reasonable since the length of a tick is far less than the timescale involved in the talkspurt packet interarrivals

The number of ticks between each cell arrival (S) is deterministic and depends on the number of bits per cell (B_c), and the encoding rate in bits/s

$$S = B_c / \text{(rate) (tick length)}$$

The first cell arrives S ticks after the start of a talkspurt, but the talkspurt is deemed to have ended immediately after the last cell arrival (i e part-filled cells are discarded) The number of cell arrivals in a talkspurt is random, with a geometric distribution Thus the number of ticks in each talkspurt is also random but is a multiple of S

The effect of CD can be considered if we note that the cells arrive in pairs (i e in adjacent time slots) and double the number of bits per cell to allow for the extended encoding time The resulting arrival pattern is very similar to the "ordinary" cell arrivals but with a different actual tick length

When determining the number of ticks in each silence it is noted the length of a silence is determined by the number of slots between cell arrivals Clearly

$$(S+1) <= \text{silence length} < \infty$$

From section 2 2 2 above we have that N_s is a random integer with a hypergeometric probability distribution

Rate	64 kbit/s	32 kbit/s
Bits per Cell	424 bits/cell	424 bits/cell
S	2454	4908
No of cells/talkspurt	53	27

Table 3 1 Typical Values

Table 3 1 shows some typical values for these numbers to give a sense of their relative sizes as well as the effect of some of the cell generation parameters discussed above For these calculations the mean talkspurt length has been assumed to be 352ms and the mean silence 650ms The transmission medium is assumed to be a 155 Mbit/s broadband network Therefore the tick length $T_s = 2\ 7\ \mu s$ approx This gives an activity rate r of 0 35 (i e 352/(650+352))

When considering arrivals from N sources it can be assumed that talkspurts can start synchronously (at the start of a time slot) since the length of a time slot is so small compared to the talkspurt length However, talkspurts on one channel can commence with a random phase respective to talkspurts on any other channel The deterministic nature of the cell arrivals means that the multiplexer output will be periodic and unvarying while the number of active (in talkspurt) lines does not change

3.3. Statistical Multiplexing

3 3 1 Multiplexer Gain

Typically in a local node of a B-ISDN network there are a number of low capacity inputs entering the node and a single high capacity output leaving it While the total capacity of the input lines is less than the output line capacity the multiplexer will be capable of handling all the inputs However, if some or all of the inputs are bursty then there is certain probability that only a subset of the inputs will be active at any one time This fact can be utilised to add more inputs so that the total input capacity is greater than the output capacity, but on average the number of active sources will be less than the output capacity The ratio of input capacity to output capacity is the multiplexer gain This is the basic concept of statistical multiplexing

The system can be fairly complex allowing a combination of source types and bit rates on the inputs To simplify the analysis the model dealt with here is a little different although retaining the fundamental properties required

Fig 3 4 depicts the model which is under consideration Voice signals are arriving on N high-capacity links, with only one voice source on each input link These are statistically multiplexed onto a single output link of the same capacity The output service time of the multiplexer is deterministic and is defined as one time unit (a tick) Cells are assumed to arrive synchronised on ticks All arrivals are queued on a First Come First Served (FIFO) basis and a single cell is removed from the queue each tick (if the queue is not empty)

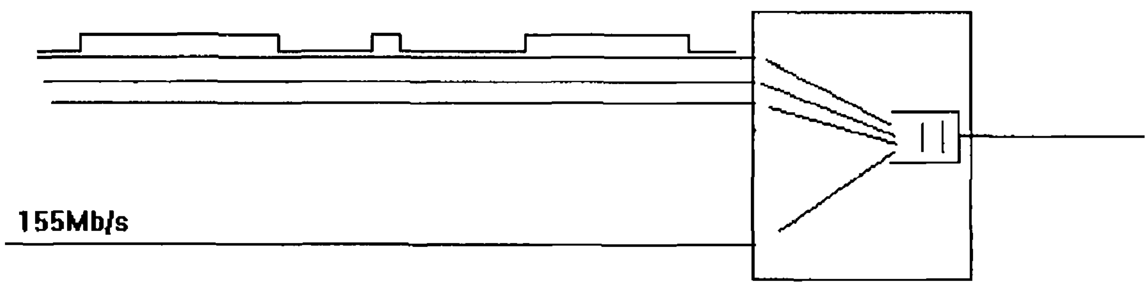
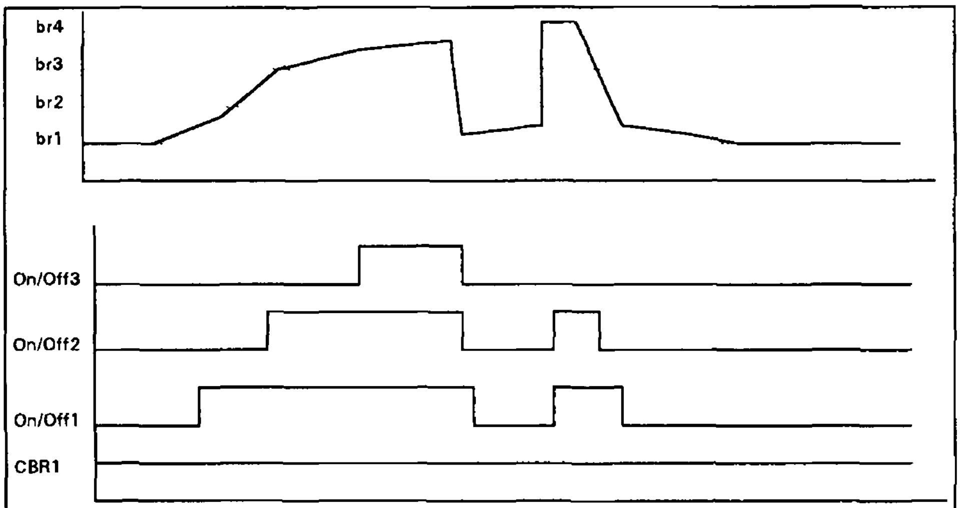


Fig 3 4 ATM Multiplexer model

This is conceptually the same as having N low capacity (i e 64kbit voice) lines Further the model may be viewed as a subset of a larger multiplexer, where there is a combination of bursty and non-bursty sources, by reducing the output capacity by the amount of bandwidth occupied by the CBR sources Finally, VBR sources can be integrated by treating them as several sources, a CBR source with a number of bursty On/Off sources associated with it Thus this simplified model still has broad general application



A simplistic view of how a CBR source and several On/Off sources can be used to represent a VBR source The VBR source in the top diagram is approximated by the superposition of the four sources in the bottom diagram The load presented by the combination of sources at any given moment will approximate that presented by the single VBR source There are limitations to this approach and it can only be used for multiplexer behaviour analysis, not for situations where the actual source is important such as traffic policing

It should be noted that here it is assumed that all N lines are carrying a call (on-line) In practise there will be significantly more than N links attached to the node since not all customers will be on the line at the same time (The probabilities covering this can be approximated using the classical Erlang formulae)

3 3 2 Modes of Operation

Each voice source is an on/off source which delivers packets with a deterministic interarrival time of S ticks Therefore the multiplexer has a "breathing space" of S-1 ticks between arrivals from a single source in which it can handle arrivals from other sources If we define the number of inputs active at a given time $n, \in [0, N]$ a discrete random variable then we have two distinct modes of operation

In the case where $n \leq S$ then the total input to the multiplexer is less than the output capacity This state is defined as **sub-burst mode** When $S < n \leq N$, more cells arrive at the input than can be handled in an interval of S ticks This is defined as **burst mode** The characteristics of queue behaviour differ radically between these two modes In general terms the queues arising in sub-burst mode are bounded while those in burst mode or not

The **gain** of the multiplexer in this model is given by N/S The problem of design is to maximise this gain while avoiding extended periods of burst mode operation This must be done under the quality of service constraints on the network which define a maximum cell transfer delay and probability of cell loss Since n is a random variable then while $N > S$ there is always a finite probability that the multiplexer will enter burst mode

3 3 3 Queue Components

The key constraints of cell delay and probability of cell loss are both closely related to the queue behaviour within the multiplexer If the multiplexer has a finite buffer then any cells arriving once the buffer is full will be lost It is therefore important to determine the probability of a buffer of a given size being full On the other hand, although increasing buffer size will reduce the cell loss probability, long buffers will lead to unacceptable delays in cell transmission The aim of the current study therefore is to characterise the queue behaviour knowing that this can be used to derive information for the parameters in which we are interested

To examine the characteristics of the queue behaviour it is necessary to examine the arrival process due to the superposition of n active sources There are two elements of this arrival process contributing to the queues, short term arrivals (micro-view) and long term arrivals (macro-view) In the short term, queues arise due to the arrival of several cells in any given tick Only one can be served and the excess cells are queued In the longer term queues arise when more than S cells arrive in a period S ticks long Since the arrivals are periodic with period S while there is no change in n, then as long as the system is in sub-burst mode the multiplexer is guaranteed to clear the workload presented to it at least once every S ticks The short term queues (referred to as **sub-burst mode queues**) are therefore bounded and can never be more than S-1 cells long Once in burst mode however, the queue will continue to build while the system remains in burst mode It will be shown later that burst mode tends to be persistent (relative to S), so the resulting queues can be very long Therefore the **burst mode queues** can dominate the queue behaviour

Queues are made up of the sum of these two components In general, the cell component is a rapidly changing (in time) "jitter" around a mean which is largely determined by the burst component

3.4. Characterising Queues in an ATM Multiplexer

3 4 1 Modelling the Arrival process

To derive analytic descriptions of the queue dynamics it is necessary to develop a model for the multiplexer and the arrival process A wide range of models have been proposed in the literature These fall into one of three categories

- Modelling the individual contributions of each cell arrival
- Modelling the superposed arrival streams
- Approximating the arrival process with a fluid model

The aim of these models is to capture the important characteristics of the arrival process In particular it is necessary to retain the correlations that exist in this stream since the queue is affected by the long term effects of small correlations [SW], [HL] The periodic nature of the cell arrivals produces strong negative correlation between successive interarrival times within a block of S ticks If looking at the multiplexed stream in a longer time scale then the intensity of arrivals varies as the number of active lines varies There exists positive correlation in the arrival intensities for adjacent periods

Since the phase of the input lines is randomly distributed with respect to each other, and the processes are stationary, it would seem that as N increases the arrival process for the superposed lines could be approximated by a Poisson process This is true for low traffic intensity (not many lines active) [K] However as traffic intensity increases the behaviour of the process becomes less and less Poissonian Also as the timescale over which the process is viewed increases the cumulative effect of the correlations mentioned above becomes more pronounced [SW] In fact a Poisson assumption seriously overestimates sub-burst queues, and leads to an underestimation of burst mode queues Therefore more complex models are required

The following sections outline the three modelling approaches

3 4 2 Cell Level Model

One way to view arrivals from an individual line is as an $S+1$-state Markov process

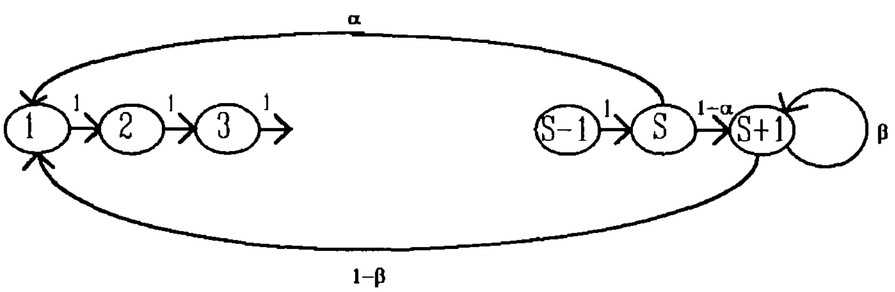


Fig 3 5 S+1 State Markov Model for a single line

Here state 1 is (arbitrarily) defined as the state whereby a cell is received from an active line State transitions occur every tick Once a cell has been received then the line will certainly not output a cell for the next S ticks so the system traverses the states 1 to S with probability 1

After S ticks the line will either emit another cell (if active) or not (if the line has switched to an inactive state)

The multiplexer can be characterised by three parameters n_t the number of cells in the buffer at time t, A_t the number of cells received by the multiplexer at time t, and b the size of the buffer Clearly

$$n_{t+1} = \min(b, \max(n_t + A_t - 1, 0)) \quad (3\ 4\ 1)$$

The state of the multiplexer at time t is fully described by the random variables $(n_t, x_{1\,t}, x_{2\,t}, \quad x_{N\,t})$, where $x_{i\,t}$ denotes the state of line i at time t Denote the probability distribution of this set of random variables by $\mathbf{P}_t(n,\underline{\mathbf{x}})$ then we can use 3 4 1 above to generate a recurrence relation

$$\mathbf{P}_{t+1} = \Gamma \mathbf{P}_t$$

where Γ is some linear operator built out of the transition matrix for the individual lines

The task of calculating Γ becomes very complex as N increases making this model impractical

3 4 3 Block Level Model

The above model can be greatly simplified by considering the behaviour of the line at a block level In this case the $S+1$-state model described above collapses into a 2-state model

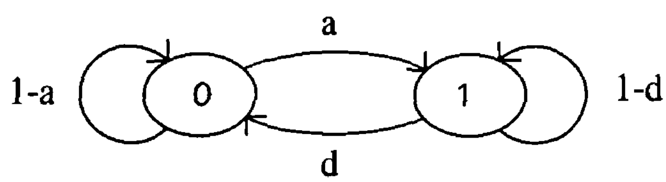


Fig 3 6 2-State Markov Model of a single line

Here state 0 is defined as the state whereby the line does not deliver a cell m the current block, state 1 the state where it does deliver a cell The transition probabilities **a** and **d** are functions of α, β and S

Using this model for each line the N-input multiplexer can be modelled as an $N+1$ state Markov process where state n is defined as the state whereby n lines are active (in state 1) A simplified diagram of this process is given in figure 3 7 Not all the state transition possibilities have been shown for clarity

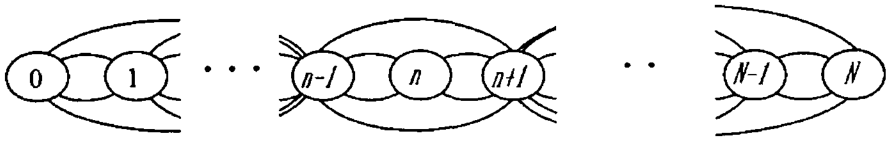


Fig 3 7 $N+1$ State Markov Model for Superposed Lines

The probability of going from state m to state n is basically binomial, although with contributions from all the states Each element of the NxN state transition matrix for this process is given by

$$p(m,n) = \sum_{\text{all possible } l} \mathbf{P}[l \text{ out of } m \text{ active lines remain active}]\ \mathbf{P}[n-l \text{ out of } N-m \text{ inactive lines become active}]$$

$$= \sum_{l=\max(0\ n-N+m)}^{\min(m,n)} \binom{m}{l}(1-d)^{l} d^{m-l} \binom{N-m}{n-l} a^{n-l}(1-a)^{N-m-n+l}$$

To fully characterise the multiplexer it is necessary to consider the queue length at any given time This increases the number of states in the Markov process (each state corresponds to n lines active with a queue length of b) so that the transition matrix becomes (bxN)x(bxN) The problem quickly becomes intractable

Since N is generally of the order of hundreds (or even thousands) this straightforward approach is not very useful It is also limited in that the sub-burst mode queues are not modelled A category of models called Markov Modulated Poisson Process (MMPP) models attempt to address these problems

3 4 4 MMPP Models

A popular representation of the arrival process is as a Markov Modulated Poisson process (MMPP) **[HL]** The general technique here is to model the system as having several states in which Poisson arrivals occur with the rate depending on the state of the system Ide **[I]** models an N-input system with an $N+1$ state Markov model, state j being the state in which j lines are active The arrival rate is simply the arrival rate of a single active line multiplied by the number of lines active A problem with this approach is that as N increases the solution becomes intractably complex

A common approach is to represent the system by a 2-state MMPP In **[BMLRW]** an interesting model is presented which attempts to capture the behaviour of the queue in both burst and sub-burst mode

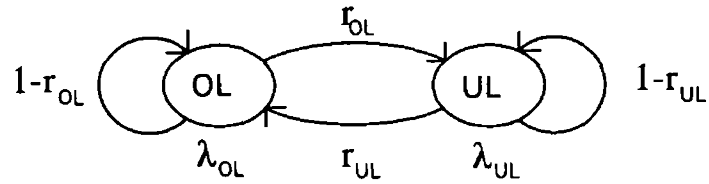


Fig 3 8 2-State MMPP Model of N-input Multiplexer

In this model the N + 1-state process described above is broken into two regions an overload region comprising the states $\{S+1, S+2, \quad ,N\}$ in which the cell arrival rate exceeds the multiplexer capacity and an underload region comprising states $\{0, 1, \quad ,S\}$ The model has four parameters r_{OL} (r_{UL}), the transition rate out of the overload (underload) state and λ_{OL} (λ_{UL}) the mean arrival rate of the cells in the overload (underload) state The cell arrival rate in each state is approximated as a Poisson process The accuracy of this model depends on the choice of these parameters

The time τ which the system spends in the OL state can be identified with a time T that is the time to absorption of an N-S state transient Markov process made up of the states $\{S, S+1, \quad , N\}$ where S is the absorbing state (This assumes that the system is stable and that the OL condition will end at some stage[1]) The transition matrix Q for this system can be used to calculate the distribution of this time [B] (the details are beyond the scope of this discussion) Then r_{OL} must be chosen so that the decay of τ approximates that of T **[BMLRW]** shows a way of reducing this problem to a matter of evaluating the maximal real-part eigenvalue of **Q** and setting r_{OL} equal to it The value λ_{UL} is derived by averaging the rates in the underload region, given Λ, the rate of cell arrivals due to a single active line and the various p_i, the probability of being in state i

$$\lambda_{UL} = \Lambda \sum_{i=0}^{S} i \frac{p_i}{P_{UL}}, \quad \text{where } P_{UL} = \sum_{i=0}^{S} p_i$$

λ_{OL} is derived by defining a new random variable, ν the number of cells produced while in overload The details of the derivation of λ_{OL} are not described here but the technique mirrors that used to generate r_{OL} A new matrix Q^* is generated and the survivor function for ν is calculated Given these three parameters, the last parameter r_{UL} is chosen so that the overall mean arrival rate of the process is the mean arrival rate of the process being modelled, i e $Nr\Lambda$, where r is the activity rate as defined in section 3 2 3 above This gives

$$r_{UL} = r_{OL} \frac{Nr\Lambda - \lambda_{UL}}{\lambda_{OL} - Nr\Lambda}$$

The results of applying these parameters are given and compared with simulation results The model behaves quite well when predicting sub-burst mode queues Using this model upper and lower bounds for the queue length distribution can be derived These bounds are demonstrated to be quite good by the numerical results presented in the paper

3 4 5 Fluid Flow Models

Fluid flow models derive from the observation that when the cell interarrival times of the overall arrival stream are small with respect to the time between arrival rate changes then the discrete nature of the arrivals can be approximated by a continuous fluid flow [AMS], [T] The assumption loses the sense of descretisation and consequently the information relating to the sub-burst component of the queue However it leads to fairly straightforward closed form expressions for the queue behaviour in burst mode

Each of the models above has its limitations Models that try to handle the contribution of each individual cell quickly become intractable Those that model the superposed streams must for simplicity select the parameters used to characterise the stream This inevitably leads to a loss of information Fluid models lose the discrete nature of the arrival process and therefore cannot model the sub-burst mode queueing behaviour at all Some of these limitations can be overcome by combining models

3 4 6 Combined Models

Combined models utilise the fact that the queue at any time has two components

$$Q_t = B_t + C_t$$

Each component can be calculated using a different model, the results being combined to obtain the overall queue characteristics

In [NRSV] , the sub-burst mode queue model presented in [RV] and [VR] is combined with a fluid model to provide an overall model

In the cell level case it is assumed that $n < S$ (i e sub-burst mode) The queue at any given point depends only on the arrivals since the last time the queue was empty Because the system is in sub-burst mode the queue is certain to be empty at least once every S ticks Calling the current tick 0, then if we take any tick u then, if the queue is empty at u, and there are currently x cells in the queue then there must have been $(x + u)$ arrivals in the time period $[u,0]$ So

$$P[Q = x] = P[(x + u) \text{ arrivals in } [u, 0]]\ P[(n-(x+u)) \text{ arrivals in } [S,u]]\ P[Q_u = 0]$$

Since the probability of an arrival in any one slot is $1/S$ this probability is given by

$$P[Q_0 = x] = \binom{n}{x+u}\left(\frac{u}{S}\right)^{(x+u)}\left(1-\frac{u}{S}\right)^{(n-x-u)} P[Q_u = 0]$$

Since $(n - x - u)$ cells have been delivered in $(S-u)$ ticks we can define the traffic intensity ρ to be

$$\rho = \frac{n-x-u}{S-u}$$

The probability of an empty queue at u is $1 - \rho$ which gives the final element of the equation. Summing this probability over all possible u gives us the final expression

$$P[Q_0 > x] = \sum_{u=1}^{S-x}\binom{n}{u+x}\left(\frac{u}{S}\right)^{(x+u)}\left(1-\frac{u}{S}\right)^{(n-x-u)}\frac{S-n+x}{S-u}, \quad 0 \le x \le n,\ n \le S$$

The burst-mode queue component can be calculated from a fluid-flow model. [NRSV] present some ideas on developing such a model but for their numerical results they use the model described in [AMS]. Fig 3 9 is a copy of the results obtained in [NRSV] for the two components. It can be seen that the sub-burst queue is the dominant force for low queue lengths, the burst mode becoming more important as the queue length increases. The queue length distribution is the sum (convolution) of these two distributions. This general characteristic will emerge in the simulation results presented in subsequent chapters.

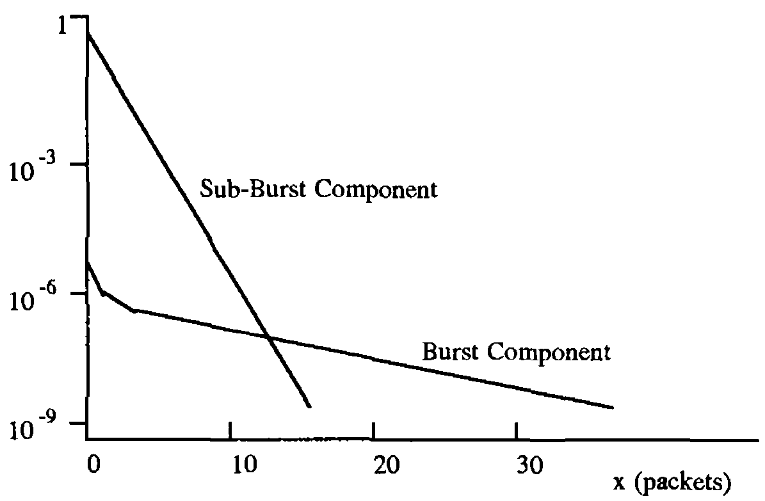


Fig 3 9 Burst and Cell components of Queue length distribution a la NRSV

3.5. The Three State Markov Model

3 5 1 Introduction

Having considered the range of models presented in the literature, the last part of this chapter introduces a new model, coined the 3-state Markov Model, which may provide useful insights into multiplexer behaviour [DB]

The fundamental difference between the 3-state model and previous models is that it deals with variable bit rate sources Whereas previously the intercell arrival time of each source is deterministic during an active burst, in a variable bit rate source the intercell arrival is non-deterministic, and is controlled by some probabilistic mechanism It is interesting to consider the case where such a source is chosen so that its characteristics exactly match that of a constant bit rate on/off source in terms of mean intercell arrival rate and activity ratio It is intuitively plausible (although by no means certain) that a superposition of many of these sources will behave in a manner similar to the superposition of the constant bit rate sources It is also clear that the behaviour will be fundamentally different in some respects due to the different correlations which would arise What is interesting is to establish where this alternative model could be usefully applied and where it might break down

3 5 2 Defining the Model

In this model we consider an on-off source with geometrically distributed intercell arrival times while on The length of each active period is also geometrically distributed with mean $\overline{T}_A$ and the silence periods are geometrically distributed with mean $\overline{T}_S$ We define three states for the line state 0 corresponds to the line being in an active burst and delivering a cell, state 1 to the line being active but not delivering a cell and state 2 to the line being inactive We consider the transitions from silence to activity and back to be independent of the cell transitions - in particular we consider that the probability that we receive a cell immediately after a transition to activity, or that the line becomes inactive immediately after a cell delivery, is negligible The resulting model is shown in fig 6 1

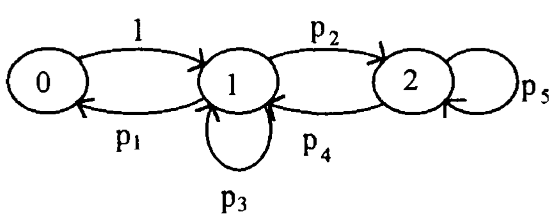


Fig 6 1 3-State Markov Model for Variable Bit Rate Line

Clearly

$$p_1 + p_2 + p_3 = 1$$
$$p_4 + p_5 = 1$$

The probability that the system remains in silence (state 2) for k ticks is $p_5{}^k$ The probability that a system transition from silence to activity occurs after $k+1$ ticks is $p_4\ p_5{}^k$ The mean time spent in a silence period is therefore

$$\overline{T}_S = 1 + p_4 \sum_{k=0}^{\infty} (k+1) p_5{}^k$$

The mean time between cell arrivals, $\overline{S}$, corresponds to the mean time spent in state 1 A similar argument to that for the silence length leads to

$$\overline{S} = 1 + p_1 \sum_{k=0}^{\infty} (k+1) p_3{}^k$$

To evaluate the mean time spent in an active state we note that both states 0 and 1 correspond to an active state Therefore the probability of remaining in such a state is $(p_1 + p_3)$ and we can apply a similar argument to obtain

$$\overline{T}_A = 1 + p_2 \sum_{k=0}^{\infty} (k+1)(p_1 + p_3)^k$$

We can solve these five equations in five variables to obtain the following results

$$p_1 = \frac{1}{2(\overline{S}-1)} \left(1 \pm \sqrt{1 - 4\frac{(\overline{S}-1)}{(\overline{T}_A - 1)}} \right) - \frac{1}{(\overline{T}_A - 1)}$$

$$p_2 = \frac{1}{\overline{T}_A - 1}$$

$$p_3 = 1 - \frac{1}{2(\overline{S}-1)} \left(1 \pm \sqrt{1 - 4\frac{(\overline{S}-1)}{(\overline{T}_A - 1)}} \right)$$

$$p_4 = \frac{1}{\overline{T}_S - 1}$$

$$p_5 = 1 - \frac{1}{\overline{T}_S - 1}$$

For the values of p1 and p3 we are only interested in the real roots, so $\frac{\overline{T}_A - 1}{\overline{S} - 1} > 4$ In practise, this means that the line must deliver, on average, more than 4 cells per active period This is not a problem since in all practical circumstances the mean number of cells delivered per active period is substantially higher (Indeed, if the mean was in the order of 4 then the line behaviour tends more towards a non-bursty variable bit rate source and would be better modelled as a Poissonian system) It is still necessary to establish which real root to accept as the correct one Since the probability must be positive we have

$$p_1 = \frac{1 \pm \sqrt{1-4z} - 2z}{2(\overline{S}-1)}, \text{ where } z = \frac{(\overline{S}-1)}{(\overline{T}_A - 1)}$$

$$\Rightarrow \pm\sqrt{1-4z} \geq 2z - 1$$

This is clearly only true for the positive root

The transition matrix, **P**, for this system is

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ p_1 & p_3 & p_2 \\ 0 & p_4 & p_5 \end{bmatrix}$$

We can use this to evaluate the stationary probability matrix $\overline{\Pi}$ (defined as $\Pi\mathbf{P}=\Pi$) with the result

$$\Pi = \begin{bmatrix} \frac{p_1(1-p_5)}{(p_1+1)(1-p_5)+p_2} & \frac{1-p_5}{(p_1+1)(1-p_5)+p_2} & \frac{p_2}{(p_1+1)(1-p_5)+p_2} \end{bmatrix}$$

These equations allow the model to be parameterised

3.6. Conclusion

3 6 1 Scope of Current Work

The brief review of the modelling approaches given here serves to underline the main problems associated with multiplexer analysis

- Complexıty An ATM multıplexer can have hundreds of ınputs Even ın the sımple case where these are all on-off ınputs wıth determınıstıc ınter-arrıval tımes when actıve, the problem of representıng thıs quıckly becomes ıntractable

- Two Modes of Operatıon The sub-burst and burst modes of operatıon have dıstınctly dıfferent characterıstıcs Models whıch work well to characterıse one regıon do not work so well wıth the other In order to encompass both modes compromıses have to be made whıch reduce the accuracy, or ıncrease the complexıty, of the model

Some of the ıssues raısed here wıll be addressed ın the followıng chapters All the models ın the lıterature are based around small systems where N ıs ın the order of 100 and the ınterarrıval tımes of cells are short It ıs not clear how the characterıstıcs of these models scale up wıth ıncreasıng N, or even ıf ıt ıs reasonable to apply the same models to large N Thıs problem wıll be addressed ın chapter 5 The area of prıme ınterest ıs the regıon of operatıon whereby the length of queues due to burst mode operatıon are comparable to those due to sub-burst mode operatıon, or are at least bounded wıth a known probabılıty It ıs not of ınterest to the engıneer to model sıtuatıons of extended burst mode, or of low traffic ıntensıty

The subsequent chapters shall concentrate on these two areas

[B1] P T Brady, "A Technıque for Investıgatıng On Off Patterns of Speech *Bell Sys Tech J* , 44, Sept 1965

[B2] P T Brady, "A Statıstıcal Analysıs of On-Off Patterns ın 16 Conversatıons Bell Sys Tech J , Jan 1968

[G] J G Gruber, "A Comparıson of measured and Calculated Speech Temporal Parameters Relevant to Speech Actıvıty Detectıon", *IEEE Trans Commun* Vol COM-30, No 4, Aprıl 1982

[Y] Y Yatsuzuka, "Hıghly Sensıtıve Speech Detector and Hı-speed Voıceband Data Dıscrımınator ın DSI-ADPCM Systems", *IEEE Trans Commun* vol COM-30, No 4, Apr 1982

[SW] K Srıram, W Whıtt, Characterızıng Superposıtıon Arrıval Processes ın Packet Multıplxers for Voıce and Data *IEEE Jour Sel Areas Commun* Vol SAC 4, No 6, Sept 1986

[CG 722] CCITT Recommendatıon G 722, 7 kHz Audıo-Codıng wıthın 64Kbıt/s , Blue Book, Melbourne Nov, 1988

[CG 721] CCITT Recommendatıon G 721, "32 kbıt/s Adaptıve Dıfferentıal Pulse Code Moduladıon (ADPCM) , Blue Book, Melbourne, Nov 1988

[SKS] K Srıram, R Scott McKınney, M Sherıf, "Voıce Packetızatıon and Compressıon ın Broadband ATM Networks , *IEEE Jour Sel Areas Commun* ,Vol 9, No 3, Aprıl 1991

[HL] H Heffes, D Lucantı, "A Markov Modulated Characterısatıon of Packetısed Voıce and Data Traffic and Related Multıplexer Performance ', *IEEE J Sel Areas Commun* , vol 4, Sept 1986

[K] B Kım, "Characterızatıon of Arrıval Statıstıcs of Multıplexed Voıce packets , *IEEE J Sel Areas Commun* , Vol SCA-1, No 6, Sept 1983

[I] I Ide, Superposition of Interrupted Poison Processes and its Application to Packetised Voice Multiplexers , *Proc Int Teletraffic Cong 12* Turin, Italy, May 1988, Paper 3, 1B2

[BMLRW] A Baiocchi, N Melazzi, M Listanti, A Roveri, R Winkler, Modelling Issues on an ATM Multiplexer within a Bursty Traffic Environment , *Proc ITC 13*, Denmark, June 1991, 2C 2 1

[B] U N Bhat, Elements of Applied Stochastic Processes , pps 70 -112, Wiley, New York , 1984

[AMS] D Anick, D Mitra, M Sondhi, Stochastic Theory of a Data Handling System with Multiple resources , *Bell Syst Tech J* Vol 61, No 8, 1982

[T] R Tucker, Accurate method for Analysis of a Packet Speech Multiplexer with Limited Delay , *IEEE Trans Commun* Vol 36, No 4, April 1988

[NRSV] I Norros, J Roberts, A Simonian, J Virtamo, The Superposition of Variable Bit Rate Sources in an ATM Multiplexer , *IEEE Jour Sel Areas Commun* Vol 9, No 3, April 1991

[VR] J Virtamo, J Roberts, Evaluating Buffer Requirements in an ATM Multiplexer , *GLOBECOM 89* Dallas, TX, Oct 1989

[DB] D Botvich, personal communication

4. Modelling the ATM Multiplexer

4.1. Model Definition

4 1 1 Introduction

To address the modelling problems described in chapter 3 it is usually necessary to simulate the system being described The rest of this report is devoted to analysing some dynamics of a multiplexer system This chapter describes the simulation approach that was used Some of the design factors affect different aspects of the simulation results There is necessarily a trade-off between execution speed, versatility and flexibility

4 1 2 Model Definition

The code here is designed to model a general situation whereby there are N inputs to a multiplexer with one output Inputs can be of any type and input types can be mixed in any ratio whatsoever The multiplexer contains a buffer, the length of which is defined at run time

The parameters for a simulation run are the number and type of inputs, the length of the buffer and the initial buffer queue if any How this general model is implemented is described in the following section

The model extracts the probability density function of the queue length as its primary result It also produces some statistics for the behaviour of each individual line on the multiplexer (the activity ratio, total number of cells delivered by the line, the total number of transitions from active to inactive by that line where appropriate) as well as overall multiplexer statistics (total number of cells handled by the multiplexer, number lost, probability of cell loss, total time which the multiplexer spent in burst mode) Each simulation run also monitors the rate of convergence of the system to a steady state This is done by taking "snapshots" of the pdf at various stages during the run and calculating the sum of the differences for each queue length value (see section 4 2 5) All of these statistics help give a greater understanding of the system as well as serving to identify potential errors (for example, grossly incorrect line activity statistics could indicate an internal fault such as an overflow error)

4.2. Description of Model

4 2 1 Object Oriented Analysis

One of the prime difficulties with any simulation is that it is not always clear what parameters will be important, nor is it clear what information needs to be garnered from the simulation run The code must be written with flexibility and expansion in mind This is facilitated by using object oriented programming (OOP). With this approach each element of the model is viewed as a separate object with its own data and functions for accessing that data. The data is hidden from the rest of the system which can only access it through the object's functions. The advantage of this approach is that if an object needs to be redesigned to change its behaviour, then, as long as the interface functions remain the same, there is no need to change the rest of the code. Objects are self-contained, re-usable blocks which have been chosen to match the physical characteristics of the system as closely as possible.

Using this approach it is possible to define three functionally different objects within the model: a source, an input line and the multiplexer itself. The source object represents a data source and completely characterises the source in terms of its traffic parameters. The input line represents the activity of one particular line in terms of delivering cells, interarrival times, etc., which is determined by the type of source associated with the line. An Input object represents one possible instance of the range of behaviours open to a Source object due to the random behaviour of the source. A Source object defines the random behaviour, an Input object executes that behaviour. The Multiplexer object represents a collection of input lines, the buffering, loss and delivery method of the multiplexer. In any given configuration there will be one Source object for each source type, N Input objects and one Multiplexer object.

4.2.2 Source Object

A Source is an object which completely characterises a source. There is a single instance of a Source object for each type of source which the model supports. There are four basic source objects defined in the model. All possible sources can be implemented directly using one of the four base objects or a further class can be derived from one of the base objects. The Source object hierarchy is defined in fig. 4.1.

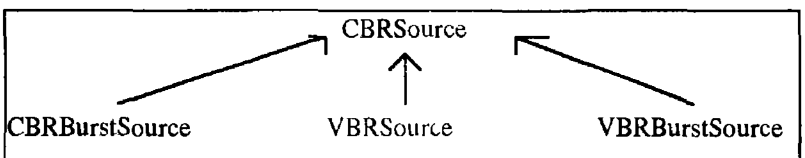


Fig. 4.1 Source Object Hierarchy (by convention arrows point from derived classes to base class)

The CBRSource represents the simplest type of source possible (a Constant Bit Rate Source). This source delivers cells with a deterministic inter-arrival time and is continuously active. Therefore it can be characterised using only two attributes:

- `name` The name of the source (for identification purposes)

- `period` The period, or mean intercell arrival time for the source

The next simplest source is the CBRBurstSource This represents a constant bit rate source which exhibits on/off behaviour To characterise this source the following additional attributes are required

- `blocks` The mean length of an active period in blocks Here a block is a period of time which is the mean intercell arrival time in ticks This is equivalent to the mean number of cells generated in an active period

- `activity` The ratio of time spent active

- `activeRngFPtr` The probability distribution of periods of activity (e g uniform, geometric)

- `inactiveRngFPtr` The probability distribution of periods of inactivity

Since not all sources provide a deterministic inter-arrival time a further category (class) of sources is required This is the variable bit rate source This source is constantly active but the intercell arrival time is controlled by some probabilistic mechanism This source can be characterised by the two attributes used for a CBRSource with the following additional attribute

- `periodRngFPtr` The distribution of the inter-cell arrival times

Finally a bursty VBR source can be characterised by combining the VBRSource attributes with the CBRBurstSource attributes

Using these four classes all types of source which can be encountered can be represented To make the source type invisible to the objects which use the sources each type offers an identical range of interface functions (methods)

- An initialisation function which allows all the attributes to be specified The parameters passed to this function depend on which source type is being initialised

- `meanInterCellTime()` A function which returns the (mean) inter-cell arrival time

- `interCellTime()` A function to return a number representing the time to the next cell arrival For VBR sources this is random, for CBR sources it is the same as the period

- `activeTime()` A function to return a number representing the (possibly random) length of an active burst For CBR sources this returns -1 as an identifier

- `inactiveTime()` A function to return a number representing the (possibly random) length of an inactive burst For CBR sources this is 0

- `hasActivity()` A function returning the ratio of time spent active to total time

- `hasMeanOnTime()` A function to return the mean length of an active period

- `hasMeanOffTime()` A function to return the mean length of an inactive period

Other variables and functions are contained in the Source object but are not described here since they are used primarily for housekeeping or internal reasons Details of how one of these source objects could be used to implement a source are given in section 4 3 1

4 2 3 Input Object

The Input object controls how a source delivers its packets to the multiplexer It is a representation of a line This object contains the core of the model There is one instance of an Input object for each line Since the Input object contains the model description, different input objects need to be defined for different models The input object contains the following attributes

- `sourcePtr` A pointer to a Source object representing the source type for this line

- `active` A flag to state whether the line is in an active state or not

- `length` The length (in ticks) of the current state, whether active or inactive

- `nextPacket` The time (in ticks) until the next cell is delivered by this line

- `packets` The total number of cells delivered so far by this input

- `lostPackets` The total number of those cells which were lost due to a full buffer at the multiplexer

- `activeTime` The total time (in ticks) that this input has been active

- `activePeriods` The total number of separate periods of activity of this input

The following interface functions are provided for this object

- An initialiser which allows the source type to be specified The input is randomly initialised to be active or inactive If active the length of the active period and the time to the next tick are randomly (uniformly) initialised This means that the line is initialised in a "steady-state" and minimises the need for a settling time when running the simulation (Results show that the system contains very little memory so the settling time, regardless of initial conditions, is very short However, it is desirable to "kick-off" the system in a state which is as close as possible to the eventual expected steady-state condition)

- A function `service()` which represents the passage in time of one tick and changes the state of the line accordingly This function implements the model The function operates as follows

```
IF line is active
    decrement time to next cell
    IF time to next cell is zero
        deliver cell to multiplexer
        generate new time to next cell (from Source)
        update statistics for this line
    END IF
END IF
decrement length of current state
IF length of current state is 0
    toggle state and update statistics
    update length and time to next packet
END IF
```

  The return value of this function determines whether a cell is delivered to the multiplexer or not

- `incLostPackets()` A function to update the number of packets lost

- A selection of functions to extract the statistical information about the input

If a different model is to be implemented a different `service()` function would be required This may require a different set of variables within the input object Although it would be possible to make the Input object more general to make installing different models easier the resulting loss in efficiency would lead to longer simulation run times It is more satisfactory to rewrite the Input object description to suit An example of this is described in section 4 3 2 where the 3-state Markov model for a line is used

4.2.4. Multiplexer Object

The Multiplexer object represents the action of a multiplexer. There is one instance of the Multiplexer object for each multiplexer being simulated (normally one). This object has the following attributes:

- `InputPtr:` A pointer to a singly linked list of Input objects representing the inputs to the multiplexer. The linked list structure offers efficiency advantages over using an array or any other arrangement and allows maximum flexibility.
- `QueuePdf:` A pointer to an array which is the size of the buffer. This array holds the number of instances of a given queue length and from this the probability density function of the queue distribution can be built up.
- `OldPdf:` A pointer to a second array which is the size of the buffer. This array can hold an older copy of the queue pdf for use in determining the rate of convergence of a particular model.
- `InputData:` A pointer to a singly linked list of source descriptors. This list contains all the information relating to the number and type of sources entering the multiplexer. There is one list element per source type. This is used for simulation management and documentation purposes.
- `Inputs:` The total number of inputs.
- `Buffer:` The size of the buffer.
- `Queue:` The size of the queue at any given moment.
- `Packets:` The total number of packets received by the multiplexer and the number that have been lost due to buffer overflow.
- `Block:` The length of a block (equivalent to the longest mean interarrival time of the input sources - this represents the period of the output stream).
- `Roe:` The average traffic intensity for the simulation.

The following functions are provided for interfacing with the Multiplexer object:

- An initialiser which allows the buffer length and initial queue to be specified. An initialised multiplexer object has no inputs - these must be added explicitly.

- `attachInput()` A function which attaches a given number of inputs of a given type to the multiplexer The number of inputs is only limited by the available memory on the system (and run length considerations!) Only inputs of a single type are permitted to be attached by a call to this function However, multiple calls to the function are allowed when mixing source types

- `tickService()` A function to simulate the passage in time of one tick This function traverses all the inputs, activating their `service()` routines, and adds any cells delivered by an input to the buffer If the buffer is full the cell is discarded Finally, if the buffer is not empty a cell is removed from it The statistics relating to the simulation are updated This function returns a flag indicating whether the multiplexer delivered a cell to its output in this tick

- `blockService()` A function to represent the passage of one block (a time period of the mean intercell arrival time in ticks) This simply calls the tick service routine once for every tick in a block It is necessary to use this system to maintain statistics about the length of time spent in burst mode

- `convergence()` A function to provide some information on the rate of convergence of the model by calculating the error given by

$$\varepsilon = \sum_{\text{buffer length}} \left| \mathrm{f}(x) - \mathrm{f}'(x) \right|$$

  where $\mathrm{f}(x)$ is the current pdf of the queue length and $\mathrm{f}'(x)$ is the pdf from some time previously The function calculates and return ε and saves the current pdf for future calls to this function

- Various functions for extracting the statistical information

It was originally intended to extract statistics for the number of packets lost for each individual line thereby giving a indication of the loss probability per line However the algorithm described above treats each line in sequence and immediately discards a cell if the queue is full The effect of this is to weight the loss probability so that it is dependent on the line's position The probability of loss for the last line treated in each tick would be higher than that for the first This is not an accurate reflection of the situation in real-life To overcome this it is necessary to choose packets to be dropped in some random fashion Although fairly straightforward and perfectly feasible, this would lead to an unacceptable increase in the run time of the algorithm so this feature was not implemented

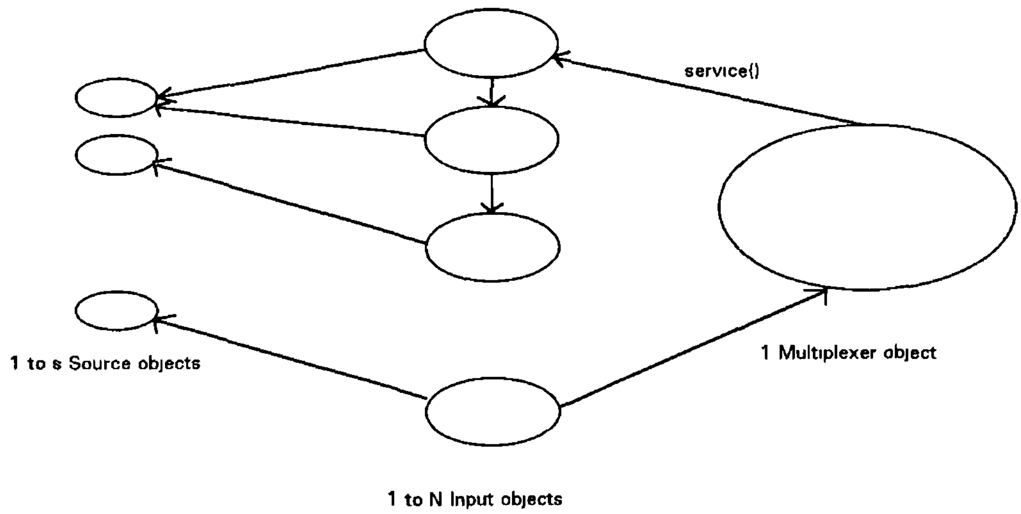


Fig 4 2 Interaction of simulation objects *s* source types are represented by *s* Source objects Each one of *N* inputs is represented by an Input object This object can interrogate the Source associated with it to determine its attributes A single Multiplexer object represents the entire multiplexer This calls the `service()` routine for each Input in sequence until the required number of "ticks" have elapsed Each `service()` represents one "tick"

4 2 5 Integration

The three object types above are integrated by the `main()` function This creates an instance of each required source type It then creates an instance of a multiplexer object with a given buffer size, and initial queue length Then an appropriate number of inputs of each source type can be attached to the Multiplexer object This action creates and initialises all the input objects necessary The simulation is then enacted by calling the multiplexer block service routine for the predetermined number of blocks Statistics are recorded in a log file which gives useful background data on the run as well as providing diagnostic information The pdf of the queue is recorded in a data file The file names may be specified on the command line Data is stored in ASCII text columns

4.3. Expanding Model

4 3 1 Defining a new Source

All sources can be specified using one of the four classes defined in section 4 2 2 above The initialisation information required for each source differs however Table 4 1 gives the initialisation parameters

Source Type	Attnbutes
CBRSource	Text name of source Interarnval tıme
CBRBurstSource	Text name of source Interarnval tıme Mean number of cells delıvered durıng an actıve penod Actıve ratıo Dıstnbutıon of actıve tıme as a poınter to a functıon whıch returns numbers wıth the appropnate dıstnbutıon Dıstnbutıon of actıve tıme as a poınter to a functıon whıch returns numbers wıth the appropnate dıstnbutıon
VBRSource	Text name of source mean ınterarnval tıme Dıstnbutıon of ınter-arnval tımes as a poınter to a functıon
VBRBurstSource	Text name of source Interarnval tıme Mean number of cells delıvered durıng an actıve penod Actıve ratıo Dıstnbutıon of actıve tıme as a poınter to a functıon whıch returns numbers wıth the appropnate dıstnbutıon Dıstnbutıon of actıve tıme as a poınter to a functıon whıch returns numbers wıth the appropnate dıstnbutıon Dıstnbutıon of ınter-arnval tımes as a poınter to a functıon

To define a typıcal voıce source, for example, the followıng parameters would be used

Text Name	"voıce"
Intercell arnvals tıme	2000 (based on a tıme unıt of 2 7μs)
Mean number of cells durıng actıve penod	53
Actıve ratıo	0 35
Dıstnbutıon of actıve penod	geometnc
Dıstnbutıon of ınactıve penod	geometnc

To define the dıstnbutıons a functıon must exıst whıch returns random numbers wıth geometnc dıstnbutıon The functıon call to buıld the source (assumıng that the random functıon ıs called `georandgen()`) ıs then

```
CBRBurstSource("voıce", 2000, 53, 0.35, georandgen, georandgen),
```

Dıfferent dıstnbutıons can be represented by passıng dıfferent functıon poınters However, all functıons used must have the followıng C declaratıon format

```
long  func(double),
```

The parameter to be passed to the functıon ıs stored ın the Source object Thıs may have to be ınıtıalısed dependıng on what the functıon ıs expected to do The neatest way to do thıs ıs to derive a class of Sources from one of the four classes to implement the specific class required Some examples of how this is done is included in the source code (srctypes h)

4 3 2 Altering the Model Dynamics

The model is contained in the `Input service()` routine Two models have been implemented for this report The first, described in section 4 2 3 above, is the straightforward simulation The second is the 3-state Markov Model described in the previous chapter The implementation of this model is described here as an example of how different models can be implemented

The 3-state model is implemented via a different Input object definition This altered Input object has the following structure

- A pointer to a Source object representing the source type for this line
- A flag to identify which of the three states the line is in
- Pointer to a transition matrix for the model
- Pointer to a "next-state" matrix for the model
- The total number of cells delivered so far by this input
- The total number of those cells which were lost due to a full buffer at the multiplexer
- The total time (in ticks) that this input has been active
- The total number of separate periods of activity of this input

The following interface functions differ from those provided by the previous input object

- An initialiser which allows the source type to be specified This initialisation function builds up the transition matrix based on the parameters of the source type The particular form of this matrix is not the standard mathematical form Instead the probabilities have been re-arranged so that the highest transition probability occurs first in each row This allows a much faster algorithm to be used for implementing the model The disadvantage of this approach is that a separate matrix (the "next-state" matrix) is needed to keep track of which transition relates to which probability entry in the matrix Further, the layout of the matrix is determined by the model and only works with the 3-state model that is implemented The input is randomly initialised to be active or inactive

- A function `service()` which represents the passage in time of one tick and changes the state of the line accordingly This function implements the model The function operates as follows

```
generate a random number uniformly distributed in the range 0-1
Select the row of the transition matrix corresponding to current state
FOR each element in the row
        IF random number lower than element
                BREAK
END FOR
determine next state from next state matrix
IF state is "active with cell ready"
        RETURN cell
ELSE
        RETURN no cell
```

  It should be noted that some statistical information is lost using this algorithm It is not possible to keep track of the number of state transitions for an individual line without some extra processing Although not difficult this would slow down the simulation which is already quite slow due to the high proportion of calls to the random number generator

Only the functionality of these functions differ from the corresponding ones in the previous Input object described The interface to the object remains the same which means that no changes need to be made to either the Source or Multiplexer objects to link in this model Other models can be implemented in a similar fashion So long as the range of interface functions provided is identical to that in the default Input object the changes will be invisible to all other sections of the code

4 3 3 Limitations

The primary limitations on the code are memory and execution speed Each input requires a block of memory to control its behaviour, as does each source type and the multiplexer itself The larger the buffer in the multiplexer, the more memory is required Depending on the number of inputs problems begin to arise with buffer lengths of around 500 cells on a PC (larger buffers are allowed on a SUN) Similarly there is an upper limit on the number of inputs - although simulations with 3000 inputs have run satisfactorily

The code has been optimised as much as is possible The run time is directly proportional to the product of the number of "ticks" and the number of inputs On a PC-386 (33 MHz) the model executes at about 10μs per input per tick The 3-state model runs significantly more slowly

It is worth noting that there is generally a run-time overhead associated with OOD This is mainly due to the extra work performed resolving object pointers and determining which function to call where virtual functions are used This is typically in the region of 10% (as compared to C code) There is one virtual function call in the inner loop of the simulation This could be removed (or taken outside the loop) where performance is critical

The base random number generator used in all the simulation work is due to L'Ecuyer [E] and is a combined linear congruential generator with a period of 2 3×10^{18} Consequently care should be taken about run lengths which may exceed this value There are approximately R calls to the rng for a constant bit rate source where R is

(Number of lines x total simulation time in ticks)/(Mean ticks active x 2)

For a variable bit rate source this number increases by a factor approximated by the mean number of cells delivered per active period But for the Markov implementation the number of calls is approximately the number of ticks in the simulation by the number of lines

The geometric random number generator used is derived from an example given by Luc Devroye [D] and requires just one call to the uniform random number generator per call

---

[E] P L Ecuyer, "Efficient and Portable Combined Random Number Generators", *Communications of the ACM* Vol 31, No 6, June 1988

[D] Luc Devroye, Non Uniform Random Variate Generation , p 87, Springer-Verlag, 1986

5. Queueing Behaviour and Multiplexer Resourcing

5.1. Introduction

The models outlined in chapter 3 all try and determine the nature of multiplexer behaviour in two modes, burst and sub-burst queueing In this chapter a more detailed look is taken at this queue behaviour and in particular, a simple method of determining the required buffer size is introduced The results show that it is not feasible to try to buffer burst mode packets in large systems, while sub-burst mode queues can very easily be handled

5.2. Coping with Burst Mode Queues

5 2 1 Distribution of Active Lines

In section 3 4 2 above a two-state Markov Model for representing a single line (in macro-view, i e without considering distribution of cell arrivals within a block) was presented Here we use this model to consider the superposition of N such lines and consider the steady-state probability of a particular number n lines being active, p(n)

$$\mathrm{p}(n) = \binom{N}{n} \mathrm{P}[n \text{ lines are active}]\ \mathrm{P}[N-n \text{ lines are inactive}]$$

The probability of any single line being active is given by the ratio of active time to inactive time (r) which is equivalent to ($a/(a+d)$) in the model Thus

$$\mathrm{p}(n) = \binom{N}{n} r^n (1-r)^{N-n}$$

For the voice source presented in chapter 3 the probability of a line being active is 0 35 The function p(n) for N inputs of this type is shown in figure 5 1

This distribution forms the basis from which we can derive an idea of the critical region of operation The critical region is here defined as that region whereby burst-mode queue length is bounded and of the same order as sub-burst queue lengths The requirement that the queue length be bounded (in the sense that it does not go off to infinity) implies that the system is stable and burst-mode behaviour does not occur for long periods Of course it is necessary that the queue length be such that the mean cell delay is acceptable and this also forces an upper bound on the queue length What is needed is some function relating N to S so that, given S, it is possible to define an N whereby maximum multiplexer gain is achieved without significantly increasing the probability of packet loss The probability of burst mode is given by

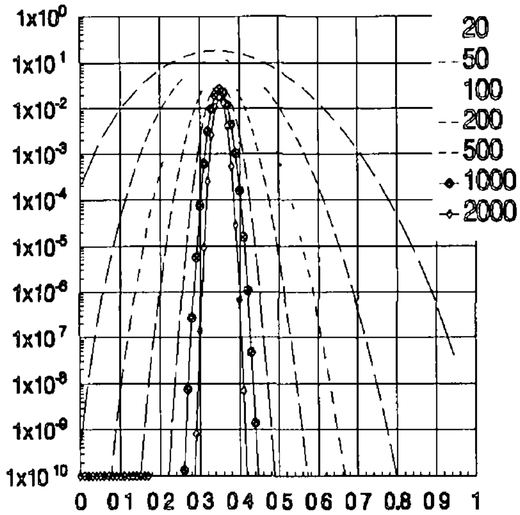


Fig 5 1 Distribution of active lines for various N

$$P[\text{BurstMode}] = P[n > S] = \sum_{n=S+1}^{N} \binom{N}{n} r^n (1-r)^{N-n}$$

The sub-burst region is that region for which $n < S$ Any value of N will have an associated probability of burst mode behaviour It remains to be determined what probability of burst mode will lead to acceptable burst-mode queues

5 2 2 Burst Mode Queuing

A definition of the critical region can be obtained if we consider a multiplexer with a fixed buffer size B We assume that b_c slots are required for the cell component leaving b_b slots for the overflow element When the system is in burst mode $(n - S)$ excess cells are delivered every block The burst mode is transient, lasting for a value of τ blocks, and the number of active lines will reach a value n_{max} before retreating to a value less than S The mean number of excess cells delivered per block while in burst mode is

$$\bar{c} = \left[\sum_{n=S+1}^{n_{\max}} p(n)n\right] - S$$

$$= \left[\sum_{n=S+1}^{n_{\max}} \binom{N}{n} r^n (1-r)^{N-n} n\right] - S$$

Fig 5 2 shows two possible profiles for an excursion into burst mode In Fig 5 2a a typical profile is shown Fig 5 2b shows a "best case" excursion For this best case situation we can approximate the number of excess cells per block by

$$\bar{c} \approx \frac{n_{\max} - S}{2}$$

In the more typical case of fig 5 2a, the mean number of excess cells per block will be higher than this approximation It is useful, though, to look at this "best case" situation to gain an understanding of the burst mode queue behaviour

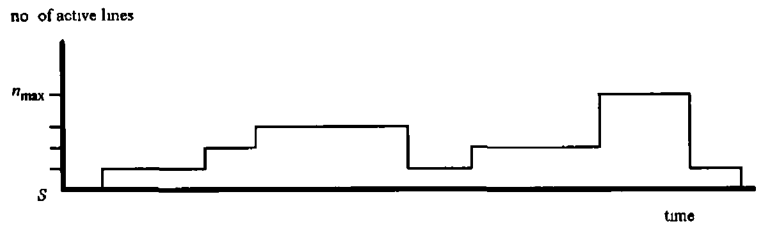


Fig 5 2a Typical Burst mode excursion

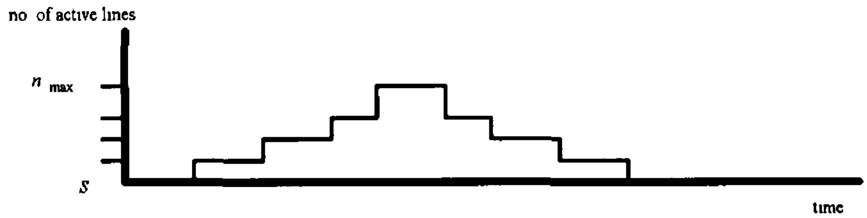


Fig 5 2b "Ideal" Burst mode excursion

Since the system is in burst mode for τ blocks we have the condition that

$$\left(\frac{n_{\max}-S}{2}\right)\tau \leq b_b$$

which leads to the following relationship for the maximum number of active lines which can be permitted for a given buffer size

$$n_{\max} \leq \frac{2b_b}{\tau}+S$$

where n_{max} and τ are dependent random variables

The number of lines we can support is dependent on τ, and in particular is dependent on the typical length of a burst mode excursion, $\overline{\tau}$ To evaluate $\overline{\tau}$ it is useful to refer back to the N state Markov model for the system described in section 3 4 3 This is represented by an NxN state transition matrix where

$$\mathrm{p}(m,n) = \sum_{l=\max(0,n-N+m)}^{\min(m,n)} \binom{m}{l}(1-d)^l d^{m-l}\binom{N-m}{n-l}a^{n-l}(1-a)^{N-m-n+l}$$

The vector π consisting of the column vector of elements {p(0), p(1), p(N) } is the stationary probability vector of this process where the p(n) is defined at the beginning of this chapter We are only interested in whether the system is in burst mode or not We can therefore partition the matrix into four regions

$$
\mathbf{P} = \left[\begin{array}{ccc|ccc}
p(0,0) & p(0,1) & p(0,S) & p(0,S+1) & p(0,N-1) & p(0,N) \\
p(1,0) & p(1,1) & p(1,S) & p(1,S+1) & p(1,N-1) & p(1,N) \\
 & & & & & \\
p(S,0) & p(S,1) & p(S,S) & p(S,S+1) & p(S,N-1) & p(S,N) \\
\hline
p(S+1,0) & p(S+1,1) & p(S+1,S) & p(S+1,S+1) & p(S+1,N-1) & p(S+1,N) \\
 & & & & & \\
p(N-1,0) & p(N-1,1) & p(N-1,S) & p(N-1,S+1) & p(N-1,N-1) & p(N-1,N) \\
p(N,0) & p(N,1) & p(N,S) & p(N,S+1) & p(N,N-1) & p(N,N)
\end{array}\right]
$$

We can collapse this entire model into a two state model where the states correspond to burst mode and sub-burst mode which has a 2x2 transition matrix

$$
\mathbf{P}' = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}
$$

The model resulting from combining states in a Markovian model like this is generally not Markovian, however we can still derive the transition matrix from the original model [W] Because the original process has a stationary probability vector the new process also does

The element p(n) of this vector can be viewed as the number of visits to state n per unit time The elements of **P** can be viewed as the fraction of these that are transitions from state m The fraction of transitions across the partition of states must be the same, regardless of any Markovian behaviour, therefore we can interpret the elements p_{ij} in terms of the original elements as follows

$$
p_{ij} = \sum_{\text{all states in } i} \frac{(\text{fraction of transitions from state } m \in i)\ (\text{fraction of transitions which are from } m \text{ to state } n \in j]}{(\text{fraction of time spent in any state } \in i)}
$$

which is equivalent to

$$
p_{ij} = \sum_{\text{all states } m \in i} \frac{\text{P[system was in } m]\ \text{P[transition from } m \text{ to } n \in j]}{\text{P[system is in any state } \in i]}
$$

Therefore we can derive the elements of the collapsed matrix from **P** as follows

$$
p_{00} = \sum_{m \le S}\sum_{n \le S} \frac{p(m)p(m,n)}{\mathrm{P}[m \le S]}, \text{ where } \mathrm{P}[m \le S] = \sum_{m \le S} p(m)
$$

$$p_{01} = \sum_{m \le S} \sum_{n > S} \frac{p(m)p(m,n)}{\mathrm{P}[m \le S]}, \text{ where } \mathrm{P}[m \le S] = \sum_{m \le S} p(m)$$

$$p_{10} = \sum_{m > S} \sum_{n \le S} \frac{p(m)p(m,n)}{\mathrm{P}[m > S]}, \text{ where } \mathrm{P}[m > S] = \sum_{m > S} p(m)$$

$$p_{11} = \sum_{m > S} \sum_{n > S} \frac{p(m)p(m,n)}{\mathrm{P}[m > S]}, \text{ where } \mathrm{P}[m > S] = \sum_{m > S} p(m)$$

The two divisors used in these summations are the elements of the stationary probability matrix for the new process

It remains to be seen whether the new process is Markovian A model resulting from the collapsing of a larger Markovian model will only be Markovian if the probability of remaining in state i is independent of which sub-state $m \in i$ the system is in [B] Or, more formally, if

$$P_{ij} = \sum_{m \in i} P_{nm} \text{ for all } n \in j$$

This is not the case here where the probability of exiting sub-burst mode increases as more lines are active That is to say

$$\left(\sum_{m \in i} P_{nm} \text{ for } n \approx S \right) \neq \left(\sum_{m \in i} P_{nm} \text{ for } n \ll S \right)$$

However, we have made the assumption that the system is stable, which is equivalent to saying that it spends most of its time in sub-burst mode (state 0 of the derived model) The large model transition matrix typically takes the following form

$$\mathbf{P} = \left[\begin{array}{ccccccc|ccccc}
X & Y & Z & & & & & & & & & \\
Y & X & Y & Z & & & & & & & & \\
Z & Y & X & Y & Z & & & & 0 & & & \\
 & Z & Y & X & Y & Z & & & & & & \\
 & & Z & Y & X & Y & Z & & & & & \\
 & & & Z & Y & X & Y & Z & & & & \\
 & & & & Z & Y & X & Y & Z & & & \\
\hdashline
 & & & & & Z & Y & X & Y & Z & & \\
 & & 0 & & & & Z & Y & X & Y & Z & \\
 & & & & & & & Z & Y & X & Y & Z \\
 & & & & & & & & Z & Y & X & Y \\
 & & & & & & & & & Z & Y & X
\end{array} \right]$$

Here the diagonal (X) values are close to 1 (with the maximum occurring close to the partition for the heavily loaded system under consideration) The values on either side drop off quickly and tend to 0 This diagonal form becomes more pronounced as N gets large As the system tends towards a more diagonal form the behaviour of the collapsed model becomes more Markovian It is therefore reasonable to state that, in the operation region of interest, excursions to the burst state will be strongly Markovian and thus the burst mode excursion time will be dominantly geometric with mean given by the inverse of p_{10} So,

$$\bar{\tau} \approx \frac{1}{p_{10}}$$

Relating this back to the question of how many active lines we can support gives us

$$n_{\max} \leq 2p_{10}b_b + S$$

In other words, n_{max} is linear with respect to b_b but with a slope of $2p_{10}$, assuming an "ideal" excursion profile p_{10} is very low, in the area of interest, (in the order of 10^{3} or less) and tends to 0 as N gets large

It should be remembered that n_{max} here is taken from a "best case" assumption for the excursion In general this will not be true, the mean number of cells delivered per block while in burst mode will be higher than assumed here and a larger buffer will be required

5 2 3 Effect of the CLT

Given that P[loss] is approximated by P[$n > n_{max}$], it is obvious that we can improve P[loss] by increasing the buffer size However, we gain only small improvements for large buffer sizes Further, as N increases, p_{10} decreases reducing the effect of any buffer This means that although buffers strongly benefit performance of small systems they have little or no effect on large systems once a buffer large enough to accommodate cell level queues exists This result could have been anticipated from fig 5 1

As N increases fig 5 1 shows that the ratio of the variance to the mean decreases Assuming that S as a proportion of N remains constant this means that for very high N the two state process tends towards

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

In other words the process will remain in whatever state it starts off in The result of the last section is saying that, given that the process starts off in sub-burst mode, the probability of it going into burst mode decreases as N increases (with fixed N/S) However, if burst mode is entered the state will persist It is therefore impractical to buffer the excess cells It should be noted that this applies only to large N, typically 5000 or more

In chapter 6 simulation results will show the effect of buffer size on small systems, and will also demonstrate the scaling effects implied by fig 5 1

As a final comment this result shows that, from a resourcing point of view, it is best to choose N such that the $P[n > S]$ is less than the desired probability of loss Figure 5 3 shows the relationship between multiplexer gain and the period of the multiplexed signal under this stipulation for various probabilities of $n > S$ This graph was extracted from the results for fig 5 1 by taking cross-sections along four probabilities The levelling out of the gain is evident

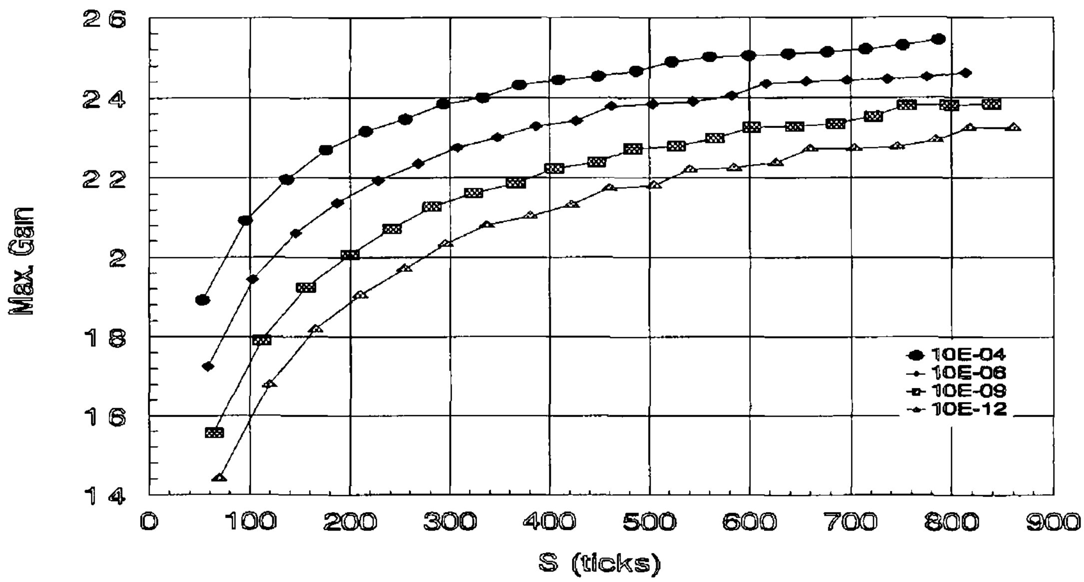


Fig 5 3 Multiplexer Gain versus S for various burst mode probabilities

5.3. Sub-burst Queueing

5 3 1 A Scaling Mechanism

In the previous section it was assumed that the buffer B had a burst and cell component This section takes a brief look at the bounds on sub-burst queues and the effect of scaling N upwards

If N and S are scaled upwards proportionately, but the mean arrival rate per line per block is kept constant, then the joint distribution of arrivals at any finite number m of ticks *within the same block* approaches that of m independent Poisson variables with mean ρ ([KT], [D]) Because the queue must be empty at least once in every block we lose sight of the long term correlations In the terminology of [SW] we work on a *short* timescale, by scaling the typical correlation time S with N Thus one expects that in sub-burst mode the queue length would be reasonably well described by that of a queue with M/D/1 arrivals, (although as the mean ρ

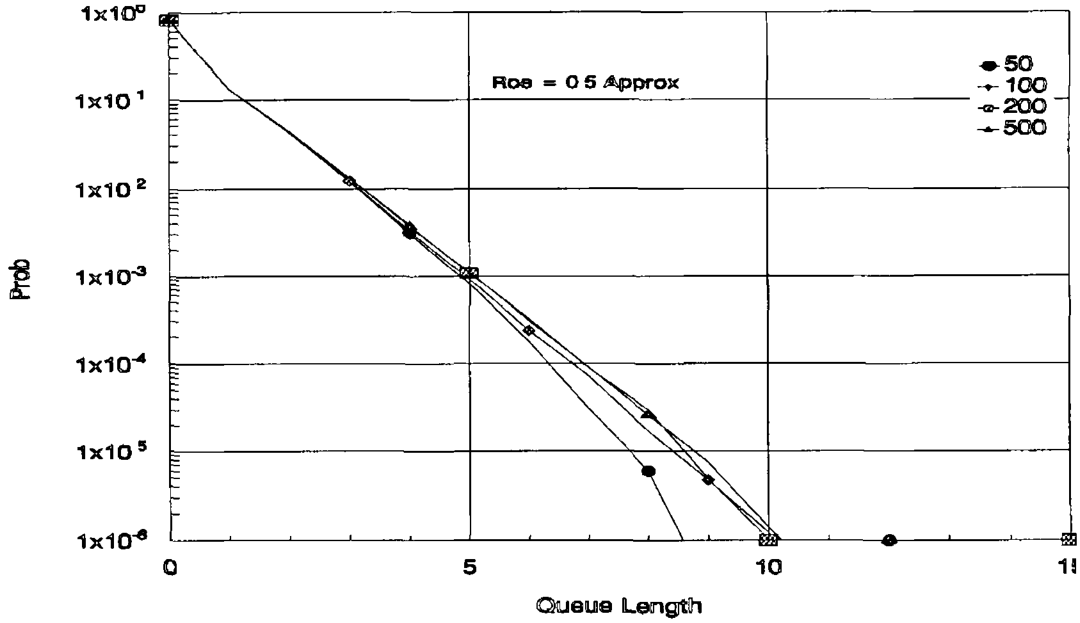


Fig 5 4 Sub-Burst Queue Component for Various S

approaches 1 excursions into burst mode will become more likely)

In effect, the sub-burst queue component is independent of N by this scaling mechanism (fig 5 4) [CDC]

When considering the effect of increasing N we need to consider what happens to the model dynamics In particular we want to preserve correlations between arrivals Using a tick as the unit of time we can identify the important parameters S the inter cell arrival period, N the number of lines, T_A the mean number of ticks spent active and T_S the mean number of ticks spent in silence We need to scale T_A with S in order to maintain correlations This is best done by ensuring that the mean number of cells delivered per active period, C_A, remains constant As we scale S upwards we also need to increase N

This approach has been tested by simulation and the results demonstrate the veracity of the argument Fig 5 4 shows the sub-burst mode queue component for various values of S To ensure that no burst mode operation occurred ρ was maintained at about 0 5 The simulation ran for over 10,000,000 samples For S in the range 100-500 there is almost no visible difference between the cell distributions This is as expected For S of 50 some deviation is visible In fact this deviation becomes more and more pronounced as S decreases This occurs because N is also decreasing (in this case $N/S \approx 1$ 4) As N decreases the effect of the CLT becomes less pronounced Individual line activity influences the result more and more Fig 5 5 shows how the queue distribution varies as ρ increases for a given S Fig 5 5a shows the result of burst mode operation, with the queue distribution becoming uniform for all queue lengths beyond a small value (for an infinite buffer)

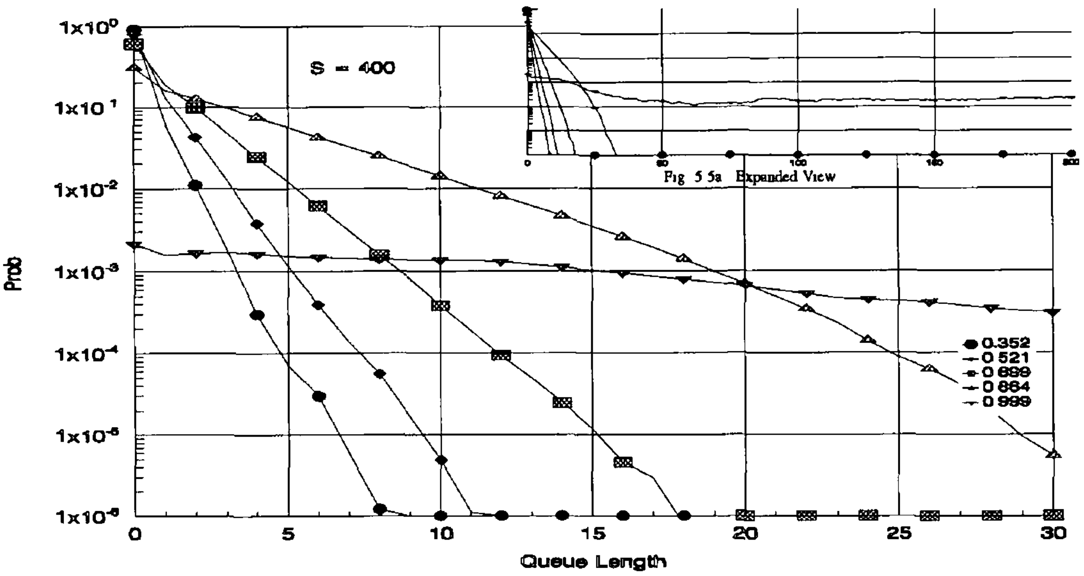


Fig 5 5 Queue Distribution for Various ρ

This result is interesting because it allows results taken from small scale systems to be extrapolated to large systems - the sub-burst queue behaviour remains the same (for a given ρ) However, *N* does not increase linearly with *S*, but increases at a faster rate The requirement is to ensure that the burst mode probability remains constant (If we were to scale *N/S* such that it remained constant it would not be possible to extrapolate burst mode behaviour of small scale models to that of large models since the region of operation would have changed, even though ρ has stayed the same) So, although it is safe to extrapolate sub-burst queue behaviour from small models, care must be taken when dealing with burst mode behaviour

5.4. Conclusion

In this chapter it has been shown that coping with burst-mode behaviour for large systems by adding large buffers is not practical, even for an ideally behaved system It has also been shown that the important aspects of multiplexer queues can be examined using reduced scale models, so long as the scaling mechanism preserves the correlations of the original processes The behaviour of the original system is not totally preserved however In particular, as the number of input lines is scaled down (regardless of *S)* the effect of the law of large numbers becomes less pronounced Thus the probability of entering burst mode will be higher for a given ratio of *N* to *S* It is essential then to ensure that the scaled values have the same probability of burst mode

These results show that models can be tested using much smaller systems than normally expected in "real-life", allowing greater efficiency in terms of simulation time In the next chapter various aspects of system behaviour shall be examined in detail, utilising scaled simulations

---

[W] R W Wolff, "Stochastic Modelling and the Theory of Queues , Prentice-Hall, New York, 1989, Chapter 3

[B] U N Bhat, "Elements of Applied Stochastic Processes", Wiley, New York, 1984, Chapter 5

[KT] S Karlin, H M Taylor, *A First Course in Stochastic Processes*, New York Academic, 1975

[D] N G Duffield, Local Mean Field Markov Processes An application to message switching networks", *Probab Theory Relat Fields*, to appear

[SW] K Sriram, W Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", *IEEE Jour Sel Areas Commun* ,Vol SAC-4, No 6, Sept 1986

[CDC] D Corry, N Duffield, T Curran, "Sub-Burst Mode Queueing in an ATM Voice Multiplexer , *9th UK Teletraffic Symposium* April 1992

6. Evaluation of Multiplexer Performance

6.1. Introduction

This chapter presents a selection of results from an extensive set of simulation runs The results show the multiplexer performance under various conditions of load and traffic type

There are five parameters which can reasonably be expected to affect multiplexer behaviour These are traffic density (load), burstiness of traffic, multiplexer load (i e gain), multiplexer buffer length, and input source mix The parameters of interest are the probability of cell loss, the queue length probability distribution function and the mean queue length Side issues which are also interesting are the stability of the model in terms of sensitivity to initial conditions These parameters are presented over the following sections

The reference source used in all simulations was a constant bit rate on/off source scaled to represent a voice source Unless otherwise stated the simulation parameters were

Mean cell interarrival time $\overline{S} = 100$

Activity ratio $r = 0\ 35$

Mean cells per active burst = 53

Number of inputs N = dependent on load requirement

Buffer length = 300

Each simulation was run over 1,000,000 ticks representing approximately 3 seconds of real time operation The buffer length of 300 used represents a buffer of about 6000 cells in a more realistic situation This is larger than would be expected (due to the resulting cell delays) but is necessary in order to reflect the fact that the loads used in the simulation are also higher than could be sustained in a real situation The reason for this is that the simulations aim to evaluate sensitivity to various parameters as opposed to actual loss probabilities Because of the short runs (1,000,000 samples) it is necessary to enforce cell loss probabilities of 0 0001 or greater In realistic circumstances the loss should be 10^{-9} or better The simulations examines trends, not absolute behaviour

6.2. Presentation of Results

6 2 1 Sensitivity to ρ

Fig 6 1 shows how mean queue length is affected by ρ (This graph is extrapolated from the data from a single series of runs with different load) In this case an infinite buffer was assumed The system can comfortably support loads up to about 0 9 without any significant queue build-up The queues start to build up very rapidly above 0 95 as the system wanders into burst mode with more frequency and greater duration

It is interesting to look at the probability of cell loss as ρ increases To do this a finite buffer must be chosen For this set of results a buffer length of 3*S* (300) was chosen (The significance of the buffer length on performance is discussed later) Figure 6 2 shows **P[loss]** v ρ This graph is an average of three separate runs for a CBR source, each with a different random number generator seed Here a similar characteristic is observed as would be expected (Further runs would be required to produce a smoother characteristic)

Fig 6 3 shows the queue pdfs obtained in two of the runs used to generate the previous graphs Here the effect of burst mode can clearly be seen For the lower load the queue never exceeds sub-burst levels For the higher load, though, occasional excursions into burst mode have occurred Note that the buffer length of 300 is adequate to cope with these At this kind of level of *S* buffering greatly improves system performance

Since we are primarily interested in the area close to burst mode the rest of this discussion will concentrate on high (> 0 9) traffic intensities

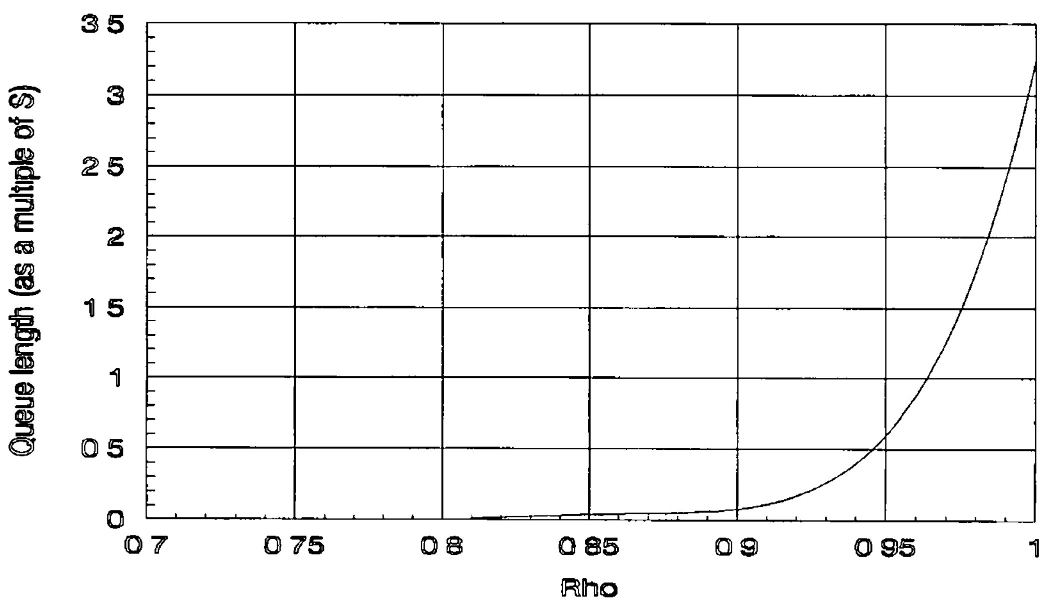


Fig 6 1 Queue length as a function of ρ

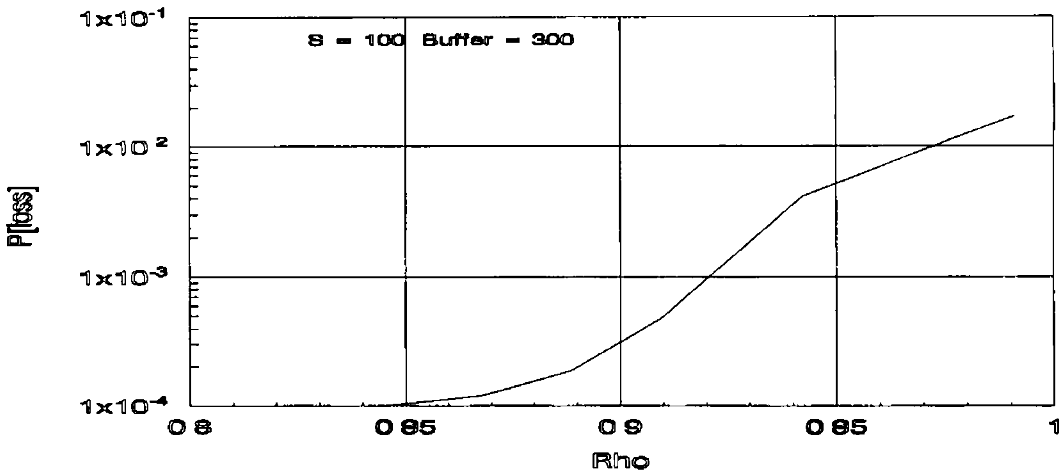


Fig 6 2 P[loss] as a function of ρ

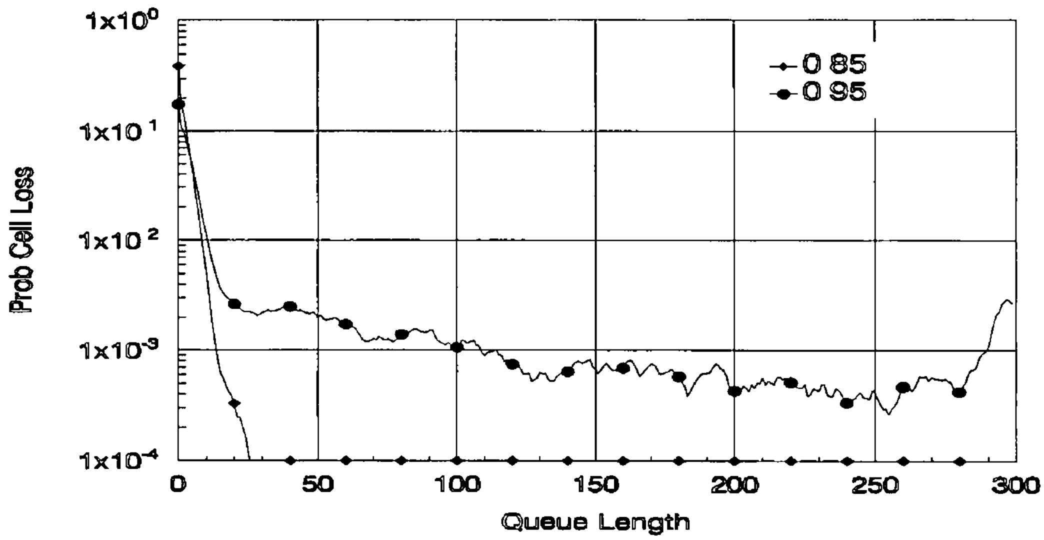


Fig 6 3 Typical pdfs for CBR source for different loads

6 2 2 Sensitivity to Burst Length

It is not sufficient to simply define the activity ratio r of a source and expect all sources of that ratio to perform equally It is possible that the actual length of the active burst will affect the behaviour of the system Long bursts will tend to reduce the probability of a change in the number of lines active from block to block thereby increasing the tendency of a system to remain in burst mode Short bursts should improve system throughput

To examine just how much this parameter affects performance several runs were made with differing burst lengths In each case $S = 100$, $r = 0\ 35$ and the mean number of cells delivered per burst was varied A buffer length of 300 was used and the simulation run for 1,000,000 ticks

Figures 6 4 and 6 5 show the comparative queue pdfs for a constant bit rate source for various values of burst length The range of burst length is from a mean of 5 cells per burst up to 1000 cells per burst The results show that system performance is relatively insensitive to burst length Although some slight improvement (to burst level queues) at low burst lengths is noticeable this quickly disappears and for burst lengths of greater than 50 cells/burst there is little or no difference The clearest indications of this are given by the run statistics for each trial which show little correlation between burst length and mean queue lengths Indeed the fluctuations that do occur are well within the bounds of random variations for this run It would be necessary to run multiple simulations with the same parameters and different random number seeds to extract with any certainty any sensitivity that may exist

However one element which did come from the simulations is that it is possible to achieve slightly higher gain with sources of shorter burst lengths To maintain a mean ρ as required it was necessary to input 260 sources of burst length 5 as opposed to 243 sources of burst length 50 This slight improvement disappeared for burst lengths greater than 50 It is not possible to draw any conclusions from this since that gain is small enough to be within the range of random error Further investigation would be necessary to evaluate whether this is a general feature of short burst sources

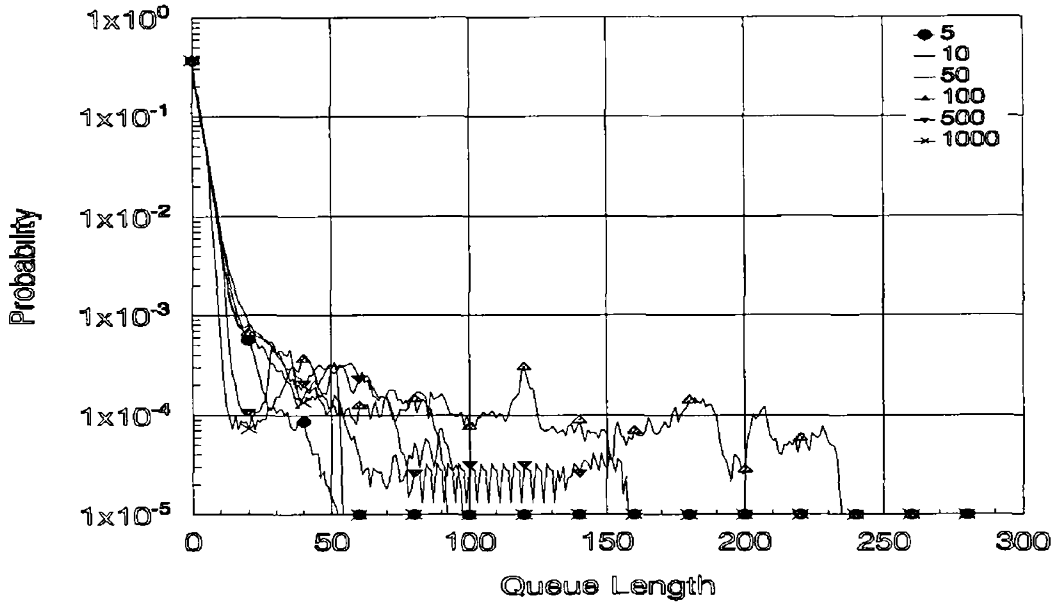


Fig 6 4 Queue pdf for various values of mean burst length (cells/burst) for $\rho \approx 0$ 85 for a voice (CBR) source The general statistics for this run were

cells/burst	Queue	ρ	P[loss]	%age burst time
5	2 24	0 847	0	3
10	2 63	0 853	0	3
50	2 82	0 848	0	2
100	4 52	0 840	0	2
500	2 76	0 845	0	2
1000	1 85	0 848	0	1

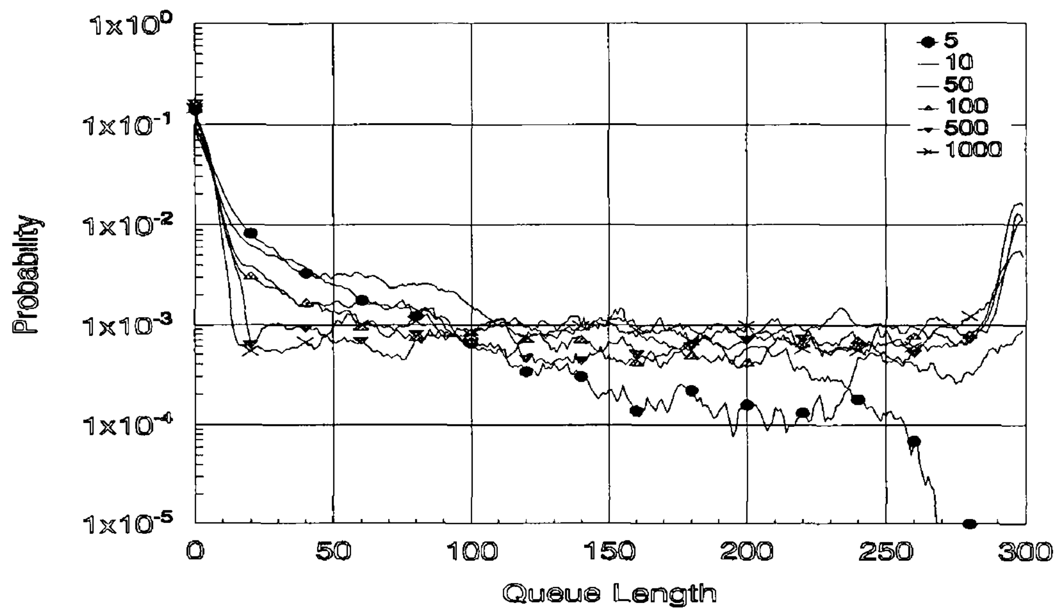


Fig 6 5 Queue pdf for various values of mean burst length (cells/burst) for $\rho \approx 0$ 95 for a voice (CBR) source The mean statistics for this run were

cells/burst	Queue	ρ	P[loss]	%age burst time
5	18 85	0 947	0	28
10	39 70	0 956	0 000408048	31
50	63 60	0 955	0 00512033	29
100	46 52	0 944	0 00343698	23
500	52 82	0 940	0 00455527	19
1000	71 23	0 953	0 00428106	25

6 2 3 Sensitivity to Buffer Length

In chapter 5 it was shown that the multiplexer gain is relatively insensitive to buffer length Fig 6 6 shows how buffer length affects performance for a given gain value for several values of ρ In this case S = 100, N = 273, 286 and 290 respectively (for increasing ρ) and the mean burst length was 53 cells The resulting ρ values are 0 95, 0 97 and 0 99 The graph shows the cell loss probability averaged over 5 runs for buffer lengths of 0, 20, 50, 100, 300, 500, 1000 and 2000 cells (with differing random number seeds) The results clearly show the two regions of operation For buffer lengths less than 20 the probability of loss is very high due to losses even at cell level But from 20 onwards there is a small but steady decrease in cell loss probability as the buffer length increases The improvement in performance is quite significant - a factor of ten for each 10S cells added This is not surprising for the low values of N used here As N increases the rate of improvement also increases which is in agreement with the arguments in chapter 5 However, for larger S the mean time spent in burst mode will increase (for a given ρ) which reduces the sensitivity to buffer length As ρ tends towards 1 the slope of this line tends towards 0 showing a reduced sensitivity to the buffer length

With every increase in buffer length the corresponding mean cell delay also increases, until a steady-state value is reached The steady-state value will only be reached when the buffer length is sufficient to prevent any cell loss Otherwise, the mean queue length will be distorted by the finite buffer Fig 6 7 shows the mean delay as a function of buffer size for the traffic intensities (The mean delay has been calculated by taking the mean queue lengths and multiplying by the deterministic service time of 2 7μs) The delays for this run are quite reasonable However, if applied to a realistically large system the effect is to increase the absolute value of the (burst) queue and therefore the delay (The characteristic remains the same) The delay caused by sub-burst queues is unaffected due to the consistency of sub-burst behaviour as the size of the system increases

The mean delay is the upper limiting factor on the buffer size It is expected that the overall network delay (end-to-end) will not be allowed to be more than 10ms The delay per node will be significantly less than this

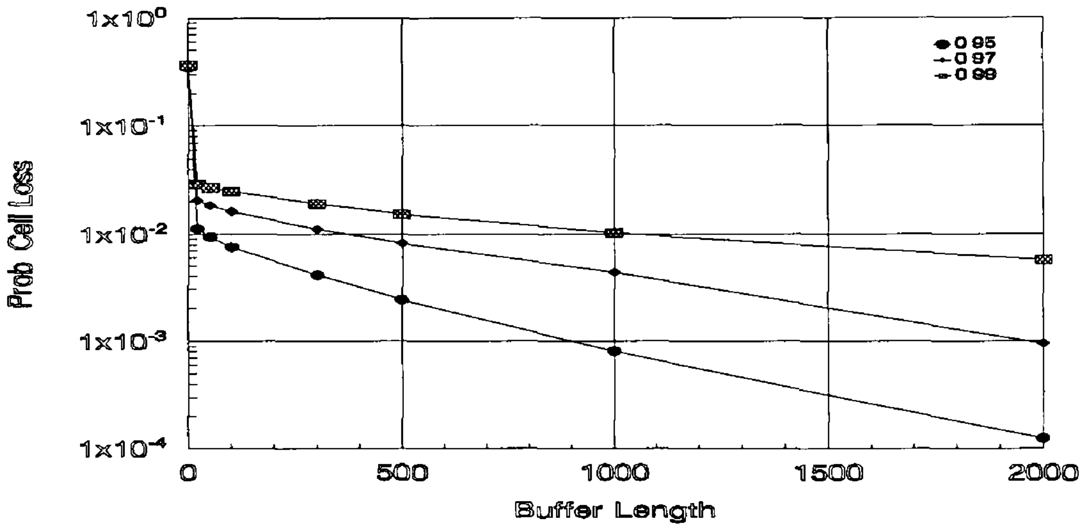


Fig 6 6 P[loss] versus buffer length (voice source)

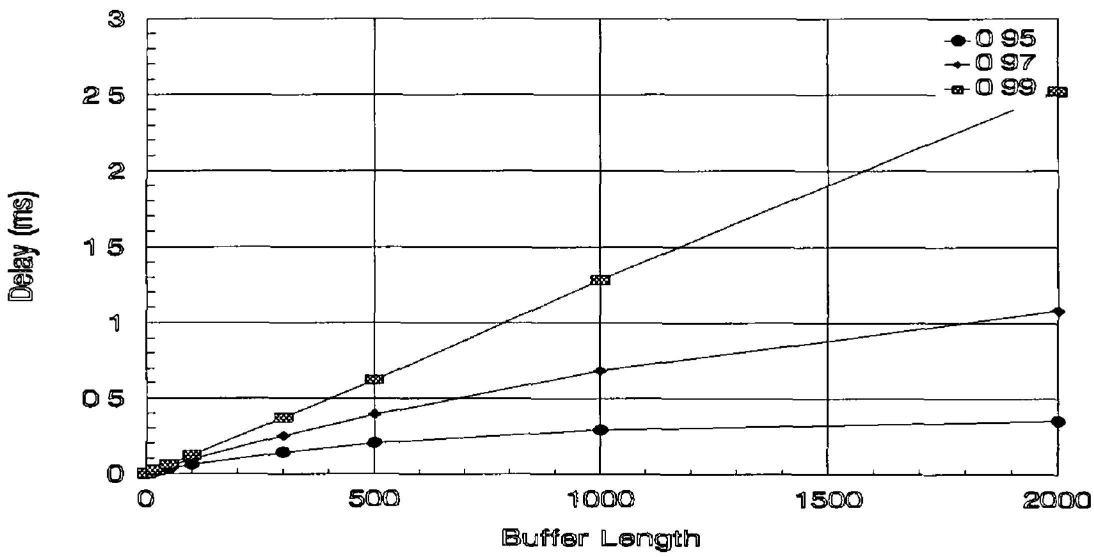


Fig 6 7 Mean Delay as a function of buffer length

6 2 4 Sensitivity to S/N in Sub-Burst Mode

In chapter 5 some consideration was given to the problem of scaling In this section more detailed results are presented for the sensitivity of the system to scaling By scalmg in this sense we mean increasing the cell inter-arrival time S It is intended to keep the mean traffic intensity for each line the same, therefore the mean number of cells delivered per burst is maintained constant Two scalmg mechanisms are considered In the first, N and S are scaled linearly In the second N increases at a faster rate in order to compensate the CLT effect and maintain a constant probability of loss

The sensitivity of behaviour to this type of scaling depends on which region of activity we are most interested in In chapter 5 a typical pdf for the sub-burst queue distribution for a range of S was shown Fig 6 8 gives a further proof of the relative insensitivity of the sub-burst mode queues to this kind of scalmg Here the mean sub-burst mode queue length is plotted as a function of S Clearly the sub-burst behaviour is not sensitive to the scaling of N and S

As we scale up linearly though, a change appears Fig 6 9 shows the probability of cell loss for a mean ρ of 0 93 (This graph is taken from three sets of simulations with differing random number generator seeds) The table shows other results from the run Although the load remains constant a change in multiplexer behaviour occurs For the low values of S the system operates mainly in burst mode As S increases a changeover takes place and the system reverts back to sub-burst mode behaviour The percentage of time spent in burst mode decreases from almost 50% down to 0 The mean queue length reflects this with high queues (relative to S) occurring while the system experiences burst mode and levelling out to a steady-state value as burst mode activity disappears The buffer length for this run was 50 cells in order to ensure that cell level queues would be catered for but that almost no burst level behaviour would be tolerated (for higher S)

This result is important in that it raises questions about trying to extrapolate system behaviour for a given load from small models to large ones It also is encouraging in that it points to the improvements in performance which are possible on the larger scale

6 2 5 Sensitivity to S/N in Burst Mode

In order to extrapolate burst mode behaviour from small models it is necessary to scale N non-linearly with S The scaling must be done so that the probability of burst mode remains constant Figs 6 10 and 6 11 shows the effect when the system is scaled like this If the scalmg is done to keep the load constant, then the probability of loss decreases with increasing S If the scalmg is done so that the probability of loss is constant then the load increases (and multiplexer efficiency improves) Note the increase in queue lengths in the table with fig 6 10

As the number of inputs increases so does the mean (sub-burst) queue, even though the probability of loss is constant In a practical situation the probability of loss would be held at a much lower value

It can be concluded from this that the system is partially sensitive to *S/N* when burst mode behaviour is being considered This is, in fact, a consequence of the results discussed in the previous chapter The important factor is *N* As *N* increases the probability of entering burst mode decreases The ratio of active to inactive lines will tend to remain close to the activity ratio As long as *S* is significantly higher than this there is little chance of burst mode However, if burst mode should occur, it will persist for a relatively long time

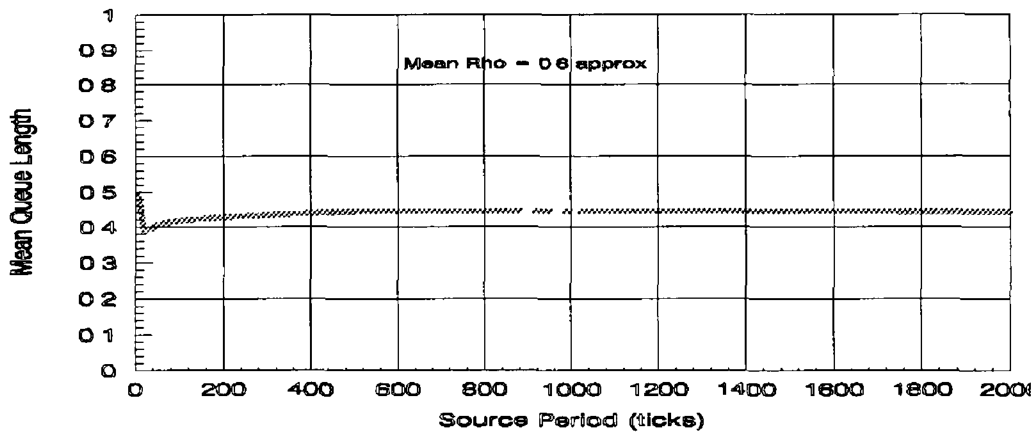


Fig 6 8 Mean sub-burst Queue length versus *S*

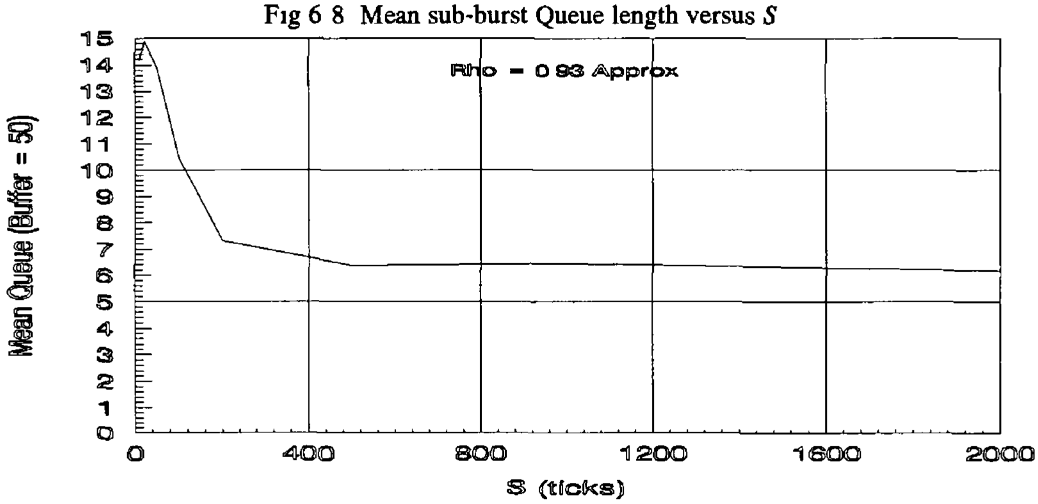


Fig 6 9 Mean Queue versus S for a fixed Load with a (CBR) source The general mean statistics for this run were

S	*N*	Queue	ρ	P[loss]	%age burst time
10	27	14 19	0 938	0 049676	47
20	54	14 89	0 946	0 034791	43
50	134	13 93	0 949	0 0184164	33
100	269	10 49	0 935	0 00724529	20
200	537	7 32	0 933	0 00235039	10
500	1343	6 37	0 933	0 000798895	4
1000	2686	6 46	0 933	1 31268x10 5	N/A
2000	5371	6 18	0 932	7 51131x10 6	N/A

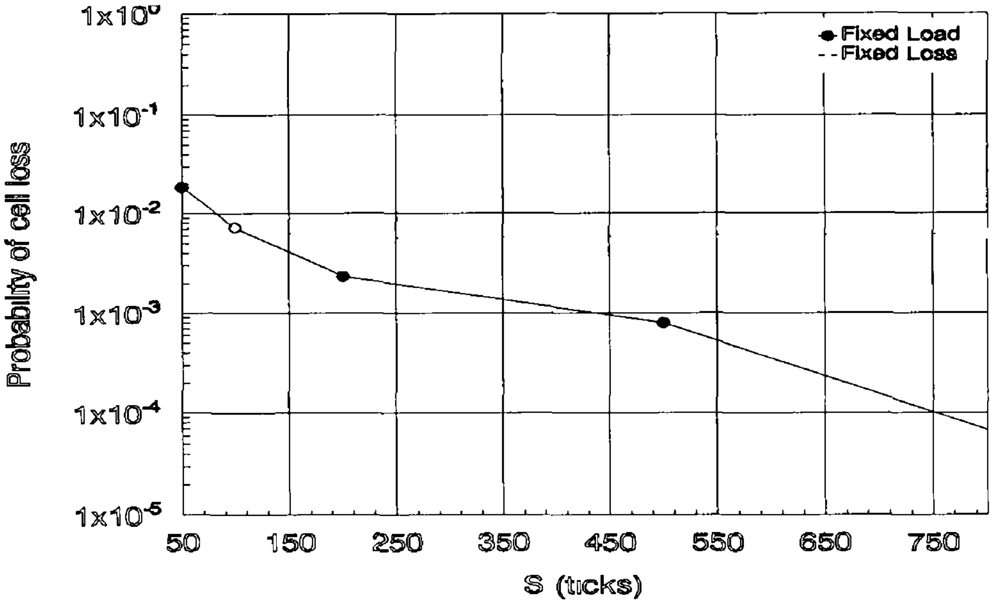


Fig 6 10 P[loss] under two different scaling mechanisms for a (CBR) source The fixed load results come from the results for fig 6 10 above The statistics for the fixed Loss results are

S	*N*	Queue	ρ	P[loss]	%age burst time
50	129	8 20	0 905	0 00723982	20
100	269	10 49	0 935	0 00724529	20
200	550	12 05	0 957	0 00555728	20
500	1415	16 05	0 979	0 00586067	22
800	2240	18 17	0 986	0 00620393	22

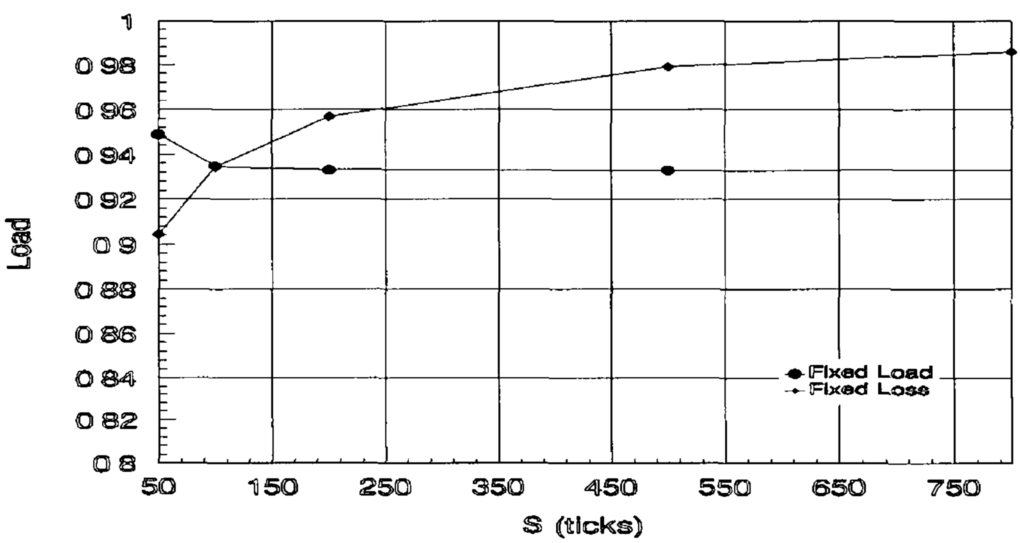


Fig 6 11 Traffic Load for the loss probabilities of fig 6 10

6 2 6 Mixed Intensity Sources

In a real situation it is extremely unlikely that all sources attached to a multiplexer will be homogenous It is possible to artificially create a homogenous scenario by breaking up the total available bandwidth into separate bands, each of which is allocated to only one type of source However, this scenario can only be applied to the situation where each source type has a separate buffer Although this is an interesting situation (especially since it is not clear how the multiple buffers affect each other given that only one cell from one buffer can be served in a tick) it is beyond the scope of this report (Extendmg the software to examine this situation would not be difficult)

An alternative situation is where a mixture of traffic sources share a single buffer This mixture can take two forms - either a mixture of sources of the same type but with differing traffic intensities or a mixture of sources of different type The latter situation is examined in the next section This section looks at a mixture of similar sources with different traffic intensities

There are many questions which can be addressed here One of the most basic ones is how the traffic mixture affects performance To evaluate this the system performance under three different traffic mixtures was examined Four sources were mixed All had a mean burst length of 53 cells/burst and an activity of 0 35 For source type **A**, $S = 4$, **B**, $S = 10$, **C**, $S = 50$ and for type **D**, $S = 100$

Figure 6 12 shows a typical pdf for the queue for three mixtures of "voice-like" (CBR) sources for ρ of about 0 83 Also shown is a typical pdf for a similar traffic intensity for both the highest and lowest intensity sources in the mix The performance of a mixture lies between the bounds provided by a homogenous collection of the highest (as an upper) and lowest (as a lower) intensity sources The performance of a mixture is dominated by the high intensity sources Mix 2, which is heavily weighted towards the high intensity sources, has the worst performance Mix 3, although performing better than the other mixes (it has only 5 high intensity sources), still performs significantly worse than traffic free of high-intensity sources These results are intuitive, clearly higher intensity traffic puts greater load on the multiplexer The performance for a mixture will fall somewhere in between that for homogeneous traffic of the highest and lowest intensities However, high intensity lines have a disproportionate effect on the performance

It should be noted that from the results of section 6 2 2, the sensitivity of the system to source intensity decreases once the intensity is above a certain level Therefore one would expect that adding sources with a period of 300 to sources with a period of 1000 would have relatively little effect

Fig 6 13 shows the same graph for higher intensity traffic (0 94) Here the difference between the mixes is not as obvious This implies that burst mode behaviour for a mixture is not significantly affected by the mixture type

This results suggest that care must be taken when adding high intensity sources to a multiplexer A small number of high intensity sources will tend to drive the multiplexer into burst mode earlier, even though the load remains the same

6 2 7 Mixed Traffic Types

Ultimately the B-ISDN network will carry a large range of traffic types where not only the source intensity but also the statistical behaviour will be different Although a full analysis of the interaction of deterministic and non-deterministic arrivals on the multiplexer queue is beyond the scope of this report a quick look at some simulation results is worthwhile as an indicator of this behaviour

Fig 6 14 shows a typical pdf for several variations of mixed traffic types Here the intensity of the sources is comparable (S = 100) but the mixture ratios differ The ratio of voice to source (in percentage) as respectively 20 80, 50 50, and 80 20 The results are inconclusive Although the VBR source tends to have slightly worse performance, the combination of mixes are interleaved Further analysis would be required using longer simulation runs to establish how sensitive performance was The system seems to be relatively insensitive to the mixture, certainly in comparison to its sensitivity to source intensity

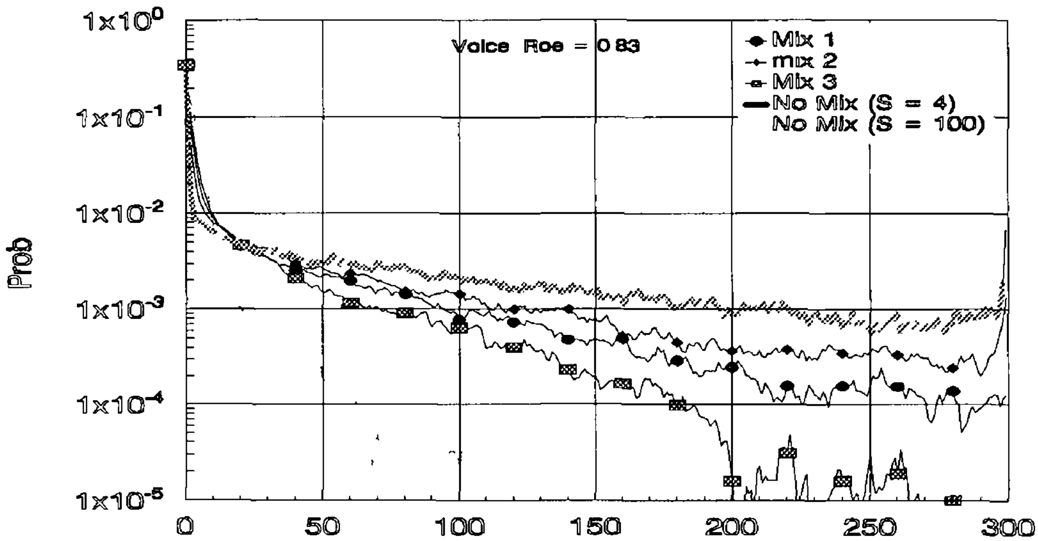


Fig 6 12 A sample distribution for mixed sources (ρ ≈ 0 83) The mixture is as follows

Source types	A	B	C	D
Distribution of sources in mix 1	2	7	33	67
Distribution of sources in Mix 2	4	9	16	31
Distribution of sources in Mix 3	1	4	51	90

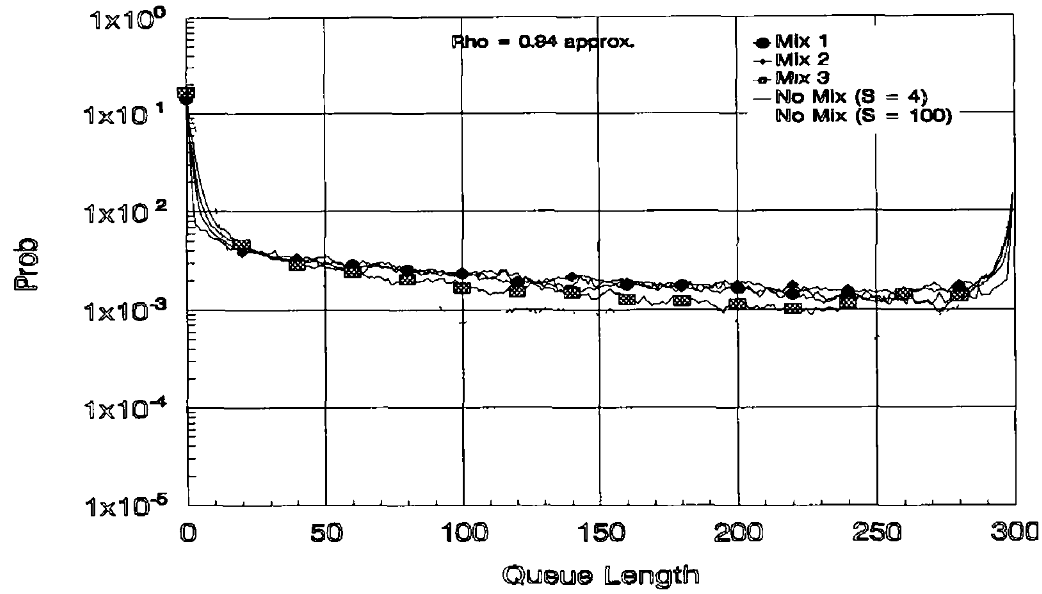


Fig 6 13 A sample distribution for mixed sources ($\rho \approx 0.94$) The mixture is as follows

Source types	A	B	C	D
Distribution of sources in mix 1	2	7	38	86
Distribution of sources in Mix 2	4	11	19	35
Distribution of sources in Mix 3	1	6	49	99

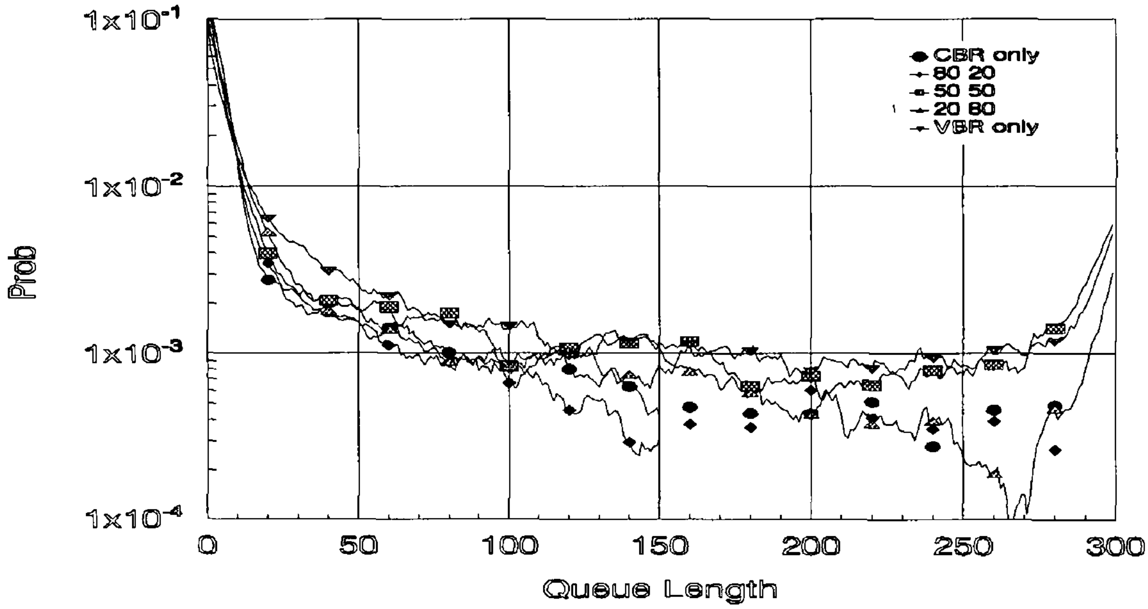


Fig 6 14 Comparison of typical pdfs for mixed traffic types

6.3. Evaluatıon of Model Performance

6 3 1 Three Models

Wıth the ıntroductıon of the 3-state model ın chapter 3 we have three dıfferent models In the first ınstance we have a "brute-force" sımulatıon of a bursty CBR source, wıth determınıstıc ınterarrıval tımes The second sıtuatıon ıs a "brute-force" sımulatıon of a bursty VBR source wıth geometrıc ıntercell arrıval tıme dıstrıbutıon The thırd model ıs the 3-state model descrıbed ın chapter 3 These models all dıffer ın theır characterıstıcs Thıs sectıon takes a look at the relatıve performance of these models, wıth partıcular reference to the 3-state model

Fıg 6 15 shows the comparatıve performance for the voıce, VBR and model loss probabılıtıes as a functıon of ρ It can be seen that the performance ıs very sımılar The VBR source shows the worst performance wıth consıstently hıgher loss probabılıtıes Thıs ıs consıstent wıth removıng the determınıstıc ınter arrıval tımes of the packets ın the Markov model, the resultıng system dısplays more "randomness" In essence the block level characterıstıcs of the system are reduced as are the correlatıons ın the arrıval stream The resultıng traffic behavıour ıs more Poıssonıan The 3-State model seems to perform sımılarly to the CBR model Thıs ıs not expected sınce ıt sımulates the VBR model However, further runs show the pdf of the model to be consıstently worse than that of the voıce (the sample pdfs ın figs 6 16 and 6 17 show thıs effect) Fıg 6 15 shows the average loss of three runs taken over 9 values for ρ Although thıs ıs sufficıent to show the broad sımılarıty between the model behavıours ıt would be necessary to run further sımulatıons to produce a truly accurate pıcture of the dıfferences

In fig 6 18 the comparatıve sensıtıvıty to buffer length for the models ıs shown Thıs demonstrates that the models behave very closely The VBR and the 3-state model are almost ıdentıcal, showıng losses whıch are consıstently about twıce that for the voıce source The rate of decay of the loss ıs comparable

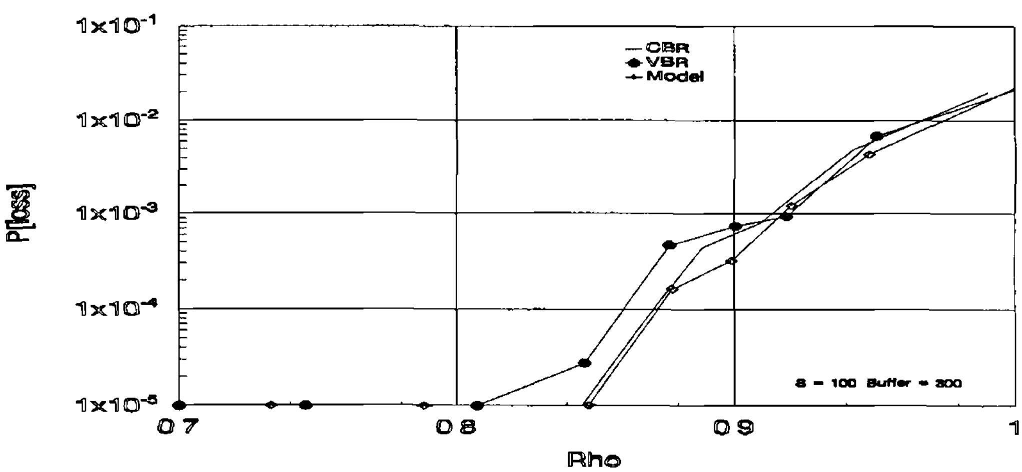


Fıg 6 15 Comparatıve performance of models

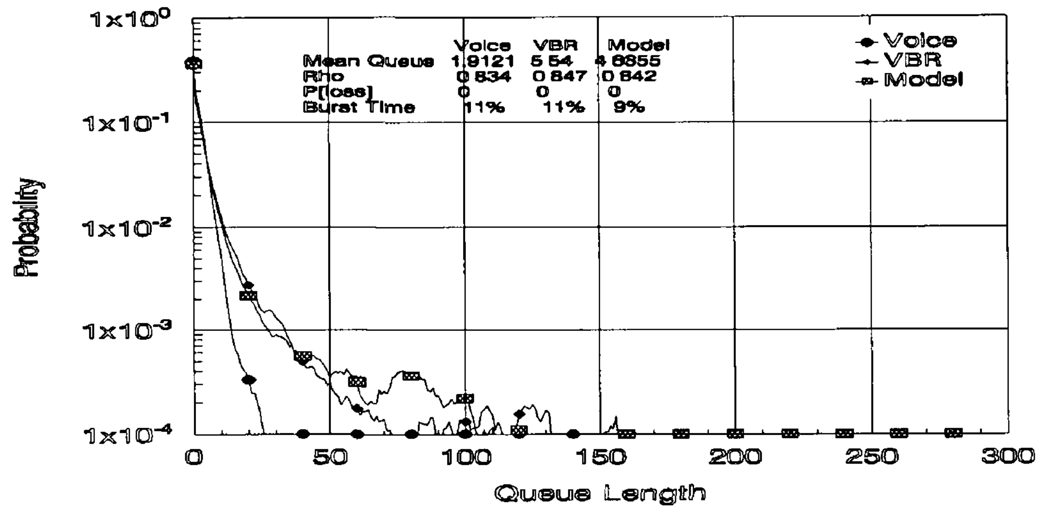


Fig 6 16 Queue Length pdf for $\rho \approx 0$ 85

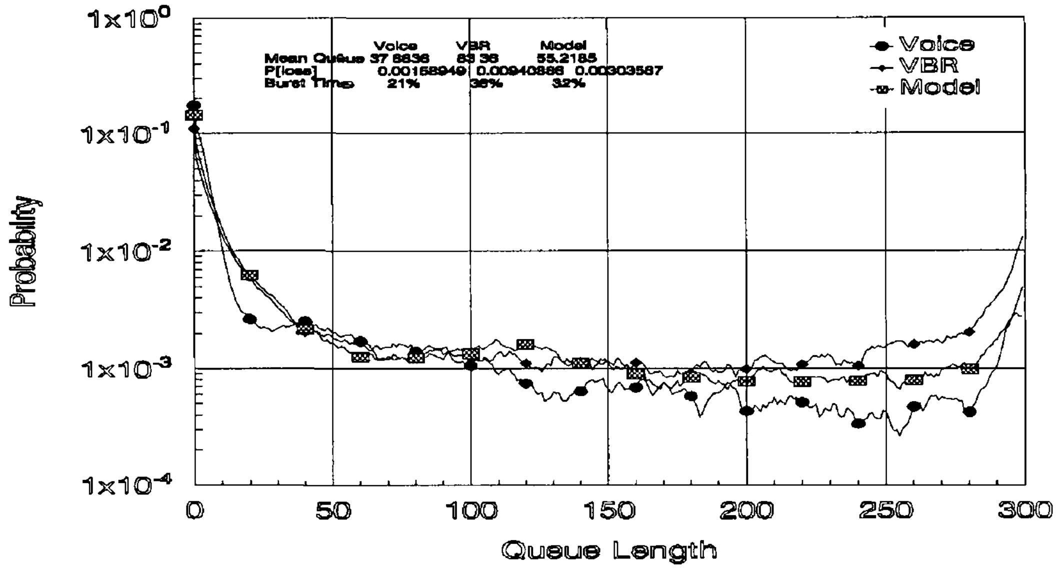


Fig 6 17 Queue Length pdf for $\rho \approx 0$ 95

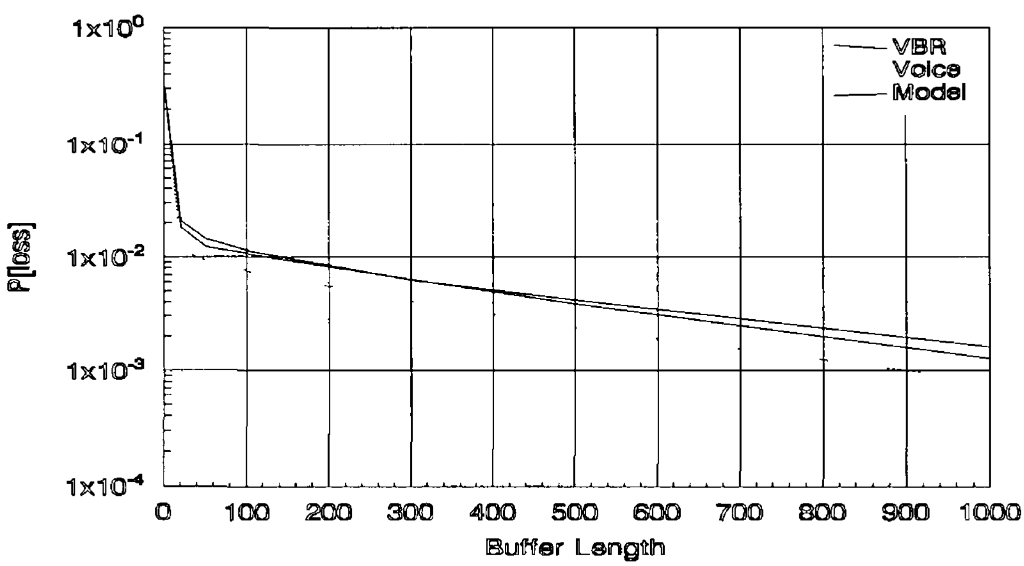


Fig 6 18 P[loss] versus buffer length for different models

6 3 2 Model Convergence

One important aspect of behaviour is the sensitivity of the system to initial conditions This affects whether radically different behaviour will occur given a particular starting point This has a bearing on simulation runs since no general conclusions can be drawn from randomised runs if the system exhibits gross sensitivity to particular states The issue is, how quickly does the model converge to a reliable result?

Another way of looking at this is to examine the memory in the system in order to determine how long a particular initial condition persists In all the simulation runs the system was deliberately kicked off in a (random) state which was close to the expected steady-state condition (This is not strictly true All runs began with a queue length of 0 which would simulate the state of the system after a prolonged period of sub-burst mode activity This is acceptable where burst mode is transient) As part of the simulation run a periodic check was kept on the error between the current pdf of the queue and one taken at a previous instant (in practise this was done ten times in every simulation at regular intervals) For a stable system it would be expected that the error between the current picture and the previous one would decrease at first and then fluctuate around a steady (small) value The overall error (that is the error between the initial check and the current one) will tend to decrease to zero over time The rate of decrease of this error gives an indication of the sensitivity of the system, or the rate at which the simulation converges to a reasonably reliable value

Fig 6 19 shows sample convergence rates from two CBR simulation runs, one for 1,000,000 ticks (2 7s of real time) and one for 10,000,000 ticks (27s of real time) Here the total error is plotted against the number of tenths of execution time It is interesting to note that although the error rate is comparable after 2 7 secs in both runs, there is not a significant improvement by running the system for a further 24 seconds The initial, shorter run, is sufficient to establish the main trends of the behaviour This is borne out by fig 6 20 which shows the actual pdf produced by these two runs It can be seen that the shorter run produces an accurate picture of the pdf, but that the longer run "smoothes" it out Of course, the length of the run determines the number of samples and therefore the reliability of the pdf information For 1,000,000 samples probability values of 10^{-5} and below cannot be considered reliable

Because the 3-state model uses a completely different algorithm from the straight simulation it is worth checking the convergence rates for this model Fig 6 21 shows that the convergence behaviour is similar to the straight model

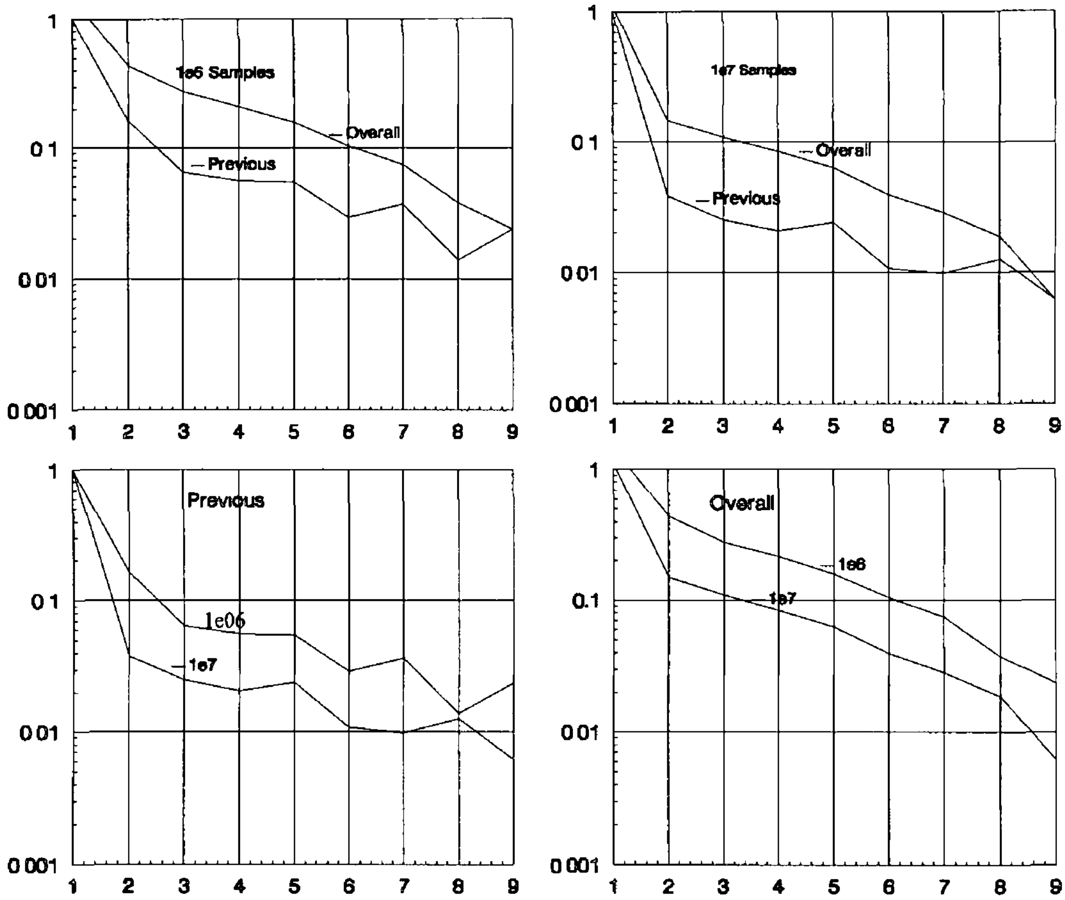


Fig 6 19 Convergence behaviour of the CBR simulation The x axis is the number of blocks of simulation executed where 1 block is the total number of samples/10

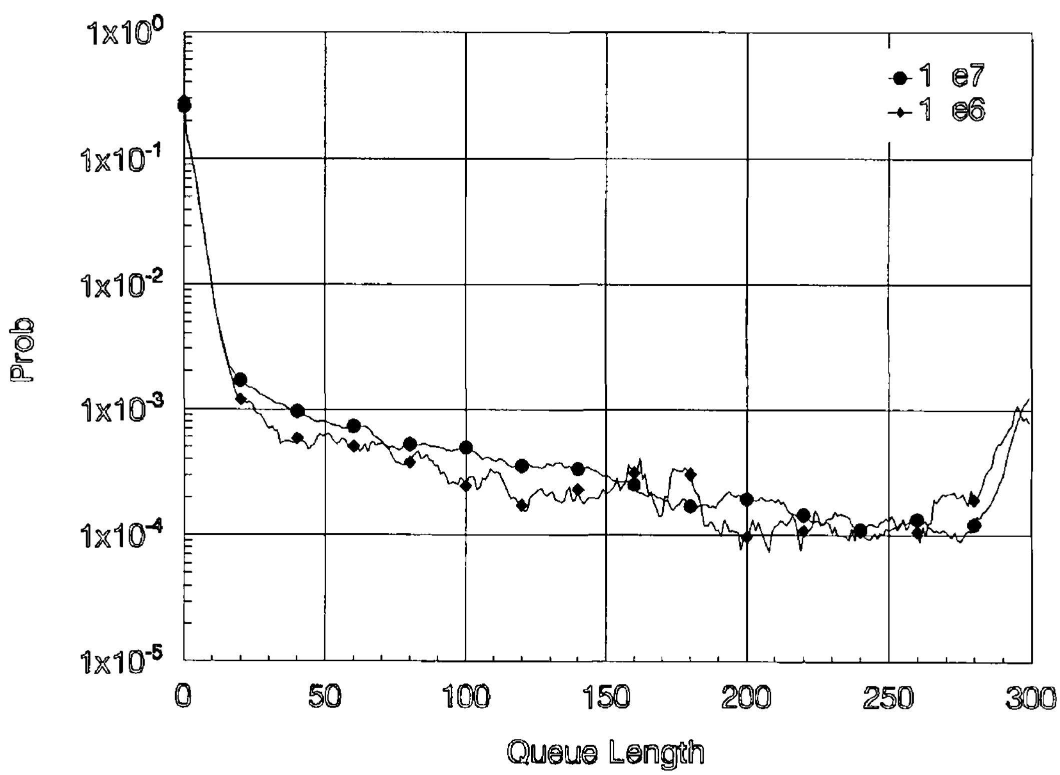


Fig 6 20 Comparative pdf for 2 7s and 27s of simulation time

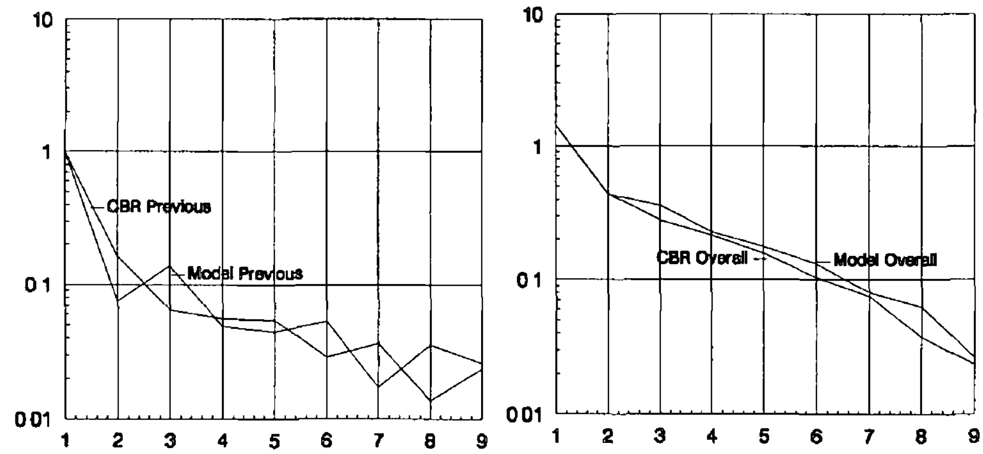


Fig 6 21 Comparison of model to CBR convergence

7. Conclusions and Further Study

7.1 Conclusions

7 1 1 Statistical Multiplexing

This study has concentrated on the question of statistical multiplexing as a way of enhancing system performance The question of whether statistical multiplexing will be a feature of ATM networks is still not resolved by the CCITT standards committee There needs to be further research into all the issues involved However, this study has shown that statistical multiplexing is feasible under certain conditions

To address these issues this study concentrated on simulating an idealised network node whereby *N* high capacity inputs, each carrying a single call, are multiplexed onto one output of the same capacity The multiplexer was assumed to have a single buffer operating as a FIFO queue

This multiplexer was simulated in software (C++) utilising several varieties of idealised source The source characteristics were chosen to reflect the behaviour of normal voice By analysing the voice source it can be seen that a single high capacity (155Mbit) link can carry over 2000 active voice lines simultaneously With statistical multiplexing this implies that a theoretical maximum of over 6000 on-line calls could be supported by a single multiplexer node The simulation runs were designed to examine overall system trends as opposed to exact performance figures

It has been shown that a superposition of such sources can be modelled as a Markov chain, but that such models tend to be intractable for realistic situations due to the large state-space involved The standard method of overcoming this problem is to apply these models to a smaller state space and try to extrapolate the results to the more realistic situation

Three questions were addressed in the study

What are the problems of buffer resourcing for such a multiplexer?

How does such a multiplexer perform under various conditions?

What are the implications of extrapolating results from small models to large models?

As a further issue one particular model was compared to the simulation results and proved to offer possibilities for giving an insight into system performance

7 1 2 Network Resourcing

In chapter 3 an overview of the many varieties of model for the multiplexer was provided These models range through various degrees of complexity and strive to characterise the system performance in terms of cell loss probability or mean cell delay It was shown that there are two components of the buffer queue, the cell component and the burst component

In chapter 4 a way of estimating buffer lengths for high traffic conditions was presented Although greatly simplified, this approach demonstrates that it is unrealistic to attempt to buffer the burst component queue when there is a very large number of inputs

The conclusion drawn on the issue of network resourcing is that it is of no practical benefit to build buffers into multiplexer nodes on the B-ISDN When applying statistical multiplexing the number of inputs should be chosen so that the probability that only S out of N active calls will be in talkspurt at a given time is less than the desired cell loss probability of the node Buffering sufficient to cope with cell queuing must be provided, but these buffers are relatively short

7 1 3 System Performance

When considering system performance the sensitivity of behaviour to various source characteristics was examined with the following results

Load The mean queue length and cell loss probability is highly dependent on traffic load For loads up to about 0 85 the system operates in a predominately sub-burst mode and cell loss/cell queues are negligible For higher loads excursions into burst mode take place and some cell loss or queue delay is likely However, reasonable system performance is possible up to loads of 0 97 Further, performance at high loads improves when the number of input lines is very high (given that the intensity of each individual line has decreased proportionately)

Burst Length By burst length here, we mean the average length in time of an active spurt The intensity of cells per spurt remains constant, but the mean number varies It was found that the system performance is relatively insensitive to variations of this kind in the traffic as long as overall traffic intensity remained the same In chapter 3 some of the problems with defining voice characteristics were presented In particular, defining the burst length is very difficult and dependent on many factors The results of the simulation show that, so long as overall activity and intensity is constant, the details of the voice source are not important A multiplexer will work just as well in America as in Japan, despite the vastly different voice characteristics!

Buffer Length System performance is affected by buffer length, but the performance improvement is not proportional to the buffer size Running a system with no buffer produces very large cell losses due to sub-burst losses Once a small buffer (20% of S typically) has been provided subsequent buffer size increases produce relatively little performance improvement This behaviour becomes more pronounced under higher loads

Traffic Mixture The system performance is somewhat sensitive to traffic mixture Although performance given homogenous sources is easy to control, the addition of high intensity sources adversely affects system throughput This effect is most pronounced for very high intensity sources (with interarrival times of 10s) and fades away as the intensity increases This is a very important issue for the implementation of statistical multiplexing Since ATM has the advantage of being flexible and of maximising bandwidth it is important that sources can be efficiently mixed A further problem is that the total number of high bandwidth sources which can be supported is relatively low and the advantage of statistical multiplexing disappears This is the biggest obstacle to implementing statistical multiplexing in the B-ISDN

Modelling The model examined in the simulations is the Markov 3-state model for a single line This model differs significantly from the reference source in that the cell interarrival time is not deterministic, but is geometrically distributed with a mean intercell arrival time equal to the deterministic intercell arrival time of the reference source

The model was found to mirror the behaviour of the simulation closely, although with consistently higher loss probabilities Its sensitivity to the various parameters was comparable This points to the possibility of using the model to obtain an upper bound on the critical system performance benchmarks such as cell loss and mean delay

7 1 4 Scaling Issues

The issue of scaling is an important one Early literature concentrated on the small scale systems (with N in the order of 100 and an S of tens) which were common then The large scale systems to be implemented cannot practically be simulated The tendency is to extrapolate the small scale results to large systems In this study the system sensitivity to scaling was examined Here the traffic parameters were scaled so that the mean intensity remained the same, and the mean number of cells per active period remained the same N was scaled up with the intercell arrival time Two results were obtained

Firstly, the sub-burst (cell component) behaviour of the system is relatively insensitive to this type of scaling In fact, sub-burst mode queues remained consistent at all times

Secondly, the burst mode behaviour is highly dependent on the scaling mechanism used For a given load, the system may perform mostly in burst or sub-burst mode depending on S (not the ratio of N/S) To preserve the burst mode behaviour it is necessary to scale N non-linearly with S, preserving the probability of burst mode occurring, which will affect the mean load

The result of this is that care should be taken when using results from the literature To characterise system behaviour it is not sufficient to speak in terms of mean load, or system gain It is also necessary to specify the traffic intensity When extrapolating from a small scale system to a realistic system this must be borne in mind However, since system performance generally improves as size increases this is not a critical problem

One way that scaling does become a critical problem is the problem of mixing source intensities described above This results in a complex mixture of burst and sub-burst behaviour High intensity sources will behave with the characteristics of a small system, low intensity sources will behave with the characteristics of a large system

7 1 5 Summary

The results of this study are as follows

Performance A statistical multiplexer is sensitive to source intensity, the input traffic mixture and the offered load It is not sensitive to variation in the mean burst time so long as overall activity remains the same Sensitivity to buffer length decreases with increasing number of inputs (as long as input intensity is reduced accordingly)

Resourcing It is not feasible to provide buffers in large statistical multiplexers which will handle overload conditions For large systems the probable length of an overload burst, coupled Supporting mixed traffic types in a statistical multiplexer will be very difficult whereas homogeneous sources will be relatively easy The potential multiplexer gain for large systems is close to the theoretical maximum

Models The proposed 3-State model provides a means of assessing multiplexer performance which is close to the performance expected of voice traffic Although the trends are the same the actual cell loss values are higher which could allow the model to act as a means of determining an upper bound on performance

The conclusion drawn in this study is that statistical multiplexing is feasible for homogeneous sources, or sources of low intensity The system performance is stable and therefore network quality of service can be guaranteed Nodes should be designed such that the probability of more than S lines beings on-line and carrying a talkspurt at any one moment is less than the desired node cell loss probability, where S is the mean cell interarrival time for the sources

The issue is much more problematic when mixed intensity sources are considered One solution might be to utilise the Virtual Circuit capability of ATM and assign different virtual circuits to sources of equal traffic intensity The low intensity traffic could be statistically multiplexed onto a single virtual circuit with standard full capacity multiplexing used for the high intensity traffic

Finally, the study has shown that results from small scale models can be extrapolated to large models once the effect of the central Limit theorem is taken into account

7.2 Further Study

7 2 1 Simulation Enhancement

The simulation code used is stable and efficient The use of object oriented design allows reusability of the code and integration mto other object oriented programs

Although the "brute force" simulation is reasonably fast it is still impractical for large simulations (2000 lines or more) due to time and memory problems Although the memory problems can be overcome (using paging) the resulting run times will still be in the order of days to obtain absolute cell loss figures for realistic systems One method to overcome this problem would be to utilise one of the several techniques for increasing the occurrence of rare events These techniques are not trivial but would greatly enhance simulation performance

The 3-state model implementation is much slower than that for the straight simulation The implementation has been optimised as much as is practical using a code profiler The slow run time is primarily due to the high number of call to the random number generator The random number generator used has been optimised as much as possible but still utilises a substantial number of floating point operations If another, more efficient, generator is to be used care must be taken to ensure that its period can cope with the runs envisaged Chapter 4 discusses ways to estimate the number of generator calls required by each simulation run One option might be to implement the random number generator used m assembly code, but the resulting efficiency gain might not be very significant Some slight improvement would be gamed by replacing all calls to the generator by the generator code itself Again the improvement is unlikely to be significant

The code can be easily used to examine more complex multiplexer situations An example would be a multi-buffer multiplexer, where each buffer serves a collection of homogenous sources This would require adding a new object which consisted of a collection of the defined multiplexer objects The new object would then service each buffer m turn

The code as it stands has no useful user interface The objects provided form a basis for developing a tool for system resourcing by providing the means of simulating source and traffic mixes A user interface would have to be provided for this, but the objects themselves are flexible enough to cope with the variety of situations that the user could desire

7 2 2 Areas for Further Study

The most complex area encountered in the study was the problem of mixed traffic performance Because of the huge range of traffic possibilities and mixtures it is difficult to identify the most systematic approach which would be fruitful However, this area is very important for the practical implementation of multiplexing

Although some variable bit rate traffic was considered this category will be an important one for B-ISDN networks In particular, video sources will provide high intensity VBR traffic It seems likely that statistical multiplexing of this traffic will not be possible due to the intensity and the relatively low number of sources a single link can support but this area requires further investigation

The 3-State model could provide insights into multiplexer behaviour It will be necessary to carry out simulations which obtain the absolute values for cell loss under certain fixed conditions and then test the model using these