PATTERNS OF CELL LOSS IN ATM MULTIPLEXERS

BY

PHILIP BRANCH BSc, ADipBus, MACS

SUBMITTED TO THE FACULTY OF ENGINEERING
IN PARTIAL FULFILLMENT OF THE DEGREE OF MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

UNIVERSITY OF TASMANIA

NOVEMBER, 1993

DECLARATION

This thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution and that, to the best of my knowledge and belief, the thesis contains no material previously published or written by another person, except when due reference is made in the text of the thesis.

Philip Branch

ABSTRACT

Keywords: B-ISDN, congestion, ATM, statistical multiplexer, cell-loss.

Cell loss is often assumed to occur at random. However, when an ATM cell is lost because of buffer overflow at a statistical multiplexer, there is an altered probability that the cells following it will be lost.

In this report, conditional probabilities of cell-loss due to congestion are modelled mathematically, and then simulations used to verify the models are described. Conditional probabilities are modelled and simulated using M/M/1/K, M/D/1/K, M/D/1/K with minimum interarrival times and SPP/D/1/K queues.

Results from the models and simulations show that cell loss exhibits 'bursty' behaviour. If one cell is lost, the probability of following cells being lost is substantially higher than the overall cell loss probability. Consequently, cell loss has to be characterised by more than just an average rate. The average and variance of the number of cells lost are proposed as additional statistics to characterise cell loss caused by buffer overflow. In this paper these statistics are calculated and tabulated for M/M/1/K and M/D/1/K multiplexer models.

Some of the repercussions of the bursty nature of cell loss for Broadband services are discussed.

ACKNOWLEDGMENTS

I would like to thank the following people for their assistance and support during the research and preparation of this thesis. My supervisor, Dr David Lewis; my wife, Miss Ann Ridgeon; and Mr Marc Stoksik and Dr Richard Lane for help with 'Mathematica'.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Scope of Project.

In this report, ATM traffic and congestion control issues are surveyed before examining in detail patterns of cell loss in an ATM multiplexer and its consequences.

Once overflow occurs in a node, the probability of the following cells being lost is high, because the buffer is close to full. This results in a clustering of cell losses due to buffer overflow. Although of importance in traffic management, a literature search suggests this area has not been investigated in any depth. Only two papers have been found that refer to it. The first [Ramaswami] discusses conditional probability of packet loss where arrivals are cyclic, while the second [Ohta and Kitami] discusses it briefly in the context of Forward Error Correction schemes.

The report is structured as follows: Chapter 1 surveys ATM traffic management. Chapters 2, 3, 4, 5 and 6 describe mathematical and simulation modelling of ATM multiplexer cell loss caused by buffer overflow. Chapter 7 is the conclusion, which is followed by a bibliography and appendices. Appendices A to D are derivations of some of the formulae used. Appendices E to F are listings of SIMSCRIPT programs used for simulation modelling of M/M/1/K, M/D/1/K, M/D/1/K with minimum interarrival times, and SPP/D/1/K respectively.

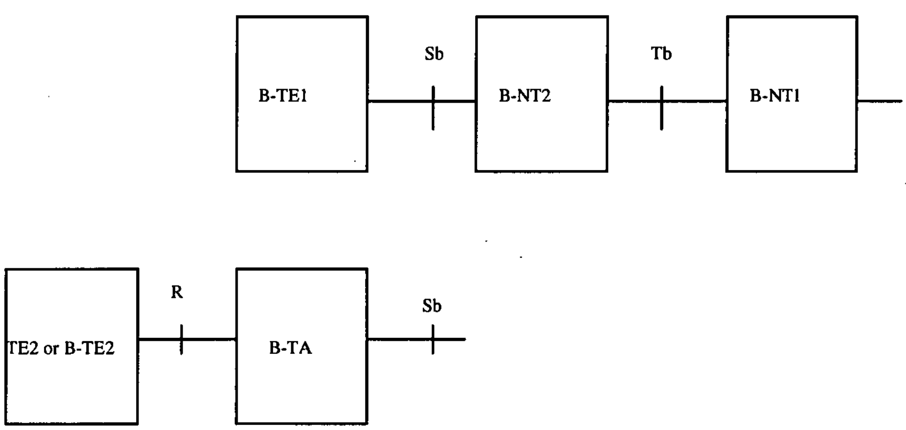
1.2 B-ISDN and ATM

1.2.1 B-ISDN

B-ISDN is an abbreviation for Broadband Integrated Services Digital Network. B-ISDN will provide digital services requiring a broad range of bandwidths, from narrow band services such as voice to wideband services such as High Definition Television. B-ISDN will make possible high speed image transmission, interactive video, and other exotic services as well as providing established services like voice, telemetry and data transfer. B-ISDN is possible because of advances in fibre optics, semiconductor manufacturing, reliable programming and increased demand for high bandwidth services. Fibre optic cable can provide bit rates in the hundreds of Megabits to Gigabits range. B-ISDN is expected to evolve as a backbone service for local, metropolitan and wide area networks and eventually be extended to the office desktop and the home.

The following figure illustrates the B-ISDN reference configuration [Handel]. It is similar to the Narrowband ISDN reference configuration.

Figure 1. B-ISDN Reference Configuration



B-TE1 is broadband ISDN Terminal equipment. TE2 or B-TE2 is non ISDN equipment. B-TA is broadband Terminal adaptor equipment for non-ISDN equipment. B-NT1 and B-NT2 are the network terminators. B-NT2 need not be present. For example B-NT2 might be a customer network that interfaces to a B-ISDN network. R, S_b and T_b are the interfaces between these equipment types.

1.2.2 ATM Networks

Asynchronous Transfer Mode has been chosen as the mode of transmission of B-ISDN. ATM has evolved from the DQDB MAN. ATM is a universal, packet service. ATM packets, called cells, are a uniform 53 octets in size. The small size of the cells makes ATM flexible enough to emulate any service, as well as simplifying switching.

ATM can make more efficient use of the available channel capacity. ATM allocates cells to a service as the service requires. Some B-ISDN services tend to be bursty with periods of high activity interspersed with periods of low activity. A large bandwidth service may take up a large proportion of a channel's capacity while transmitting, but if it is a 'bursty' service it may have long periods of relative inactivity. By comparison, a service using Synchronous Transfer Mode (STM) is allocated a channel of fixed bandwidth regardless of any variability in its requirements. A constant bit rate service using STM will be efficient in its channel use, but a bursty service using STM will not be, as the channel's capacity must meet the peak requirements of the service. With a bursty service, this peak is reached only occasionally, so some of the channel is unused most of the time.

ATM networks are high speed, fibre optic based networks. At the User Network Interface (UNI), an output virtual channel of 622.08 Mbits/s and an input Virtual Channel of 155.52 Mbits/s is proposed. Multiplexers provide the UNI and connect the links of the network. Buffering of cells may occur at a multiplexer to cope with short term traffic surges.

The following figure shows the protocol reference model for ATM networks. The relation between the ATM and the OSI reference model has not been clearly defined yet, however the PHY layer corresponds to layer one of the OSI model while the AAL and ATM layers correspond to the lower edge of layer two of the OSI model.

Figure 2. ATM Protocol Reference Model Sublayers (DePrycker)

Convergence	CS	AAL
Segmentation and Reassembly	SAR	
Generic Flow Control Cell header generation / extraction Cell VPI / VCI translation Cell Multiplex and demultiplex		ATM
Cell rate decoupling HEC sequence generation /verification Cell delineation Transmission frame adaption Transmission frame generation /recovery	TC	PHY
Bit timing Physical medium	PM	

The ATM Adaptation Layer maps higher level data units into cell payloads. There are four AAL services, classified by the need for timing between source and destination, bit rate, and connection mode. This is illustrated in the following diagram.

Figure 3. Service Classes for AAL (DePrycker)

	Class A	Class B	Class C	Class D
Timing between source and destination	Required		Not Required	
Bit rate	Constant	Variable		
Connection mode	Connection oriented			Connectionless

For traffic classes A, B, C, and D there corresponds an AAL type 1, 2, 3, and 4.

Class A traffic is connection oriented, constant bit rate, and there is a fixed timing relationship between the source and the destination. Variability in the delay needs to be small. An example of Class A traffic is PCM voice traffic at 64 kbit/s.

Class B traffic is similar to Class A, but has a variable bit rate. An example of Class B traffic is variable bit rate video.

Class C traffic is also connection oriented, variable bit rate, but there is not a timing relationship between the source and destination. An example of type C traffic is connection oriented data transfer.

Class D traffic is similar to Class C, but is connectionless.

The AAL has two sub-layers, the Convergence Sub-layer and the Segmentation and Reassembly Sub-layer. The convergence sublayer has different functions, depending on the service. It may have Forward Error Correction, handling of lost cells or timing information. These functions are implemented through prefixing and suffixing of a header and trailer.

The SAR sublayer's function is segmentation and reassembly of convergence sublayer program data units (CS-PDUs) into ATM cell payloads. The CS-PDU is split into service data units whose length depends on the AAL type. The mapping of CS-PDUs to ATM cells is discussed in the next section.

The ATM layer receives an SAR-PDU from the AAL layer which it prefixes with a cell header to form an ATM cell.

The ATM layer deals in units of cells. Its four functions are Cell multiplexing / demultiplexing; Cell VPI / VCI translation; Cell header generation / extraction; and Generic Flow control. Cells are multiplexed and demultiplexed from different connections (identified by different VCI and / or VPI) onto a single cell stream. At an ATM switch, the VCI or VPI may need translation for transmission to the next node. The cell header is also affixed to (or removed from) the body of the cell at this sublayer. A User Network Interface flow control mechanism using the GFC bits is also implemented at this level.

The following figure is a diagram of an ATM cell at the User Network Interface. GFC is the Generic Flow Control field; VPI is the Virtual Path Indicator; VCI is the Virtual Channel Indicator; PT is the Payload Type; RES is reserved, possibly for future traffic types; CLP is the cell loss priority; HEC is the Header Error Control.

Cells at the Network Node Interface do not have a GFC and have a larger VPI.

The Generic Flow Control field is four bits long and is intended to be used for controlling short term overload at the customer network. Multiple terminals may share a single access link to the network. The GFC is used to control overload from terminals on the same access link.

The VPI is a sixteen bit connection identifier used for routing of the cell. As the cell traverses the network, its value may change.

The VPI identifies an end to end path for the cell. It is eight bits at the UNI and twelve bits at the NNI.

The PT identifies whether the cell contains user or network information. The only value defined so far is user information, with a value of 00.

The CLP bit can be set by the user, or the network. If the traffic is CBR, it is set to zero, indicating a high priority. If it is VBR, it may be zero or one. It may be set by the Virtual Leaky Bucket algorithm or if the traffic is lower priority.

The HEC is a CRC code to protect the header data. Although defined at the ATM level it is used at the PHY level.

Figure 4. ATM Cell Layout at User Network Interface

<table>
<tr><th colspan="8">Bit Number</th><th></th></tr>
<tr><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>Octet Number</th></tr>
<tr><td colspan="4">GFC</td><td colspan="4">VPI</td><td>1</td></tr>
<tr><td colspan="4">VPI</td><td colspan="4">VCI</td><td>2</td></tr>
<tr><td colspan="8">VCI</td><td>3</td></tr>
<tr><td colspan="4">VCI</td><td colspan="2">PT</td><td>RES</td><td>CLP</td><td>4</td></tr>
<tr><td colspan="8">HEC</td><td>5</td></tr>
</table>

The PHY layer is made up of two sublayers: the Physical Medium (PM) and Transmission Convergence (TC) sublayers. The TC sublayer deals in bits provided from the PHY sublayer. Its five functions are Cell rate decoupling, HEC sequence generation and verification, Cell delineation, Transmission frame adaptation, and Transmission frame generation / recovery. The PM sublayer is responsible for the transmission and reception of bits. The two functions it performs are bit timing and specification of the actual transmission medium.

1.2.3 Mapping of Services onto ATM.

We now describe how Convergence Sublayer Program Data Units (CS-PDUs) of each traffic type are mapped onto ATM cells.

For type 1 traffic each CS-PDU is split into 47 byte SDU payloads that are prefixed with a one byte of sequence information.

For type 2 traffic the CS-PDU is split into SAR payloads that are then prefixed by a sequence number and Information type and is suffixed by a length indication and CRC. The information type may be a BOM, COM, or EOM. All the details of this traffic type have not been resolved.

In type 3 traffic, the CS-PDU is split into 44 byte SDUs and prefixed by a two byte header and a two byte trailer. The header contains segment type, sequence number and multiplexing information. The trailer contains a length indication and a CRC.

The segment type can be a BOM, COM, EOM or SSM (Single segment message).

Type 4 traffic uses the same format as type 3 traffic, although the MID field is used differently.

Each 48 byte segment constitutes the payload of an ATM cell. This cell is then prefixed with a five byte header.

Voice is sampled and mapped one sample per cell as in type 1. There is no complex assembly or reassembly.

Video traffic is transmitted using the Discrete Cosine Transform and the MPEG protocol. A reference frame is constructed using the DCT which is then segmented as for type 2 traffic. Successive frames are either transmitted as differences to the predicted frame or as a new reference frame, depending on which results in the least traffic. The MPEG protocol specifies every fifteenth frame as a reference frame, and as many previous frames as wanted can be used to predict the next frame. If cells are lost, the receiver can predict where the next frame is likely to be. Consequently, excessive cell loss can result in poor picture quality. There are no requests for a refresh.

Data traffic is mapped as described for type three or four traffic depending on whether it is connection oriented or connectionless.

1.3 ATM Traffic

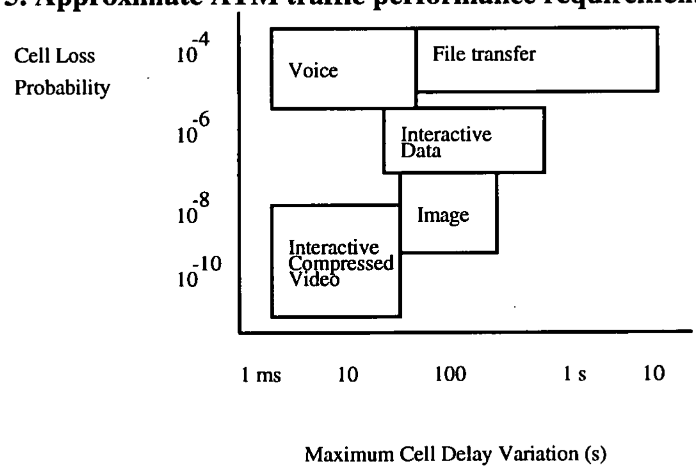
The choice of ATM as the transport mode for B-ISDN reflects the bursty nature expected of B-ISDN services. Data, video and images are all bursty. Table 1 [DePrycker] shows some typical values of burstiness. ATM can take advantage of this variability by statistical multiplexing. In statistical multiplexing, the sum of the peak input rates is greater than the carrier's output capacity. However, because of the burstiness of the inputs, the average input rate is less than the carrier's capacity, so a better throughput is obtained than with STM.

Table 1. Broadband Services: Bit rates and Burstiness [DePrycker]

Service	E[s(t)]	B
Voice	32 kbit/s	2
Interactive data	1-100 kbit/s	10
Bulk data	1-10 Mbit/s	1-10
Standard quality video	20-30 Mbit/s	2-3
High definition TV (HDTV)	100-150Mbit/s	1-2
High quality video telephony	~2 Mbit/s	5

The use of ATM has ramifications for network design, admission procedures, traffic management and fairness of access to users of the same service. Because ATM traffic is heterogeneous, different and sometimes conflicting service requirements are needed. For example, voice traffic is susceptible to delay but can cope with some cell loss, whereas data is susceptible to cell loss, but delay is less important. Figure 5 [Hong et al] shows some approximate ATM traffic performance requirements.

Figure 5. Approximate ATM traffic performance requirements (Hong et al)



Voice is PCM voice traffic. The standard provides for operation at 64 kbit/s, however with DPCM and ADPCM, it can be reduced to 16 kbit/s. Voice traffic can tolerate high cell loss rates but is susceptible to delay.

File transfer refers to the transfer of bulk data over communications links. Examples are new software releases, and backups. The time taken for file transfer is usually unimportant, but the error rate is, so retransmits are acceptable if cells are lost.

An example of interactive data is a database query. Because it is interactive, delay is less acceptable than with file transfer, but some can be tolerated. As with file transfer some error can be accepted, and corrected through cell retransmits.

With image transmission, (eg. CAD/CAM) the delay requirements are similar to interactive data, but image has more stringent error requirements, although some can be tolerated and disguised through DCT coding and extrapolation. Retransmissions may be acceptable.

Interactive compressed video (eg. Videophone) operates in real time, so has stringent delay and error requirements. Retransmission of lost cells is not possible, and error correction options are limited, since most of the redundancy has been removed from the images. DCT coding and extrapolation are the only methods of correcting errors.

Statistical multiplexing implies a finite probability that the input services demand will exceed the output capacity. Consequently buffering at the multiplexer is needed to cope with short term high inputs. However, for design simplicity and to minimise delay for delay sensitive services such as voice, the buffer must be small. Consequently, there is a finite probability that a cell will be lost because of buffer overflow. The probability of successive cells being lost after one cell is lost is the topic of the remaining chapters of the paper.

The following table shows two traffic descriptors of ATM traffic and some of their typical values. These are the average bit rate and the burstiness, where burstiness is defined as peak bit rate divided by average rate. Other traffic descriptors are: Average Burst Length, Peak Bit Rate, Homogeneity, Heterogeneity and Offered Load. The effects of a change in these parameters has been summarised [Bae] as follows:

- As average burst length increases, cell loss and delay decrease, provided the average rate remains constant;
- As peak bit rate increases, cell loss and delay increases provided the average rate remains constant;
- As the number of homogeneous sources increases, the cell loss decreases provided the average rate remains constant;
- With a heterogeneous load, the effect of the high rate traffic components dominate the cell loss and delay statistics;
- As offered load decreases, cell loss decreases, although not instantaneously.

1.4 ATM Service Categories

The quality of service in ATM traffic can be defined at the connection level, burst level and cell level. At the connection level it can be expressed in terms of connection blocking probabilities; at the burst level, in terms of burst blocking probabilities and at the cell level in terms of cell loss probabilities and cell delay.

Initially, only two qualities of service are to be defined, both at the cell level. Variable Bit Rate traffic will be statistically multiplexed with some cell loss and delay. It will be cheaper than the other service, the Continuous Bit Rate. CBR traffic will have reserved bandwidth. Consequently it will have low cell loss and delay. To identify each quality, the Cell Loss Priority (CLP) field of the cell header will be used. For CBR traffic, the CLP will be 0. For VBR, the CLP may be 0 or 1, depending on the importance of the cell (as in DCT coding) and on whether the cell has been marked by the Virtual Leaky Bucket algorithm as having exceeded the contracted bandwidth.

Qualities of service to be introduced later are VBR Reserved Connection Bandwidth and Reserved Burst Bandwidth. VBR Reserved Connection Bandwidth will be statistically multiplexed, but with a lower cell loss and delay than VBR.

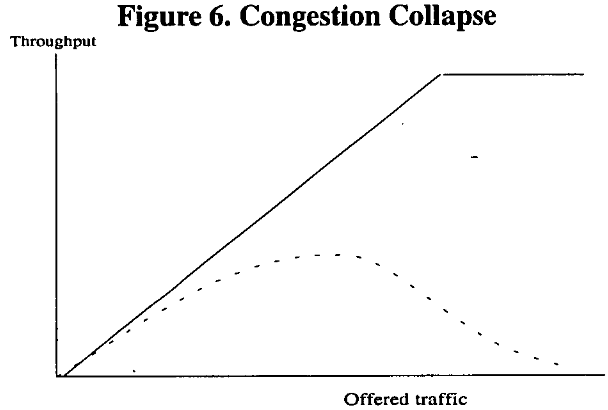
Reserved Burst Bandwidth Service will reserve bandwidth before sending a burst into the network. To reserve the bandwidth, it will send a request into the network for the burst bandwidth. If available, the bandwidth will be made available to it. Otherwise it must request the burst bandwidth again. After the request is successful and the burst is transmitted, the user will send an end-of -burst indication into the network to deallocate the reserved bandwidth.

1.5 Traffic and Congestion Controls

1.5.1 Congestion Collapse

As the load on a network increases, it would be desirable if the throughput increased linearly up to the maximum, and then stayed at the maximum beyond that. Also, it would be desirable if delay stayed within acceptable limits. Unfortunately, what tends to happen as the load increases is 'congestion collapse'. The throughput increases as the load increases, but as the load increases, so does the delay, causing timeouts. The timeouts may cause retransmits, exacerbating the congestion. Beyond a certain point, the throughput starts to decline, and may collapse entirely under the weight of retransmits and timeouts. This is congestion collapse.

Figure 6. Congestion Collapse



Performance collapse of the network caused by congestion is one of the main areas of network research. It is a problem that requires dynamic rather than static solutions. Static approaches such as multiplexers with large buffer space, high speed links and fast processors will not prevent congestion. An increase in buffer space will increase the probable delay at a node, affecting the performance of delay sensitive services such as voice. Installing high speed links and processors may cause mismatch between source and sink capacity resulting in greater overall traffic due to retransmissions. Even with high speed links and processors equally matched, congestion may still occur if several nodes direct traffic to the one node at the same time. For these reasons, congestion needs to be dealt with dynamically using protocols that specify what to do when congestion is detected rather than statically in a vain attempt to prevent it. In particular buffer overflow is a symptom of congestion and not its cause.

Some difficulties of congestion control are that protocols have to have a low overhead; be simple, as they will be implemented in hardware; be fair to users of the same service; be responsive; be robust; and maximise the overall network throughput. In ATM networks remedial action to deal with congestion needs to be local as many of the assumptions on which other network's congestion strategies are based are no longer true. In particular, the ratio of admission to propagation time is significantly large. A large fibre optic network may have 40 Mbits of data in transit. [Woodruff]

Congestion occurs when total demand exceeds available resources. When this happens, the available resources can be increased or demand can be reduced. Demand can be reduced by service denial or service degradation. Depending on the nature of the service, one of these options is appropriate. Service denial is appropriate for connection oriented services such as voice, while service degradation is appropriate for packet oriented services such as data.

1.5.2 Preventive Controls

ATM preventive controls fall into two general categories; admission controls and bandwidth enforcement. Of the two, admission controls are the most promising.[Bae and Suda].

A call is admitted if its service requirements can be met by the bandwidth that can be allocated to it. Effectively, a service contract between the network and the caller is made. The caller undertakes to transmit traffic within specified limits while the network will deliver sufficient bandwidth to provide a service of a given standard.

The most difficult aspect of admission control schemes is determining the equivalent bandwidth of a call, and hence whether the call can be admitted. Traffic attributes used in making this decision are called traffic descriptors. The most important traffic descriptors in deciding to admit a call are the peak bit rate, average burst length and burstiness of the call and what cell transmission delays and cell loss probabilities are tolerable.[Hong]. Some work has been done using Expert Systems to predict the bandwidth requirements of the call [Erfani], and using Artificial Neural Networks in deciding on call admission [Hiramatsu].

Bandwidth enforcement has been proposed by means of the leaky bucket algorithm. Conceptually, this works on the idea of a pool of tokens generated at the average throughput of the network up to some maximum. The maximum is the maximum burst length. As a cell is admitted to a node, it claims a token from the network. If no tokens are available, the cell is discarded. The effects of the leaky bucket algorithm are excessively stringent, so it has been modified to the Virtual Leaky Bucket algorithm, and is used as a reactive control.

1.5.3 Reactive Controls

Once congestion occurs the network must react to reduce it. Congestion manifests itself as timed out cells, long queuing times at nodes and buffer overflow. Congestion can be dealt with at the call level on an end to end basis much as happens in the X.25 packet level or it can be dealt with at the cell level.

Because of the high propagation time to admission time ratio, call level attempts at throttling back traffic have been found to be too slow and unstable, and hence ineffective. Because so much data is in transit, reducing traffic loads at the source may occur too late to prevent the worst effects of the congestion.

More promising is the cell level reactive control, the Virtual Leaky Bucket algorithm. This is a modification of the Leaky Bucket algorithm. It works in the same manner as the Leaky Bucket algorithm in that a pool of tokens is generated at the average throughput of the network up to some maximum, where the maximum is the maximum burst length, and as a cell is admitted to a node, it claims a token from the network. However, unlike the Leaky Bucket algorithm, if no tokens are available the Virtual Leaky Bucket does not discard the cell. The cell is still admitted, but it is marked as having violated the bandwidth limitations of the network. Should congestion occur, marked cells can be discarded from the network. The VLB can be implemented simply by a one bit flag in the cell (the Cell Loss Priority) and a counter in the node.

Another cell level scheme is to carry forward in the cell explicit forward congestion information. If there are alternative routes a network node can use a different one that avoids the congestion.

1.5.4 Priority Schemes

Priority schemes are also based on the CLP field in the cell header. Priority schemes attempt to use the fact that some cells are more important than others. For example, in video transmission, Discrete Cosine Transform coding produces data of different perceptual importance. Also, cells which have been marked by the VLB scheme will have a lowered priority. Priority scheduling schemes give priority to important cells in transmission whereas selective discard schemes drop less important cells when more important cells arrive at a full buffer.

1.6 Error Control

As in congestion control, many of the assumptions underlying error control in other networks are not true of ATM networks. In particular the propagation time to admission time ratio is considerably higher and fibre optic error rates are low, about one cell in 10^{11}.

These differences mean that flow and error control need to be independent. This can be contrasted with window based schemes in X.25 where they are not. If a window technique is applied to ATM for both flow and error control, the window would need to be large to make most use of the ATM bandwidth, but it would need to be small to provide error control. The high propagation time to admission time makes this a problem in ATM networks while it is not a problem in lower speed networks.

Schemes proposed have been the ARQ, Go-back-N and Go-back-N with selective repeat. The latter is preferred, although it may entail some reordering of cells at the end terminal. However, it appears to require less overall retransmissions than the other options. The extent of the scheme can be link by link or edge to edge. The latter is preferred because no node protocol processing is required and so is faster overall.

Another approach to error control is Forward Error Control. A scheme called CREG-VP has been devised that uses parity and cell loss detection cells to recover lost cells. [Ohta and Kitami]. Consecutive cell losses can be recovered up to a limit dependent on the number of parity and cell loss detection cells used.

CHAPTER 2

BUFFER OVERFLOW MODELLING

To derive quantitative results for the conditional probability of buffer overflow cell loss, the multiplexer queue is modelled by queues with differing arrival and service distributions.

Many telecommunication processes can be modelled by queues, and queuing theory is an important area of mathematical statistics. Unfortunately it is also a difficult area, and often simplifying assumptions need to be made to make the analysis tractable. Simulation of queues is often used to derive results quantitatively, but unfortunately many of the interesting events about which we would like to collect statistics occur relatively infrequently, meaning very long simulation runs are needed. Buffer overflow occurs infrequently at low server utilisation.

Queues are characterised by their arrival distribution, service distribution, number of servers and buffer size. The notation 'Arrivals/Service/Servers/Buffer' is used to characterise the queue. The M/M/1/K queue is the queue with Markov (random) arrivals, Markov service times, one server and finite buffer space of K. Other queues we examine are the M/D/1/K queue which is the queue with Markov arrivals, constant (deterministic) service times, one server and finite buffer space; the M/D/1/K queue with minimum interarrival times and the SPP/D/1/K queue which is the queue with Switched Poisson Process arrivals, constant service times, one server and finite buffer size. All the queues dealt with in this report are First In-First Out queues.

In this report the M/M/1/K queue is considered first. This is the finite space, single server queue with exponentially distributed arrival and service times. It is considered first because it is the easiest queue model to deal with. Many results are available for this queue and it is mathematically tractable. Once this is analysed satisfactorily, the M/D/1/K queue is considered. This is the finite space, single server queue with exponentially distributed arrival times and deterministic service times. Because cells are a uniform size this is a better model of a multiplexer than the M/M/1/K queue, although the M/D/1/K queue is much more difficult mathematically. The Markov Modulated Poisson Distribution has been suggested as a good model for bursty traffic arrivals [Heffes and Lucantoni]. We consider the two state MMPD called the Switched Poisson Process, in the SPP/D/1/K queue model. Also considered is the M/D/1/K queue with minimum interarrival times. In all these models variable rate traffic is statistically multiplexed. There is no reserved bandwidth traffic.

Mathematical models (or approximation for the minimum interarrival times) of the behaviour of all these queues are described. To verify the models simulations have been constructed and the results from the simulations tabulated with those from the mathematical models. Simulation programs were written in SIMSCRIPT II.5 and ran on an INTEL based 80-486 PC running at 33 MHz. SIMSCRIPT is a general purpose simulation language in which queues can be modelled easily. Arrival and Service times of various distributions are available as library routines, and more complicated distributions (such as the Switched Poisson Process) can be constructed from them. SIMSCRIPT has good reporting facilities.

Simulation models have the following components [Russel]:
- a mechanism for representing arrivals of new objects;
- the representation of what happens to the objects in the system;
- a mechanism for ending the simulation.

In our models the objects are 53 octet cells, the system is a multiplexer queue and the simulation ends when interrupted by the user, hopefully after a steady state has been reached. The multiplexer is represented by a First-In-First-Out queue, which requests access to a server. The cell arrival distribution is simulated by waiting a length of simulated time defined by the arrival distribution, before generating a request for the server. The request for the server causes a SIMSCRIPT queue entry to be made. The server time is simulated in the same way as the arrival time. When the queue is beyond its limit size, the entry is not made on the queue, but is recorded as a buffer overflow error. Runs of errors are recorded in a table. Reporting of errors is done every 100,000 transmitted cells by writing a line of a report to a disk file. By reporting in this way, it is easier to see when the system reaches a steady state, and data is not lost if the processor is accidentally switched off during the simulation.

The computational effort involved in the simulations is considerable, especially at low server utilisation rates less than 0.5. The queue statistics converge more quickly for some of the models than for others. The M/M/1/K models stabilise comparatively quickly, usually within an hour. However, the M/D/1/K simulations require several hours to stabilise and may require a day or more to stabilise if server utilisation is low. The SPP/D/1/K and Minimum interarrival time queues need to be run for at least a day, preferably longer.

Long simulation times are needed because buffer overflow occurs rarely at low server utilisation. Overflow occurs more frequently as the buffer size is reduced. When the arrival process is Markovian, the conditional probability of cell loss is unaffected by buffer length.

Included in the appendices are listings of the simulation programs. They are written in SIMSCRIPTII.

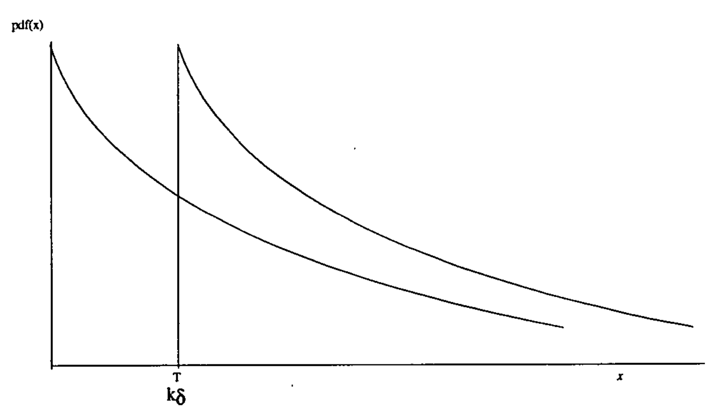
CHAPTER 3

M/M/1/K MODEL AND SIMULATION

3.1 Conditional Probability of Cell loss for M/M/1/K.

The M/M/1/K queue is the single server queue with exponentially distributed arrival and service times, with a maximum of K entries. [Kleinrock]. A useful property of the exponential distribution is its memoryless nature. If an event with exponentially distributed probability has not occurred at time T, the probability of it occurring in the time interval (T, T+x) is the same as the probability of it occurring in time interval (0, x). This is shown in the following figure, where the probability density function of an event at time zero and time T are graphed.

Figure 7. Memoryless Property of Exponential Distribution



In considering the buffer overflow problem, because of the memoryless nature of the exponential distribution, it is unnecessary to know how long it has been since the last service. Let the time until the next service be the random variable y. From the memoryless nature of the distribution, y can be taken to be 0 at the time the cell is lost. The exponential distribution is [Kleinrock]:

$$s(y) = \mu e^{-\mu y} \qquad (1)$$

The exponential probability density function of k arrivals in time x from an arbitrary starting point is given by [Kleinrock]

$$a(x) = \frac{\lambda(\lambda x)^{k-1}}{(k-1)!} e^{-\lambda x} \qquad (2)$$

We assume independence between arrivals and services. Because arrivals and services occur at random in M/M/1/K and the queue is full at the time of the overflow this would seem a reasonable assumption. For statistically independent random variables, the joint probability density function is their product [Haykin]. That is:

$$p_{xy} = a(x).s(y) \qquad (3)$$

The probability of one service in the interval (0, A) and k arrivals in the interval (0, B) is:

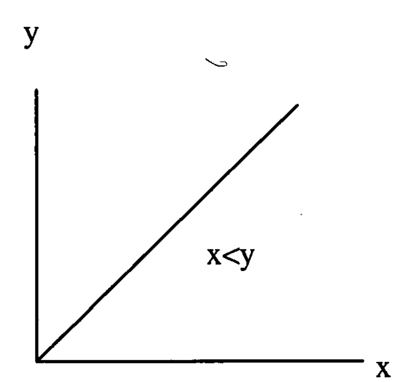
$$\int_0^A \int_0^B p_{xy}(\alpha,\beta)\, d\alpha\, d\beta \qquad (4)$$

To calculate the probability of buffer overflow occurring, we want X (arrivals) and Y (services) such that:

$0 \le X \le Y$ and $0 \le Y \le \infty$

This defines an infinite triangular area on the X, Y plane as shown in the following figure.

Figure 8. Bounds of Integration for M/M/1/K



Integrating over this area gives the probability of k arrivals before one service.

Remembering that one cell has been lost, the above probability is identical to the probability that the burst of lost cells is k+1 long.

$$Pr(burst\ of\ length\ k+1) = \int_0^{\infty}\int_0^{y}\frac{\lambda(\lambda x)^{k-1}e^{-\lambda x}\mu e^{-\mu y}}{(k-1)!}dxdy \qquad (5)$$

This integral is straightforward to calculate using integral tables [Gradshteyn], and can be shown to be:

$$Pr(burst\ of\ length\ k+1) = \left(\frac{\rho}{\rho+1}\right)^{k} \qquad (6)$$

where ρ is the server utilisation, equal to λ/μ.

The following table shows expected probabilities of burst lengths for different values of channel utilisation derived from the above formula, and observed rates from simulation experiments. The values for server utilisation of 0.4 and 0.9 are plotted in the following graphs. There is good agreement between the theoretical and simulation values. Note that the plot of the predicted cell loss is partially obscured.

Table 2. Cell Loss Burst Lengths
M/M/1/K

Burst length (simulated / predicted)

	2		3		4		5	
ρ								
0.4	0.285	0.285	0.081	0.082	0.023	0.023	0.006	0.007
0.5	0.334	0.333	0.117	0.111	0.036	0.037	0.012	0.012
0.6	0.370	0.375	0.140	0.141	0.054	0.053	0.020	0.020
0.7	0.430	0.412	0.18	0.170	0.076	0.070	0.027	0.029
0.8	0.440	0.444	0.186	0.198	0.090	0.088	0.038	0.039
0.9	0.474	0.474	0.223	0.224	0.106	0.106	0.054	0.050

Figure 9. Graph of Predicted and Simulated Cell Loss Lengths
M/M/1/K ρ=0.4

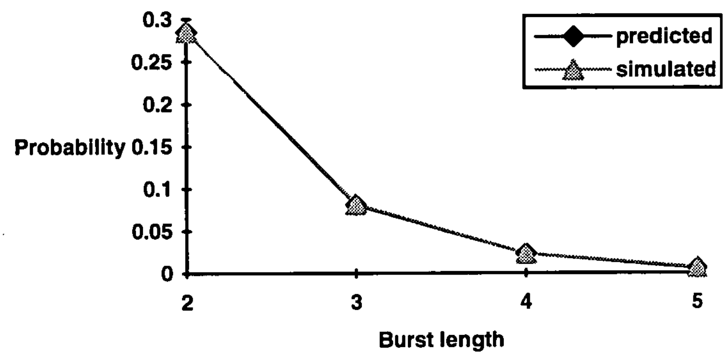
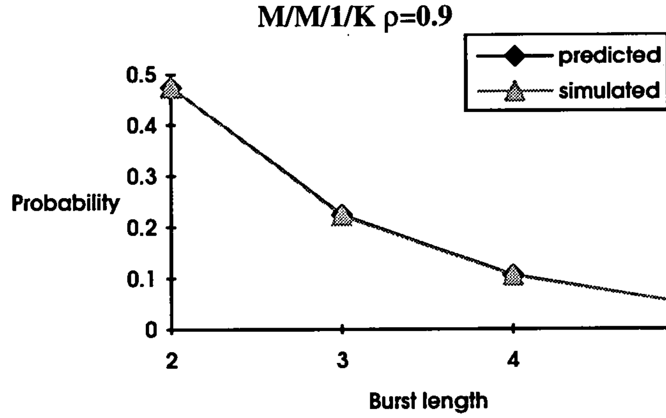


Figure 10. Graph of Predicted and Simulated Cell Loss Lengths
M/M/1/K ρ=0.9



3.2 Average Cell Loss Run for M/M/1/K

Useful statistics in characterising cell loss are the mean number of cells lost in a burst and the variance of the number of cells lost. In this and the next section analytic expressions for these statistics are derived.

The average cell loss length is defined by:

$$\bar{x} = \sum_{k=1}^{\infty} k \ Pr(k \ cells \ lost \mid 1 \ cell \ lost)$$

For the M/M/1 case the probability that the cell loss length is exactly k cells is :

Pr (number of cells lost due to buffer overflow is exactly k)

$= Pr\,(length \geq k) - Pr\,(length \geq k+1)$

$$= (\frac{\rho}{1+\rho})^{k-1} - (\frac{\rho}{1+\rho})^{k}$$

$$= \frac{\rho^{k-1}}{(1+\rho)^{k}}$$

Consequently, the average number of cells lost is:

$$\sum_{k=1}^{\infty} k \frac{\rho^{k-1}}{(1+\rho)^{k}}$$

which is shown in appendix A to be:

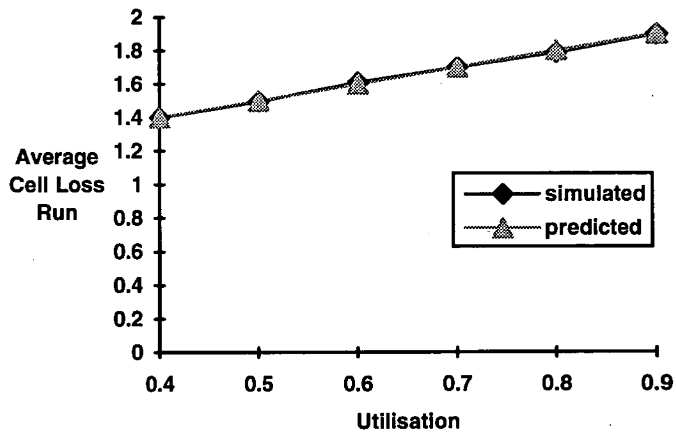
$$1+\rho \qquad (7)$$

The following table and figure show average cell loss lengths for different values of ρ derived from simulation and using equation (7). There is good agreement between the simulation and the formula.

Table 3. Average Cell Loss Simulated and Predicted M/M/1/K

ρ	Simulated	Predicted $(1+\rho)$
0.4	1.399	1.4
0.5	1.497	1.5
0.6	1.611	1.6
0.7	1.700	1.7
0.8	1.788	1.8
0.9	1.898	1.9

Figure 11. Average Cell Loss Simulated and Predicted M/M/1/K



3.3 Variance of Cell Loss Run for M/M/1/K

The variance of the average cell loss run length can be calculated as follows:

$$\sigma^2 = \sum_{k=1}^{\infty} p_k (x_i - \bar{x})^2$$

$$= \sum_{k=1}^{\infty} \frac{\rho^{k-1}}{(1+\rho)^k} (k - (1+\rho))^2$$

In appendix B we show that:

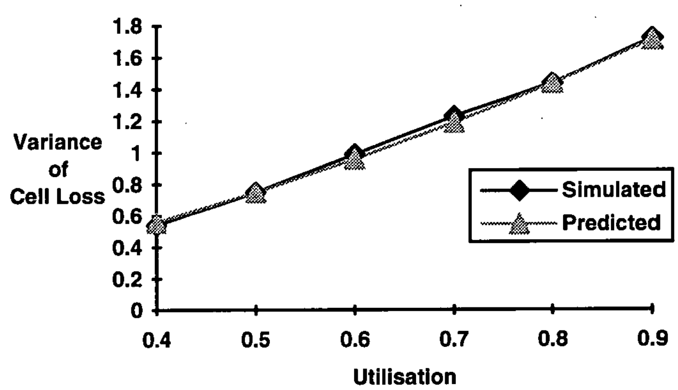
$$\sigma^2 = \rho(1+\rho) \tag{8}$$

The following table and graph shows the variance for the average cell loss for different values of ρ derived from the simulation and equation 17. There is good agreement between the simulation and the data. The small variation at utilisation of 0.6 and 0.7 is due to short simulation times.

Table 4. Variance of Cell Loss
Simulated and Predicted
M/M/1/K

ρ	Simulated	Predicted ρ(1+ρ)
0.4	0.54	0.56
0.5	0.75	0.75
0.6	0.99	0.96
0.7	1.23	1.19
0.8	1.44	1.44
0.9	1.72	1.71

Figure 12. Variance of Cell Loss
M/M/1/K



From these results, the coefficient of variation for the average run loss is:

$$\text{c.o.v} = \frac{\rho(1+\rho)}{(1+\rho)^2}$$

$$= \frac{\rho}{1+\rho}$$

CHAPTER 4

M/D/1/K MODEL AND SIMULATION

4.1 Conditional Cell Loss Probability for M/D/1/K

Although not difficult to analyse, the M/M/1/K queue is not a good model of the behaviour of an ATM multiplexer. Since all cells are 53 octets long, and the time a multiplexer takes to process a cell is dependent on the cell size and the output line speed which is constant, then the time taken to process a cell can be expected to be constant. Consequently, a better model of a multiplexer uses a constant service time. So the M/D/1/K queue is a better model. However, because service times are not memoryless this is much more difficult to analyse than the M/M/1/K queue, and a different approach is needed. By considering conditional probability results and making use of the randomness in the arrival process, an analytic model of buffer overflow in the M/D/1/K queue can be derived.

Consider any cell N which is about to arrive, and will be lost because the buffer is full. What is the probability that cell N+1 will be lost as well? A result from conditional probability [Haykin] is Baye's Rule:

$$Pr(A|B) = Pr(A \text{ and } B) / Pr(B)$$

Considering the buffer overflow problem, then:

$$Pr(\text{cell } N{+}1 \text{ is lost} \mid \text{cell } N \text{ is lost})$$

$$= Pr\,(\text{cell } N{+}1 \text{ is lost and cell } N \text{ is lost}) / Pr(\text{cell } N \text{ is lost}) \qquad (9)$$

We now attempt to derive expressions in terms of service and arrival rates for the two probabilities on the right hand side of equation (9).

$$Pr(\text{cell } N \text{ is lost}) = Pr\,(1 \text{ arrival before } 1 \text{ service and queue is full}). \qquad (10)$$

and:

$$Pr(\textit{cell } N+1 \textit{ is lost and cell } N \textit{ is lost})$$

$$= Pr(\textit{2 arrivals before 1 service and queue is full}). \tag{11}$$

Substituting (10) and (11) into (9) gives:

$$Pr(\textit{cell } N+1 \textit{ is lost} | \textit{cell } N \textit{ is lost})$$

$$= \frac{Pr(\textit{2 arrivals before 1 service and queue is full})}{Pr(\textit{1 arrival before 1 service and queue is full})} \tag{12}$$

It is necessary to assume that the events (queue is full) and (n arrivals before one service) are independent. This seems reasonable since the arrival process is independent of the queue contents.

For independent events A, B :

$$Pr\ (A \textit{ and } B) = Pr\ (A)\ .\ Pr\ (B)$$

So equation (12) becomes:

$$Pr(\textit{cell } N+1 \textit{ is lost} | \textit{cell } N \textit{ is lost})$$

$$= \frac{Pr(\textit{2 arrivals before 1 service}).\ Pr(\textit{queue is full})}{Pr(\textit{1 arrival before 1 service}).\ Pr(\textit{queue is full})}$$

$$= \frac{Pr(\textit{2 arrivals before 1 service}).}{Pr(\textit{1 arrival before 1 service}).} \tag{13}$$

Equation 13 can be generalised to k arrivals before one service.

$$Pr(\textit{cell } N+1, N+2, \ldots, N+k \textit{ are lost} | \textit{cell } N \textit{ is lost})$$

$$= \frac{Pr(k \textit{ arrivals before 1 service and queue is full}).}{Pr(\textit{1 arrival before 1 service and queue is full}).}$$

$$= \frac{Pr(k\ arrivals\ before\ 1\ service).Pr(queue\ is\ full)}{Pr(1\ arrival\ before\ 1\ service).Pr(queue\ is\ full)}$$

So, in general:

$$Pr(burst\ of\ length\ k) = \frac{Pr(k\ arrivals\ before\ 1\ service)}{Pr(1\ arrival\ before\ 1\ service)} \tag{14}$$

We now apply this result to the M/D/1/K queue.

Since arrivals occur at random and service times are deterministic, we assume the amount of time any arrival sees until the next service has a uniform distribution, with maximum value the service time. That is, the probability density function of the service time is the service rate for the interval (0, service time) and zero elsewhere. This assumption appears reasonable because of the randomness in the arrival process.

Thus the probability density function for one service in time y is:

$$s(y) = \mu, \qquad \text{for } y \in (0, \frac{1}{\mu}) \quad \text{and zero elsewhere.}$$

For the M/D/1/K queue, arrivals are exponentially distributed. The probability density function of k arrivals in time x is [Kleinrock]:

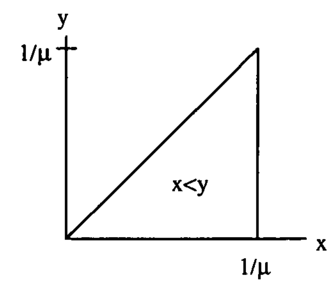
$$a(x) = \frac{\lambda(\lambda x)^{k-1}}{(k-1)!} e^{-\lambda x} \tag{15}$$

Using the assumption of independence, the joint probability density function is the product of the above two functions.

$$p_{xy} = \frac{\mu.\lambda(\lambda x)^{k-1} e^{-\lambda x}}{(k-1)!} \qquad \text{for } y \in (0, \frac{1}{\mu}), \text{ and 0 elsewhere} \tag{16}$$

Since s(y) is zero for y not in the interval $(0,1/\mu)$ then to determine the probability of k arrivals before one service we integrate over the finite triangular area shown in the following figure.

Figure 13. Bounds of Integration for M/D/1/K



That is:

$$Pr\ (k\ arrivals\ before\ 1\ service) = \int_0^{1/\mu}\int_0^{y} \mu.\lambda(\lambda x)^{k-1} e^{-\lambda x}\,dxdy \qquad (17)$$

This integral can be evaluated with the help of Integral tables [Gradshteyn] and the mathematics package 'Mathematica'. [Wolfram].

$$Pr\ (k\ arrivals\ before\ 1\ service) = \quad 1-\frac{1}{\rho}\sum_{i=0}^{k-1}\frac{\Gamma(i+1,0,\rho)}{i!} \qquad (18)$$

Where $\Gamma(a,\ z_0, z_1)$ is the generalised incomplete gamma function defined by:

$$\Gamma(a, z_0, z_1) = \int_{z_0}^{z_1} t^{a-1}e^{-t}dt$$

Consequently, when a burst of cells is lost, the probability that the burst will be of length k is given by:

$$Pr(burst\ is\ k\ or\ more\ cells\ long) = \frac{1-\frac{1}{\rho}\sum_{i=0}^{k-1}\frac{\Gamma(i+1,0,\rho)}{i!}}{1-\frac{1}{\rho}\Gamma(1,0,\rho)} \qquad (19)$$

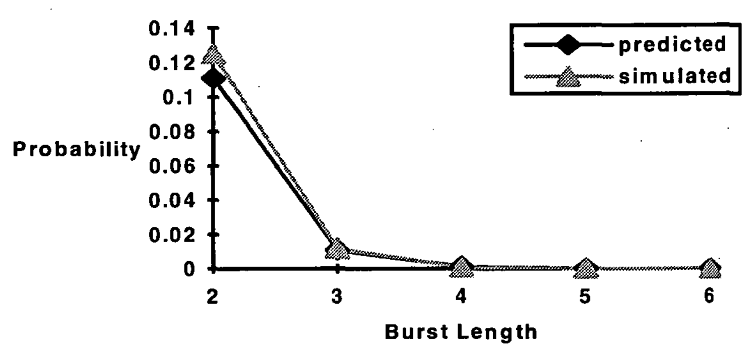
The following table uses the above formula to show expected probabilities of burst lengths for different values of queue utilisation, along with simulation data. Note that the simulation and calculated rates matches well.

Table 5. Probability of Cell Loss Greater Than or Equal to N
M/D/1/K
Simulated/predicted

	2		3		4		5	
0.4	0.111	0.125	0.0109	0.0119	0.0012	0.0009	0.0001	0.0001
0.5	0.137	0.153	0.0151	0.0182	0.0014	0.0016	0.0001	0.0002
0.6	0.166	0.181	0.0225	0.0256	0.0026	0.0030	0.0003	0.0003
0.7	0.195	0.207	0.0303	0.0337	0.0040	0.0046	0.0005	0.0005
0.8	0.224	0.233	0.0400	0.0429	0.0060	0.0065	0.0007	0.0008
0.9	0.253	0.258	0.0514	0.0528	0.0087	0.0089	0.0012	0.0017

The following figure is a graph of simulated and predicted values for server utilisation of 0.4. It shows that the predicted and simulated cell loss rates match adequately, although not as well as at higher utilisation. For lower utilisation, overflow occurs less frequently, so much longer simulation runs are needed. This may be sufficient to account for the small difference between the prediction and the simulation.

Figure 14. Graph of Predicted and Simulated Cell Loss
M/D/1/K
ρ=0.4

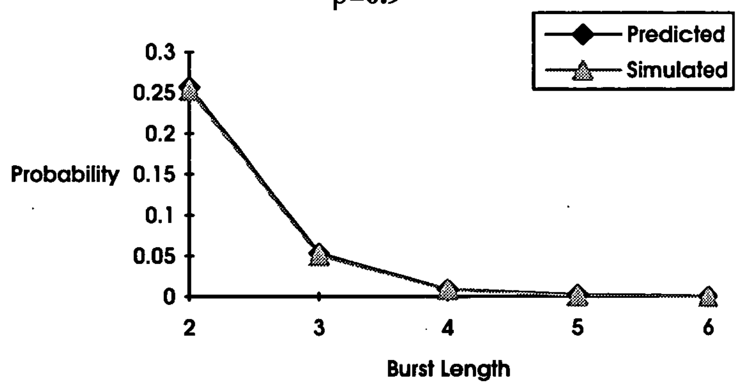


The following figure is a graph of simulated and predicted values for server utilisation of 0.9. It shows that the predicted and simulated cell loss rates match well.

Figure 15. Graph of Predicted and Simulated Cell Loss

M/D/1/K

ρ=0.9



4.2 Average Cell Loss Run for M/D/1/K

As was done for the M/M/1/K model of a multiplexer, we now derive analytic expressions for the mean and variance of the number of cells expected to be lost in a burst.

We now derive the probability that the burst is exactly length k.

Pr (number of cells lost due to buffer overflow is exactly k)

$$= Pr\,(length \geq k) - Pr\,(length \geq k+1)$$

$$= \frac{1-\frac{1}{\rho}\sum_{i=0}^{k-1}\frac{\Gamma(i+1,0,\rho)}{i!}-(1-\frac{1}{\rho}\sum_{i=0}^{k}\frac{\Gamma(i+1,0,\rho)}{i!})}{1-\frac{1}{\rho}\Gamma(1,0,\rho)}$$

$$= \frac{\frac{1}{\rho}\frac{\Gamma(k+1,0,\rho)}{k!}}{1-\frac{1}{\rho}\Gamma(1,0,\rho)} \tag{20}$$

The average number of cells lost in a run is:

$$\sum_{k=1}^{\infty} k \ \Pr(k \text{ cells lost} | \ 1 \text{ cell lost})$$

Consequently, for the M/D/1/K queue the average number of cells lost is:

$$\sum_{k=1}^{\infty} k \frac{1}{\rho} \frac{\Gamma(k+1,0,\rho)}{k!} \frac{1}{(1-\frac{1}{\rho}\Gamma(1,0,\rho))}$$

This is shown in appendix C to be:

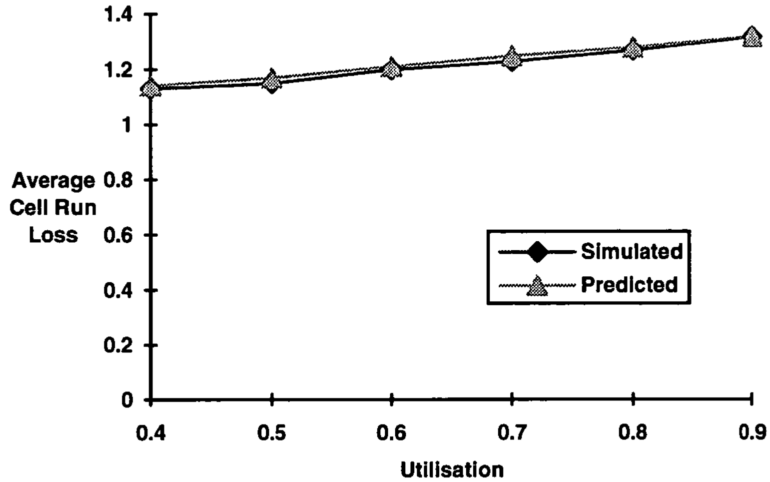
$$\frac{\rho^2}{2(\rho + e^{-\rho} - 1)} \tag{21}$$

The following table shows the simulated and predicted values for the average run loss length. There is good agreement between the simulated and predicted values. An interesting observation is that the average cell loss increases only slightly as the server utilisation increases.

Table 6. Average Cell Loss
Simulated and Predicted
M/D/1/K

ρ	Simulated	Predicted
0.4	1.13	1.14
0.5	1.15	1.17
0.6	1.20	1.21
0.7	1.23	1.25
0.8	1.27	1.28
0.9	1.32	1.32

Figure 16. Average Cell Loss
M/D/1/K



4.3 Variance for M/D/1/K

The variance of the number of cells lost in a run can be calculated as follows.

$$\sigma^2 = \sum_{k=1}^{\infty} p_k (x_i - \bar{x})^2$$

$$= \frac{1}{\rho + e^{-\rho} - 1} \sum_{k=1}^{\infty} \frac{\Gamma(k+1,0,\rho)}{k!} \left(k - \frac{\rho^2}{2(\rho + e^{-\rho} - 1)} \right)^2$$

In the appendix this is shown to be:

$$\frac{e^{\rho}\rho^2(6 - 6e^{\rho} + 4\rho + 2e^{\rho}\rho + e^{\rho}\rho^2)}{12(1 - e^{\rho} + e^{\rho}\rho)^2} \qquad (22)$$

From the above results we can calculate a coefficient of variation for the cell loss.

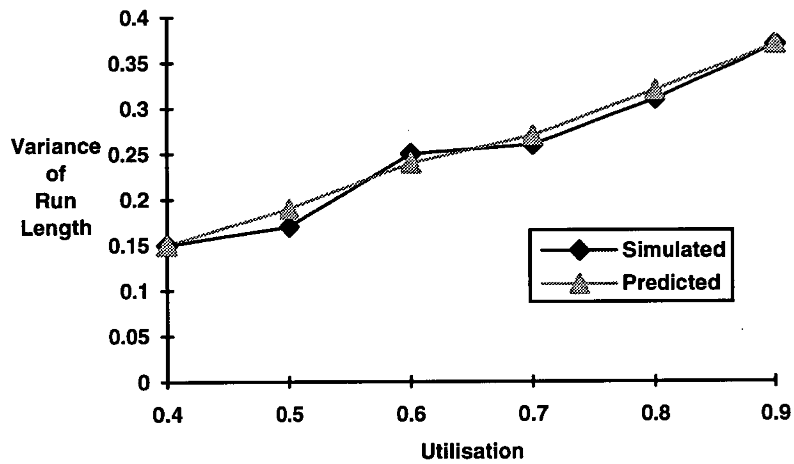
$$\text{c.o.v} = \frac{e^{\rho}(6 - 6e^{\rho} + 4\rho + 2e^{\rho} + e^{r}\rho^{2)}}{6(1 - e^{\rho} + e^{\rho}\rho)}$$

The following table and graph show the variance derived from equation 17 and the simulation. There is good agreement between the simulation and the equation.

Table 7. Variance of Cell Loss
Simulated and Predicted
M/D/1/K

ρ	Simulated	Predicted
0.4	0.15	0.15
0.5	0.17	0.19
0.6	0.25	0.24
0.7	0.26	0.27
0.8	0.31	0.32
0.9	0.37	0.37

Figure 17. Variance of Cell Loss
M/D/1/K



4.4 Another Look at M/M/1/K.

The method used above for M/D/1/K is a general one that can be applied to other queue disciplines. By using conditional probabilities the same result for the M/M/1/K queue can be derived as was derived in 3.1.

The probability of k arrivals before one service in the M/M/1 queue is $(\frac{\rho}{1+\rho})^k$

(23)

From the previous section:

$$Pr(\text{burst of length } k) = \frac{Pr(k \text{ arrivals before } 1 \text{ service})}{Pr(1 \text{ arrival before } 1 \text{ service})}$$

Then, for the M/M/1 case:

$$Pr(\text{burst of length } k) = (\frac{\rho}{1+\rho})^k / (\frac{\rho}{1+\rho})$$

$$= (\frac{\rho}{1+\rho})^{k-1} \qquad (24)$$

which is the same result as in 3.1.

CHAPTER 5

M/D/1/K WITH MINIMUM INTERARRIVAL TIMES

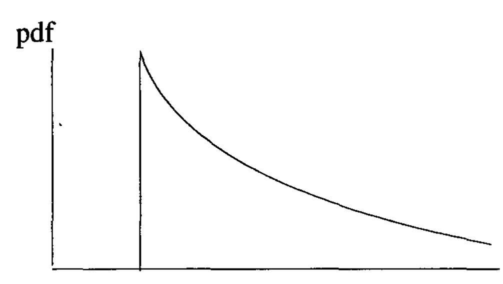
A further refinement of our model is to consider random arrivals subject to minimum interarrival times. Only approximate mathematical models of this traffic have been developed. However they give some indication of the behaviour of this traffic and compare reasonably well with simulation results.

We consider the simplest minimum interarrival time case where there is a minimum time δ which must elapse before the next arrival. The arrival probability function for this is not easy to describe, since the minimum time for the first arrival since the last is delta, but for the second arrival it is two delta and so on. This is a difficult arrival pattern to describe mathematically. However, we can approximate it by using a displaced exponential function of the form:

$$p_x = \lambda^k (x - k\delta)^{k-1} e^{-\lambda(x-k\delta)} \quad \text{, for } x > k\delta \text{ and 0 elsewhere.} \qquad (25)$$

This probability density function is a modified exponential distribution. It describes an arrival process in which there is a delay of k multiplied by the interarrival period and then k arrivals with a Poisson distribution. There is no correlation between the minimum interarrival time and the server time. The following illustration shows the arrival distribution.

Figure 18. Poisson Arrivals with Minimum Interarrival Times



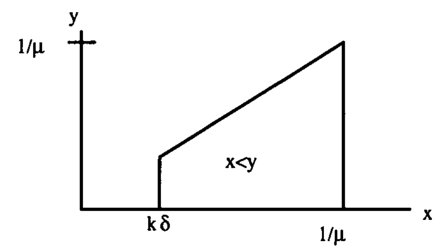
Since the arrivals are random, the time until the next service from the time of each arrival will be uniformly distributed as before, between 0 and $1/\mu$.

The bounds of integration are defined by the modified Poisson distribution. We will integrate on the bounded area defined by:

$$0 \le y \le \frac{1}{\mu} \text{ and } k\delta \le x \le y \tag{26}$$

This is shown in the following figure.

Figure 19. Bounds of Integration for M/D/1 with Minimum Interarrival Times



Thus, an expression for the probability that there will be k arrivals before one departure is:

$$\int_0^{1/\mu}\int_{k\delta}^{y} \mu\lambda^k (x-k\delta)^{k-1} e^{-\lambda(x-k\delta)} dxdy \tag{27}$$

This integral cannot be evaluated analytically, but must be determined numerically for specific parameter values. Using the same approach as before, we are able to determine expected burst lengths for different values of server utilisation. The following table shows the expected values and simulation values for $\rho=0.9$, $\mu=100$, $\lambda=90$ and $\delta = 0.001$ seconds (one tenth of service time). Although the arrival distribution does not describe the arrivals exactly, it is a reasonable approximation.

Table 8. M/D/1 with minimum interarrival times

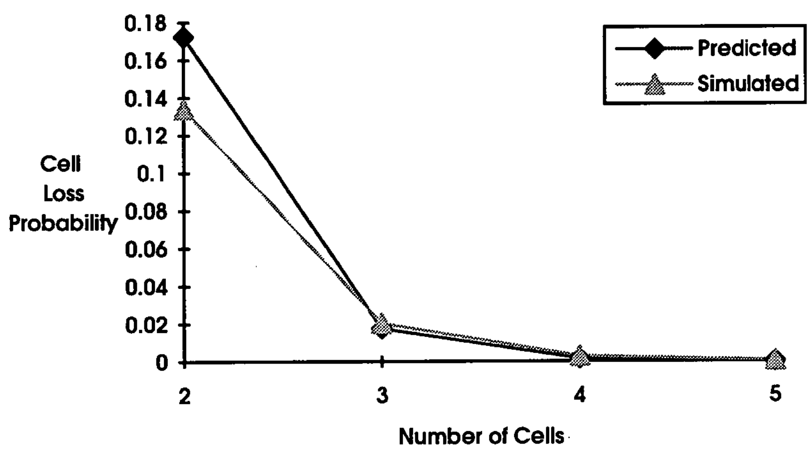
δ=0.1

Predicted / simulated

ρ	2		3		4		5	
0.9	0.173	0.134	0.0178	0.0204	0.0016	0.0030	0.00006	0.00040

The following figure is a graph of the expected and simulated data for ρ=0.9. The graph shows that the approximation to minimum interarrival times is acceptable.

Figure 20. Expected / Simulated Interarrival Times



CHAPTER 6

SWITCHED POISSON PROCESS

The *Switched Poisson Process* has been suggested as a good model for bursty traffic such as data and video, where the channel switches between periods of high and low activity. [Heffes and Lucantoni], [Rossiter].

The *Switched Poisson Process* is characterised by four parameters λ_1, λ_2, γ, ω. It is based on an underlying *Alternating Poisson Process* (APP). The process spends exponentially distributed time with mean of $1/\gamma$ seconds in an 'on' mode where arrivals occur with an interarrival time described by a Poisson distribution with mean $1/\lambda_1$; and $1/\omega$ omega in an 'off' mode where arrivals occur according to a Poisson distribution with mean $1/\lambda_2$ where $\lambda_1 > \lambda_2 \geq 0$. When $\lambda_2 = 0$ the SPP becomes an *Interrupted Poisson Process* (IPP).

Conditional cell loss probabilities can be approximated by linearly interpolating between the probabilities in the 'on' and 'off' modes. The APP will be in the 'on' mode with probability $\omega/(\omega+\gamma)$ and in the off mode with probability $\gamma/(\gamma+\omega)$. The rate of conditional cell loss lengths can be expected to be dependent on λ_1 when in the on mode and λ_2 when in the off mode. This is only an approximation as any 'hangover' effects between states is ignored.

If the cell loss rate is dependent only on the state of the system, and ignoring hangover effects, the probability of cell loss is:

$$Pr(k\ cells\ loss \mid 1\ cell\ lost) = [\gamma B(\lambda_1) + \omega B(\lambda_2)] \,/\, (\omega+\gamma)$$

where $B(\lambda)$ is the conditional probability of cell loss when the arrival rate is λ.

The simulation studies suggest that the switching rate affects the cell loss rate. The following table and graph show the behaviour of the system for a SPP arrival process in which the average time spent in the 'on' and 'off' mode are equal, but the average time between switching varies. The predicted cell loss probability for two cells with γ and ω equal is 0.168.

The table and graph suggest that when γ and ω are small, the hangover effect is most pronounced. As γ and ω increase, the hangover effect becomes less pronounced.

Table 9. Switched Poisson Process
Equal Switching Times
$\lambda_1 = 0.8, \lambda_2 = 0.4$

γ	ω	2	3
1000	1000	0.162	0.020
100	100	0.1536	0.020
10	10	0.169	0.023
1	1	0.182	0.027
0.1	0.1	0.210	0.038
0.01	0.01	0.215	0.039
0.0001	0.0001	0.218	0.038

Figure 21. Switched Poisson Process
Equal Switching Times
$\lambda_1 = 0.8, \lambda_2 = 0.4$

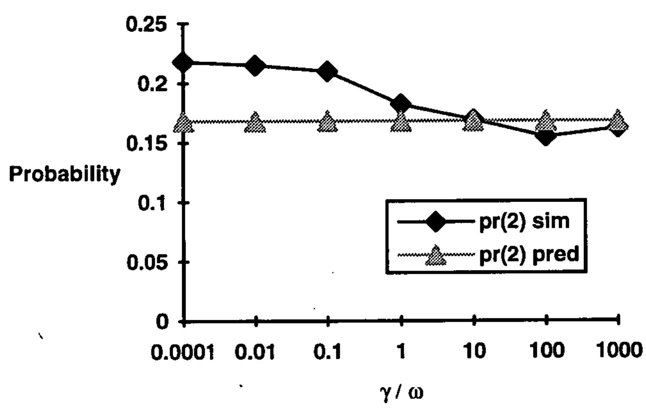


Table 10 shows predicted and simulated cell loss conditional probabilities for some *Switched Poisson Processes* with γ and ω comparable to the arrival rate. There is reasonable agreement between predicted and simulated values, although there is some evidence of a hangover effect. To plot this data we need to normalise the base so that $\gamma + \omega = 1$. We can then see the behaviour of the model as γ goes from zero to one and ω goes from one to zero. Table 10 and the following graph show the channel utilisation against the normalised gamma. The hangover effect is more apparent on the normalised graph. Since the values of γ and ω used are comparable to the arrival rate, we would expect some evidence of a hangover effect.

The normalised graph shows some bunching at higher server utilisations. Clearly, the SPP arrival process is quite complex, and linear interpolation between the two extreme points is only an approximation.

Table 10. Switched Poisson Process
$\lambda_1 = 0.8, \lambda_2 = 0.4$
Predicted and Simulated

γ	ω	2		3		4	
5.000	2.000	0.143	0.156	0.0123	0.0199	0.0026	0.0022
2.000	1.000	0.149	0.177	0.0216	0.0215	0.0028	0.0033
1.000	0.500	0.175	0.177	0.0250	0.0215	0.0028	0.0024
1.000	1.000	0.168	0.182	0.0255	0.0271	0.0036	0.0035
0.100	1.000	0.213	0.224	0.0460	0.0411	0.0060	0.0062
0.010	1.000	0.223	0.224	0.0400	0.0410	0.0060	0.0060
0.500	1.000	0.187	0.197	0.0206	0.0318	0.0044	0.0044
0.100	0.010	0.214	0.199	0.0374	0.0340	0.0056	0.0050
4.000	1.000	0.134	0.164	0.0340	0.0230	0.0050	0.0026
1.000	0.010	0.112	0.113	0.0112	0.0090	0.0012	0.0018

Table 11. Normalised γ for SPP/D/1/K

$\lambda_1 = 0.8, \lambda_2 = 0.4$

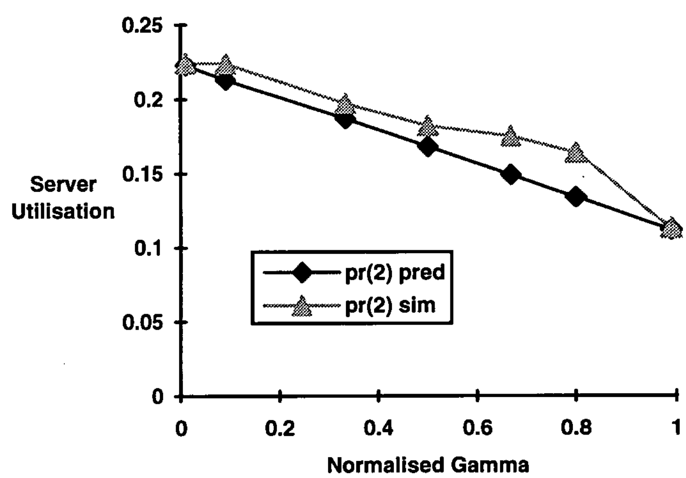
Predicted and Simulated

γ	ω	$\gamma/(\gamma+\omega)$	pr(2) pred	pr(2) sim
0.010	1.000	0.010	0.223	0.224
0.100	1.000	0.091	0.213	0.224
0.500	1.000	0.333	0.156	0.197
1.000	1.000	0.500	0.168	0.182
1.000	0.500	0.667	0.149	0.171
4.000	1.000	0.800	0.134	0.164
1.000	0.010	0.990	0.112	0.113

Figure 22. Normalised γ for SPP/D/1/K

$\lambda_1 = 0.8, \lambda_2 = 0.4$

Predicted and Simulated



CHAPTER 7

DISCUSSION OF RESULTS AND CONCLUSION

7.1 Characterisation of Cell Loss.

The most important qualitative result in this paper is that cell loss due to buffer overflow is more likely to occur in runs than at random. Consequently, stating cell loss as a simple rate is insufficient to characterise the nature of cell loss. In chapters 3 through 6 we derived probability distribution functions that show cell run loss probabilities, and for the M/M/1/K and M/D/1/K cases, average and variance of the expected number of cells lost. Because of the exponential nature of the M/M/1 case, the conditional probability of losing one cell given that the previous one has been lost is a suitable statistic for characterising cell loss. However for the other models of cell loss, this is not a suitable statistic as it is not constant for each possible number of cells lost in the run.

Although it is difficult to derive analytic expressions for them, the average length of a cell loss run and its variance are preferred statistics for characterising cell loss. These have been derived for the M/M/1/K and the M/D/1/K queues. In principle, these statistics can be derived for the approximations to the other two multiplexer models, although they are computationally difficult.

For the M/D/1/K queue, the average cell loss run increases only by a small amount as the utilisation increases. For utilisation of 0.4 the average cell loss is 1.14, while for a utilisation of 0.9 the average cell loss is 1.32.

7.2 Repercussions for ATM Traffic.

7.2.1 Statistical Multiplexing

The analysis in chapters 3 through 6 has described cell loss from the point of view of the network. We have modelled the traffic as coming into a multiplexer from several sources to be processed by one deterministic server. We have seen that the conditional probability of cell loss is substantially higher than the overall cell loss rate. However, from the point of view of an individual input into the multiplexer, the situation is quite different. Because several inputs may be transmitting concurrently, if a cell is lost by one input, the next cell is not necessarily going to be lost by the same input. So the conditional probability of cell loss experienced by an individual input is likely to be less than those modelled in chapters 3 through 6.

However, if statistical multiplexing is being used, much of the capacity of the channel allocated to variable bit rate traffic can be expected to be used by one input at one time, depending on the burstiness of the service. As we have already noted, B-ISDN services tend to be bursty. So we would expect the results of chapters 3 through 6 to be limits that will be approached as the burstiness of the service increases.

Even if the cell loss is at its maximum, will it have a significant effect on ATM traffic? For services such as voice which can cope with high loss rates, it would appear to not be significant. However, for video and data where loss rates need to be low, a string of cell losses, rather than isolated ones might be serious.

Statistical multiplexing needs to be introduced cautiously. A conservative traffic policy is needed. Once cell loss occurs, retransmissions through congested nodes may cause the overall network performance to collapse. Depending on the service, different strategies to deal with congestion are needed.

For circuit oriented services such as voice and video, it may be necessary to increase blocking probabilities to prevent additional traffic exacerbating the congestion.

7.2.2 Data Traffic

Some data transmission schemes use Forward Error Correction schemes, similar to Hamming Codes to cope with corrupted frames. If the frame cannot be corrected or interpolated, a retransmit is required. Such FEC schemes need to take into account the high probability that a frame will lose more than one consecutive cell, resulting in more retransmits than might be expected. [Ohta and Kitami] describe such a scheme. Generally, Hamming schemes introduce a great deal of additional redundancy into the message. Since optic fibre is such a reliable means of transmission, in most cases FEC schemes are not necessary, as it is more efficient to retransmit the message.

Retransmits due to congestion are to be avoided (as opposed to those due to congestion), since they may add to the traffic through already congested nodes, leading to congestion collapse.

It is worth noting that although the probability of losing one cell, given that one has already been lost is substantially higher than the overall cell loss rate, the probability of losing more than a few cells trails off quickly. For the M/D/1/K queue with a utilisation of 0.8, the probability of losing eight or more cells in a row is of the order of 10^{-6}. Consequently, using the results of this paper, FEC schemes can be designed to a given tolerance, depending on the needs of the service.

7.2.3 Compressed Video

Compressed video using MPEG protocol sends an initial image and then updates to the image followed by a refresh every 15 frames. Cell loss during the updates is not critical because Discrete Cosine Transform coding and extrapolation from previous frames can be used to disguise the loss. However, if many cells are lost or if a frame is corrupted during the initial image transmission, image quality suffers. There is no provision for retransmits in compressed video. If the initial image is protected by Forward Error Correction, then the same comments apply to it as does to data transmission. The FEC should be able to cope with less frequent bursts of errors rather than uniformly distributed errors.

Generally, FEC schemes are not used with compressed video as they introduce a great deal of transmission overhead.

7.3 Summary

This report has described mathematical models of the conditional cell loss probabilities of ATM multiplexer traffic and simulations that have tested the models. The models have been seen to match the simulations well. From the models and simulations it is apparent that cell loss due to buffer overflow in an ATM multiplexer persists. If one cell is lost, the probability of successive cells being lost is much higher than the overall cell loss rate.

The important result is that network designers need to be conservative in network design. This paper provides a mathematical understanding of observations from simulation studies that congestion persists in networks and the consequences of misengineering can be serious and prolonged. [Leland and Fowler], [Ramaswami].

Cell loss probability due to buffer overflow in a multiplexer needs to be described by more than the overall rate. Some possible statistics that can be used are the average cell loss run and its variance. For the M/M/1/K model, the conditional cell loss probability rate alone is a suitable statistic. Future research could be done into expressing cell loss runs in terms of confidence limits. For example, what is the upper limit of the cell loss run 95% of the time?.

The analysis of the conditional probabilities of cell loss in the M/M/1/K and M/D/1/K queues has included derivation of analytic expressions for the calculation of conditional probabilites, and the average and variance of cell loss run length. As well as being multiplexer models they are quite interesting in their own right.

The bursty nature of cell loss has implications for ATM traffic. Delay sensitive traffic such as voice may require higher blocking probabilities. Video traffic may incur more refreshes than otherwise expected. Data traffic might need forward error correction oriented towards bursts of cell loss followed by long periods of error free operation.

Switched Poisson Process arrivals are quite complicated, as the cell loss is dependent on both the average time in each state and the switching rate.

Future research on cell loss could involve the application of these results to any of the above areas, to extending the analysis to more complex arrival and service patterns, and to modelling cell loss from the perspective of an individual input into a multiplexer.

BIBLIOGRAPHY AND REFERENCES

1. Apostol T., *Calculus. Vol 1. (2nd edition),* Blaisdell 1967

2. Bae J., Suda T., *Survey of Traffic Control Schemes and Protocols in ATM Networks,* Proceedings of the IEEE, Vol 79, No. 2, February 1991, pp 170-189.

3. Beyer W., (Ed), *CRC Handbook of Mathematical Sciences,* CRC Press, 1978.

4. Corun, Szarba., *Multivariate Calculus.,* Mathematical Monographs 1982.

5. dePrycker M., *Asynchronous Transfer Mode: solution for broadband ISDN.,* Ellis Horwood 1991.

6. Eckberg A., *B-ISDN/ATM Traffic and Congestion Control,* September 1992 IEEE Network Magazine, pp 28-37.

7. Erfani S., Malek M., Sacher H., *An Expert System-based Approach to Capactiy Allocation in a Multiservice Application Environment.,* IEEE Communications Magazine, May 1991.

8. Fowler H., Leland W., *Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management.,* IEEE Journal on Selected Areas in Communications., Vol 9. No. 7 September 1991, pp 1139-1149.

9. Gallassi G., Rigolio G., Verri L., *Resource Management and Dimensioning in ATM Networks.* IEEE Network Magazine, May 1990, pp 8-17.

10. Gilbert H., Aboul-Magd O., Phung V., *Developing a Cohesive Traffic Management Strategy for ATM networks.,* IEEE Communications magazine., Oct 1991, pp 36-45.

11. Gonzalez R., Woods R., *Digital Image Processing (3rd edition),* AddisonWesley 1992.

12. Goldstein F., *ISDN in perspective.,* Addison Wesley 1992.

13. Gradshteyn I., Ryzhik I., *Tables of Integrals, Series and Products.,* Academic Press 1976.

14. Gross D. Harris C., *Fundamentals of Queuing Theory.,* Wiley 1974.

15. Handel R., Huber M., *Integrated Broadband Networks: An Introduction to ATM-Based Networks*, Addison-Wesley 1992.

16. Haykin S., *Digital Communications.,* John Wiley and Sons 1988.

17. Heffes H., Lucantoni D., *A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance.*, IEEE Journal on Selected Areas in Communications, Vol 4, No 6 September 1986.

18. Hiramatsu A., *Integration of ATM Call Admission Control and Link Capacity Control by Distributed Neural Networks.*, IEEE Journal on Selected Areas in Communications, Vol 9, No 7, September 1991 pp 1131-1138.

19. Hong D., Suda T., *Congestion Control and Prevention in ATM Networks.*, IEEE Network Magazine., July 1991, pp 10-16.

20. Jain R., *Congestion Control in Computer Networks: Issues and Trends,* May 1990, IEEE Network Magazine, pp 24-30.

21. Kleinrock L., *Queuing Systems. Vol 1.* John Wiley and Sons, 1975.

22. Nagle J., *On Packet Switches with Infinite Storage.,* IEEE Transactions on Communication, Vol 35, No. 4, April 1987, pp 435-438.

23. Ohta H., Kitami T., *A Cell Loss Recovery Method using Forward Error Correction in ATM Networks.*, IEEE Journal on Selected Areas in Communication, Vol 9, No. 9, December 1991, pp 1471-1483.

24. Ramaswami V., Willinger, W., *Efficient Traffic Performance Strategies for Packet Multiplexers.,* Proceedings of ITC Specialist Seminar, Adelaide, 1989. Paper No. 4.2.

25. Rossiter M., *A Switched Poisson Model for Data Traffic.,* A.T.R ,Vol 21, No 1, 1987, pp 53-57.

26. Russel E., *Building Simulation Models with SIMSCRIPT II.5.,* CACI 1983.

27. Russel E., *SIMSCRIPT II.5 Programming Language (4th edition).,* CACI 1987.

28. Stallings W., *Data and Computer Communications (2nd edition).* Macmillan 1988.

29. Stallings, W., *ISDN and Broadband ISDN (2nd edition).,* Macmillan 1992.

30. Telecom Australia, *A Course in Teletraffic Engineering.,* 1978.

31. Wernik M., Aboul-Magd O. Gilbert H., *Traffic Management for B-ISDN Services,* September 1992 IEEE Network Magazine, pp 10-19.

32. Woodruff G., Kositpaiboon R., *Multimedia Traffic Management Principles for Guaranteed ATM Network Performance.,* IEEE Journal on Selected Areas in Communications. Vol 8. No. 3 April 1990, pp 437-446.

33. Wolfram S., *Mathematica: A System for Doing Mathematics by Computer (2nd edition).,* Addison-Wesley, 1991.

APPENDIX A

Average Cell Loss Length M/M/1/K

Theorem:

$$\sum_{k=1}^{\infty}\frac{k\rho^{k-1}}{(1+k)^k}=1+\rho$$

Proof:

$$\begin{aligned}
\text{LHS} &= \sum_{k=1}^{\infty}\frac{k\rho^{k-1}}{(1+k)^k} \\
&= \frac{1}{\rho}\sum_{k=1}^{\infty}k\left(\frac{\rho}{1+\rho}\right)^k \\
&= \frac{1}{\rho}.\frac{\rho}{1+\rho}\sum_{k=1}^{\infty}k\left(\frac{\rho}{1+\rho}\right)^{k-1} \\
&= \left(\frac{1}{1+\rho}\right)\frac{1}{\left(1-\left(\frac{\rho}{1+\rho}\right)\right)^2} \\
&= \left(\frac{1}{1+\rho}\right)(1+\rho)^2 \\
&= 1+\rho
\end{aligned}$$

APPENDIX B

Variance of Cell Loss M/M/1/K

Theorem:

$$\sum_{i=1}^{\infty}\frac{\rho^{i-1}}{(1+\rho)^i}(i-(1+\rho))^2=\rho(1+\rho)$$

Proof:

$$\text{LHS} = \sum_{i=1}^{\infty}\frac{\rho^{i-1}}{(1+\rho)^i}(i-(1+\rho))^2$$

$$= \sum_{i=1}^{\infty}\frac{i^2\rho^{i-1}}{(1+\rho)^i}-2(1+\rho)\sum_{i=1}^{\infty}\frac{i\rho^{i-1}}{(1+\rho)^i}+(1+\rho)^2\sum_{i=1}^{\infty}\frac{\rho^{i-1}}{(1+\rho)^i}$$

Let $x=\frac{\rho}{1+\rho}$, then:

$$\text{LHS} = \left(\frac{1}{1+\rho}\right)\sum_{i=1}^{\infty}i^2x^{i-1}-2\sum_{i=1}^{\infty}ix^{i-1}+(1+\rho)^2\sum_{i=1}^{\infty}x^{i-1}$$

Now, we know the following [Apostol]:-

$$\sum_{i=1}^{\infty}x^{i-1}=\frac{1}{1-x}=1+\rho \text{ and,}$$

$$\sum_{i=1}^{\infty}ix^{i-1}=\left(\frac{1}{1-x}\right)^2=(1+\rho)^2 \text{ and,}$$

$$\sum_{i=1}^{\infty}i^2x^{i-1}=\frac{1+x}{(1-x)^3}=(1+\rho)^2(1+2\rho)$$

So,

$$\text{LHS} = \left(\frac{1}{1+\rho}\right)\left((1+\rho)^2(1+2\rho) - 2(1+\rho)^2 + (1+\rho)^3\right)$$

$$= \rho(1+\rho)$$

APPENDIX C

Average Cell Loss M/D/1/K

Theorem:

$$\sum_{k=1}^{\infty} k \frac{1}{\rho} \frac{\Gamma(k+1,0,\rho)}{k!} \frac{1}{1-\frac{1}{\rho}\Gamma(1,0,\rho)} = \frac{\rho^2}{2(1+e^{-\rho}-1)}$$

Where $\Gamma(a, z_0, z_1)$ is the generalised incomplete gamma function defined by:

$$\Gamma(a, z_0, z_1) = \int_{z_0}^{z_1} t^{a-1} e^{-t} dt$$

Proof:

$$\begin{aligned}
\text{LHS} &= \sum_{k=1}^{\infty} k \frac{1}{\rho} \frac{\Gamma(k+1,0,\rho)}{k!} \frac{1}{1-\frac{1}{\rho}\Gamma(1,0,\rho)} \\
&= \frac{1}{\rho(1-\frac{1}{\rho}\Gamma(1,0,\rho)} \sum_{k=1}^{\infty} \frac{k\Gamma(k+1,0,\rho)}{k!} \\
&= \frac{1}{\rho(1-\frac{1}{\rho}\Gamma(1,0,\rho)} \sum_{k=1}^{\infty} \frac{\Gamma(k+1,0,\rho)}{(k-1)!} \\
&= \frac{1}{\rho(1-\frac{1}{\rho}\Gamma(1,0,\rho)} \sum_{k=1}^{\infty} \int_0^{\rho} \frac{e^{-t} t^k}{(k-1)!} dt \\
&= \frac{1}{\rho(1-\frac{1}{\rho}\Gamma(1,0,\rho))} \int_0^{\rho} e^{-t} t \sum_{k=1}^{\infty} \frac{t^{k-1}}{(k-1)!} dt
\end{aligned}$$

Now, remembering that $e^t = \sum_{k=1}^{\infty} \frac{t^{k-1}}{(k-1)!}$ then,

$$\text{LHS} = \frac{1}{\rho(1 - \frac{1}{\rho}\Gamma(1,0,\rho))} \int_0^{\rho} e^{-t} t e^{t} dt$$

$$= \frac{\rho^2}{2(\rho + e^{-\rho} - 1)}$$

APPENDIX D

Variance of Cell Loss M/D/1/K

Theorem

$$\frac{1}{(\rho+e^{-\rho}-1)}\sum_{k=1}^{\infty}\frac{\Gamma(k+1,0,\rho)}{k!}\left(k-\frac{\rho^2}{2(\rho+e^{-\rho}-1)}\right)^2=\frac{e^{\rho}\rho^2(6-6e^{\rho}+4\rho+2e^{\rho}\rho+e^{\rho}\rho^2)}{12(1-e^{\rho}+e^{\rho}\rho)^2}$$

Proof:

$$\text{LHS} = \frac{1}{(\rho+e^{-\rho}-1)}\sum_{k=1}^{\infty}\frac{\Gamma(k+1,0,\rho)}{k!}\left(k-\frac{\rho^2}{2(\rho+e^{-\rho}-1)}\right)^2$$

Let $x=\dfrac{\rho^2}{2(\rho+e^{-\rho}-1)}$. Then

$$\begin{aligned}\text{LHS} &= \frac{1}{(\rho+e^{-\rho}-1)}\sum_{k=1}^{\infty}\frac{(k-x)^2}{k!}\int_0^{\rho}t^k e^{-t}dt \\ &= \frac{1}{(\rho+e^{-\rho}-1)}\int_0^{\rho}e^{-t}\sum_{k=1}^{\infty}\frac{t^k(k-x)^2}{k!}dt \\ &= \frac{1}{(\rho+e^{-\rho}-1)}\int_0^{\rho}e^{-t}\left(\sum_{k=1}^{\infty}\frac{t^k k^2}{k!}-2x\sum_{k=1}^{\infty}\frac{t^k}{(k-1)!}+x^2\sum_{k=1}^{\infty}\frac{t^k}{(k-1)!}\right)dt \\ &= \frac{1}{(\rho+e^{-\rho}-1)}\int_0^{\rho}e^{-t}\left(\left(\sum_{k=1}^{\infty}\frac{t^k k^2}{k!}\right)-2xe^t+tx^2e^t\right)dt\end{aligned}$$

Now $\quad e^t=\displaystyle\sum_{k=1}^{\infty}\frac{t^{k-1}}{(k-1)!}$

$$\therefore \qquad te^t = \sum_{k=1}^{\infty} \frac{t^k}{(k-1)!}$$

$$\therefore \qquad \frac{dte^t}{dt} = \sum_{k=1}^{\infty} \frac{kt^{k-1}}{(k-1)!}$$

$$\therefore \qquad e^t(t+1) = \sum_{k=1}^{\infty} \frac{k^2 t^{k-1}}{k!}$$

$$\therefore \qquad e^t t(t+1) = \sum_{k=1}^{\infty} \frac{k^2 t^k}{k!}$$

Thus,

$$\text{LHS} = \frac{1}{(\rho + e^{-\rho} - 1)} \int_0^{\rho} e^{-t}\left(e^t t(t+1) - 2xe^t + tx^2 e^t\right) dt$$

$$= \frac{1}{(\rho + e^{-\rho} - 1)} \left(\frac{\rho^3}{3} + \frac{\rho^2}{2} - \rho^2 x + \frac{\rho^2 x^2}{2} \right)$$

$$= \frac{e^{\rho}\rho^2(6 - 6e^{\rho} + 4\rho + 2e^{\rho}\rho + e^{\rho}\rho^2)}{12(1 - e^{\rho} + e^{\rho}\rho)^2}$$

APPENDIX E

M/M/1/K SIMULATION PROGRAM

```
Preamble
"    ISDN statistical multiplexer buffer simulation
"
"    Philip Branch, MTech project 1993.
"
"    Use SIMSCRIPT to model the behaviour of a multiplexer. In particular
"    determine whether cell (packet) loss occurs in bursts.
"
     normally, mode is undefined          "Force variable declaration
     Processes include generator, arrival     "Queue entries
     Resources include server             "Queue server

"Variable declarations
     Define head.date, head.time as text variable
     Define bufsize, duration as integer variable
     Define lambda, mu as real variable
     Define cell.count as integer variable
     Define cells.lost as integer, 1-dimensional array
     Define mean.arrival, mean.service as real variables
     Define total.entries as integer variable

" Time definitions
     Define .seconds to mean days
     Define .milliseconds to mean hours
     Define .microseconds to mean minutes

end    "preamble
Main

"Main routine
"===========

normally, mode is undefined

"Change time scale
     Let hours.v  = 1000
     let minutes.v = 1000

" Reserve space for statistic records
     reserve cells.lost(*) as 11
```

```
" Setup 1 server
     create every server(1)
     let U.server(1) = 1
" Set up input and output files
     open unit 2 for input, file name is "C:\SIM\PARAM.DAT"
     use 2 for input

     open unit 3 for output, file name is "C:\SIM\LOG.DAT"
     use 3 for output

" Get date and time for heading
     call date.r yielding head.date, head.time

"Print heading
     print 5 lines with head.date, head.time thus
     ISDN cell loss simulation    M/M/1   Date ********** Time ********

 Buffer          Dura
 Size  Lambda Mu  tion   1  2  3  4  5  6  7  8  9  10 >10


     read bufsize, lambda, mu, duration as I 6,  I 6, I 6, I 6

"      while eof.v = 0 do

        let mean.arrival = 1.0 / lambda
        let mean.service = 1.0 / mu
        let time.v = 0
        activate a generator now
        start simulation
        call printline

"DEBUG
        print 1 line with lambda, mean.arrival, mu, mean.service thus
lambda ***.*** mean.arrival ***.*** mu ***.*** mean.service ***.***
"END DEBUG


"         read bufsize, lambda, mu, duration as I 6, I 6, I 6, I 6
"      loop

stop
end
process arrival

normally, mode is undefined

request 1 server(1)
work exponential.f(mean.service, 2) .seconds

relinquish 1 server(1)

end
```

```
process generator

normally, mode is undefined

define counter as integer variable

"DEBUG
print 1 line with mean.arrival thus
interarrival time *******.********
"ENDDEBUG

let counter = 0
wait exponential.f (mean.arrival, 1) .seconds
"while time.v < duration do
while 2 > 1 do              " dummy loop. Continue forever.
   if counter > 10000
      let counter = 0
      call printline
   endif
   add 1 to counter

   add 1 to total.entries

   if N.Q.server > bufsize    " server queue full. This one will be lost
      add 1 to cell.count

"DEBUG
"     print 1 line with cell.count thus
"Cell lost.  cell.count = ****
"END DEBUG


   else

"     Record number of cells lost. Add 1 to each count for multiple cells.
      if cell.count > 0
         if cell.count le 10
            add 1 to cells.lost(cell.count)
         else
            add 1 to cells.lost(11)
         endif

"        There is room for an entry. Put in on the queue.
         let cell.count = 0
      endif
      activate an arrival now
   endif
   wait exponential.f (mean.arrival, 1) .seconds
loop
end
routine isdnsim

normally, mode is undefined
```

```
call resetall
let time.v = 0
activate a generator now
start simulation
call printline
return
end
routine printline

normally, mode is undefined

print 1 line with bufsize, lambda, mu, duration,
cells.lost(1),
cells.lost(2),
cells.lost(3),
cells.lost(4),
cells.lost(5),
cells.lost(6),
cells.lost(7),
cells.lost(8),
cells.lost(9),
cells.lost(10),
cells.lost(11) thus
 **** **** **** ****** **** **** **** **** **** **** **** **** **** ****
****

"Show total number of cells processed
print 2 lines with total.entries thus

Total cells processed = *********

return
end
```

APPENDIX F

M/D/1/K SIMULATION PROGRAM

```
Preamble
"    ISDN statistical multiplexor buffer simulation
"
"    Philip Branch, MTech project 1993.
"
"    Use SIMSCRIPT to model the behaviour of a multiplexor. In particular
"    determine whether cell (packet) loss occurs in bursts.
"
     normally, mode is undefined            "Force variable declaration
     Processes include generator, arrival   "Queue entries
     Resources include server               "Queue server

"Variable declarations
     Define head.date, head.time as text variable
     Define bufsize, duration as integer variable
     Define lambda, mu as real variable
     Define cell.count as integer variable
     Define cells.lost as integer, 1-dimensional array
     Define mean.arrival, mean.service as real variables
     Define total.entries as integer variable

" Time definitions
     Define .seconds to mean days
     Define .milliseconds to mean hours
     Define .microseconds to mean minutes


" Debug
     Define sv.time as real variable
" End debug

end    "preamble
Main

"Main routine
"===========

normally, mode is undefined

"Change time scale
     Let hours.v  = 1000
     let minutes.v = 1000

" Reserve space for statistic records
```

```
    reserve cells.lost(*) as 11

" Setup 1 server
    create every server(1)
    let U.server(1) = 1
" Set up input and output files
    open unit 2 for input, file name is "C:\SIM\PARAM.DAT"
    use 2 for input

    open unit 3 for output, file name is "C:\SIM\LOG.DAT"
    use 3 for output

" Get date and time for heading
    call date.r yielding head.date, head.time

"Print heading
    print 2 lines with head.date, head.time thus
    ISDN cell loss simulation    M/D/1    Date ********** Time ********

    read bufsize, lambda, mu, duration as I 6,  I 6, I 6, I 6


    let mean.arrival = 1.0 / lambda
    let mean.service = 1.0 / mu
    let time.v = 0
    activate a generator now
    start simulation
    call printline

"DEBUG
    print 1 line with lambda, mean.arrival, mu, mean.service thus
lambda ***.*** mean.arrival ***.*** mu ***.*** mean.service ***.***
"END DEBUG


stop
end
process arrival

"Deterministic server time.
normally, mode is undefined

request 1 server(1)

"  Debug. Track times of service
let sv.time = time.v
"  End debug

work mean.service .seconds

relinquish 1 server(1)

end
```

```
process generator

normally, mode is undefined

define counter as integer variable

"DEBUG
print 1 line with mean.arrival, mean.service thus
interarrival time *******.******** interservice time ******.******
"ENDDEBUG

wait exponential.f (mean.arrival, 1) .seconds
"while time.v < duration do
while 1 > 0 do  " dummy loop. continue forever
   add 1 to counter
   if counter > 1000
      call printline
      let counter = 0
   endif

   add 1 to total.entries

   if N.Q.server > bufsize     " server queue full. This one will be lost
      add 1 to cell.count

"DEBUG
"     print 1 line with cell.count thus
"Cell lost.  cell.count = ****
"END DEBUG


   else

"     Record number of cells lost. Add 1 to each count for multiple cells.
      if cell.count > 0
         if cell.count le 10
            add 1 to cells.lost(cell.count)
         else
            add 1 to cells.lost(11)
         endif

"        There is room for an entry. Put in on the queue.
         let cell.count = 0
      endif
      activate an arrival now
   endif
   wait exponential.f (mean.arrival, 5) .seconds

loop
end
routine isdnsim

normally, mode is undefined
```

```
call resetall
let time.v = 0
activate a generator now
start simulation
call printline
return
end
routine printline

normally, mode is undefined

print 1 line with bufsize, lambda, mu, duration thus
Buffer **** lambda (arrival) ***.*** mu (service) ***.*** duration *****
print 3 lines with
cells.lost(1),
cells.lost(2),
cells.lost(3),
cells.lost(4),
cells.lost(5),
cells.lost(6),
cells.lost(7),
cells.lost(8),
cells.lost(9),
cells.lost(10),
cells.lost(11) thus
   1     2     3     4     5     6     7     8     9  10 11+

******* ******* ******* ******* ******* ******* ******* ******* *******
*** ***

"Show total number of cells processed
print 2 lines with total.entries thus

Total cells processed = *********

return
end
```

APPENDIX G

MINIMUM INTERRARRIVALS SIMULATION PROGRAM

```
Preamble
"    ISDN statistical multiplexor buffer simulation
"
"    Philip Branch, MTech project 1993.
"
"    Use SIMSCRIPT to model the behaviour of a multiplexor. In particular
"    determine whether cell (packet) loss occurs in bursts.
"
     normally, mode is undefined          "Force variable declaration
     Processes include generator, arrival    "Queue entries
     Resources include server             "Queue server

"Variable declarations
     Define head.date, head.time as text variable
     Define bufsize, duration as integer variable
     Define lambda, mu as real variable
     Define cell.count as integer variable
     Define cells.lost as integer, 1-dimensional array
     Define mean.arrival, mean.service as real variables
     Define total.entries as integer variable
     Define minserve as integer variable
     Define last.time as real variable
     Define arrival.count as integer variable

" Time definitions
     Define .seconds to mean days
     Define .milliseconds to mean hours
     Define .microseconds to mean minutes

end    "preamble
Main

"Main routine
"===========

normally, mode is undefined

"Change time scale
     Let hours.v  = 1000
     let minutes.v = 1000

" Reserve space for statistic records
     reserve cells.lost(*) as 11
```

```
" Setup 1 server
     create every server(1)
     let U.server(1) = 1
" Set up input and output files
     open unit 2 for input, file name is "C:\SIM\PARAM1.DAT"
     use 2 for input

     open unit 3 for output, file name is "C:\SIM\LOG.DAT"
     use 3 for output

" Get date and time for heading
     call date.r yielding head.date, head.time

"Print heading
     print 5 lines with head.date, head.time thus
     ISDN cell loss simulation MIN TIME M/D/1 Date ********** Time ********


     1    2    3    4    5    6    7    8    9    10   >10


     read bufsize, lambda, mu, duration, minserve as I 6,  I 6, I 6, I 6, I 6

     let mean.arrival = 1.0 / lambda
     let mean.service = 1.0 / mu
     let time.v = 0
     activate a generator now
     start simulation
     call printline

stop
end
process arrival

normally, mode is undefined

request 1 server(1)
work mean.service .seconds

relinquish 1 server(1)

end
process generator

normally, mode is undefined

"Define time.temp as real variable
Define time.minserve as real variable

"DEBUG
print 1 line with mean.arrival, minserve thus
interarrival time *******.********  *****.*****
"ENDDEBUG
```

```
let time.v = 0
let arrival.count = 0
wait exponential.f (mean.arrival, 1) .seconds
let time.minserve = mean.service / 2

"DEBUG
print 1 line with mean.service, time.minserve, mean.service thus
service time *******.****** min arrival time ****.***** mean serv ****.*****
"DEBUG END

while  duration > 0 do          "Dummy test. Loop forever

  add 1 to arrival.count

  if arrival.count > 100000      "Print a line every 100,000 arrivals
    let arrival.count = 0
    call printline
  endif

  add 1 to total.entries

  if N.Q.server > bufsize      " server queue full. This one will be lost
    add 1 to cell.count
  else

"    Record number of cells lost. Add 1 to each count for multiple cells.
    if cell.count > 0
      if cell.count le 10
        add 1 to cells.lost(cell.count)
      else
        add 1 to cells.lost(11)
      endif

"      There is room for an entry. Put in on the queue.
      let cell.count = 0
    endif
    activate an arrival now
  endif

  work time.minserve .seconds    "Allow a minimum time between arrivals

  wait exponential.f (mean.arrival, 1) .seconds


"DEBUG
"     let time.temp = time.v - last.time
"     print 4 line with N.Q.server, bufsize, time.v, last.time, minserve,
"                time.temp, time.minserve thus
"N.Q.server      ***** bufsize *****
"time.v     *****.**** last.time *****.***** minserve *****.****
"interrarival time  ****.***** time.minserve *****.*****
"
```

```
"ENDDEBUG

loop
end
routine isdnsim

normally, mode is undefined

call resetall
let time.v = 0
activate a generator now
start simulation
call printline
return
end
routine printline

normally, mode is undefined

print 1 line with
cells.lost(1),
cells.lost(2),
cells.lost(3),
cells.lost(4),
cells.lost(5),
cells.lost(6),
cells.lost(7),
cells.lost(8),
cells.lost(9),
cells.lost(10),
cells.lost(11) thus
 ****** ****** ****** ****** ****** ****** ****** ****** ****** ******
******

"Show total number of cells processed
print 2 lines with total.entries thus

Total cells processed = *********

return
end
```

APPENDIX H

SWITCHED POISSON PROCESS SIMULATION PROGRAM

```
Preamble
"    ISDN statistical multiplexor buffer simulation
"
"    Philip Branch, MTech project 1993.
"
"    Use SIMSCRIPT to model the behaviour of a multiplexor. In particular
"    determine whether cell (packet) loss occurs in bursts.
"
     normally, mode is undefined          "Force variable declaration
     Processes include generator, arrival    "Queue entries
     Resources include server              "Queue server

"Variable declarations
"---------------------

" Report headings
     Define head.date, head.time as text variable

" Parameters
     Define bufsize, duration as integer variable
     Define gamma, omega as real variable
     Define lambda1, lambda2, mu as real variable

" Performance monitoring variables
     Define cell.count as integer variable
     Define cells.lost as integer, 1-dimensional array
     Define mean.arrival, mean.service as real variables
     Define total.entries as integer variable

" Time definitions
     Define .seconds to mean days
     Define .milliseconds to mean hours
     Define .microseconds to mean minutes

end    "preamble
Main

"Main routine
"============

normally, mode is undefined

"Change time scale
     Let hours.v  = 1000
```

```
    let minutes.v = 1000

" Reserve space for statistic records
    reserve cells.lost(*) as 11

" Setup 1 server
    create every server(1)
    let U.server(1) = 1
" Set up input and output files
    open unit 2 for input, file name is "C:\SIM\SPP\SPP.DAT"
    use 2 for input

    open unit 3 for output, file name is "C:\SIM\SPP\LOG.DAT"
    use 3 for output

" Get date and time for heading
    call date.r yielding head.date, head.time

"Print heading
    print 2 lines with head.date, head.time thus
    ISDN cell loss simulation          M/D/1 Switched Poisson
                                 Date ********** Time ********


" Buffer size, first Poisson rate, Second Poisson rate, service rate,
"  time in each Poisson

    read bufsize, gamma, omega, lambda1, lambda2, mu, duration
    as I 2, 5 D(10,5), I 8
"   as ** ****.*** ****.*** ****.*** ****.*** ****.*** ********

    let mean.arrival = 1.0 / lambda1
    let mean.service = 1.0 / mu
    let time.v = 0
    activate a generator now
    start simulation
    call printline

"DEBUG
    print 1 line with lambda1, mean.arrival, mu, mean.service thus
lambda ***.*** mean.arrival ***.*** mu ***.*** mean.service ***.***
"END DEBUG


stop
end
process arrival

"Deterministic server time.
normally, mode is undefined

request 1 server(1)
```

```
work mean.service .seconds

relinquish 1 server(1)

end
process generator

normally, mode is undefined

define counter, switch.flag as integer variable
define last.time as real variable
define gamma.time as real variable
define omega.time as real variable
define switch.time as real variable

let gamma.time = 1/gamma
let omega.time = 1/omega
let switch.time = exponential.f(omega.time,1)

print 6 lines with gamma, omega, lambda1, lambda2, mu, duration,
          gamma.time, omega.time
thus
PARAMETERS:-   gamma   omega  lambda1  lambda2   mu    duration
       ****.**** ****.**** ****.**** ****.**** ****.**** ********


TIMES:-     gamma.time  omega.time
       *******.***** ******.****

wait exponential.f (mean.arrival, 1) .seconds

"while time.v < duration do
while 1 > 0 do        " dummy loop. continue forever
  add 1 to counter
"  if counter > 1000
  if counter > 10000
     call printline
     let counter = 0
  endif

  add 1 to total.entries

  if N.Q.server > bufsize    " server queue full. This one will be lost
     add 1 to cell.count

  else

"    Record number of cells lost. Add 1 to each count for multiple cells.
     if cell.count > 0
        if cell.count le 10
           add 1 to cells.lost(cell.count)
        else
           add 1 to cells.lost(11)
        endif
```

```
"       There is room for an entry. Put in on the queue.
            let cell.count = 0
        endif

        activate an arrival now
    endif

" Time to flip to other Poisson rate ?
    if time.v - last.time > switch.time
        let last.time = time.v
        if switch.flag = 1
            let mean.arrival = 1 / lambda1
            let switch.flag = 2

"           Calculate time to switch over to other rate

            let switch.time = exponential.f(gamma.time,1)
        else
            let mean.arrival = 1 / lambda2
            let switch.flag = 1

"           Calculate time to switch over to other rate

            let switch.time = exponential.f(omega.time,1)

        endif

"DEBUG
" print 1 line with mean.arrival, switch.flag, switch.time thus
" mean.arr= ****.****, mean.swit= ****.****,switch.time= ****.****
"DEBUG END

    endif
    wait exponential.f (mean.arrival, 1) .seconds

loop
end
routine isdnsim

normally, mode is undefined


let time.v = 0
activate a generator now
start simulation
call printline
return
end
routine printline

normally, mode is undefined
```

```
define count, num.runs, tot.runs as integer variable
define perc as real, 1-dimensional array

reserve perc(*) as 11
print 3 lines with
cells.lost(1), cells.lost(2), cells.lost(3), cells.lost(4), cells.lost(5),
cells.lost(6), cells.lost(7), cells.lost(8), cells.lost(9), cells.lost(10),
cells.lost(11) thus
   1    2    3    4    5    6    7    8    9  10 11+

******* ******* ******* ******* ******* ******* ******* ******* *******
*** ***

" Calculate percentages of run loss
let tot.runs = cells.lost(1) + cells.lost(2) + cells.lost(3) + cells.lost(4) +
        cells.lost(5) + cells.lost(6) + cells.lost(7) + cells.lost(8) +
        cells.lost(9) + cells.lost(10) + cells.lost(11)

let num.runs = tot.runs
for count = 2 to 11 by 1
do
   let num.runs = num.runs - cells.lost(count - 1)
   let perc(count) = 100 * num.runs / tot.runs
loop

print 3 lines with perc(2), perc(3), perc(4), perc(5),
           perc(6), perc(7), perc(8), perc(9)
thus

     ***.*** ***.*** ***.*** ***.*** ***.*** ***.*** ***.*** ***.***

"Show total number of cells processed
print 3 lines with  total.entries thus
Total cells generated = *********


return
end
```