PATTERNS OF CELL LOSS IN ATM MULTIPLEXERS

BY

Ram Mohan Rao Anne

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

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

UNIVERSITY OF TASMANIA

DECEMBER, 1995

PATTERNS OF CELL LOSS IN ATM MULTIPLEXERS

BY

Ram Mohan Rao Anne

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

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

UNIVERSITY OF TASMANIA

DECEMBER, 1995

ACKNOWLEDGMENTS

I would like to thank the following people for their assistance and support during the research and preparation of my thesis. Dr David Lewis for the supervision, Professor Thong Nguyen for his encouragement, and Mr Jason Pieloor for his timely help and suggestions.

Abstract

In this project report,a study of packet loss in a finite -buffered statistical multiplexer for superimposed packetized voice sources (fixed length packets) is carried out. .Given that an ATM cell is lost at a finite buffer ,there is an altered probability that the following cells are lost. First starting with an M/M/1/K & M/D/1/K type queue system, the study of conditional cell loss probabilities is carried out .

Next the SPP/D/1/K and IPP/D/1/K queue analysis is done to estimate the cell loss at the statistical multiplexers these queues model the bursty type of sources.

In Asynchronous transfer mode(ATM) networks single multimedia source can launch the bit rates of in the order of 35 Mb/s using VBR(variable bit rate coding) methods .Hence at 155 Mb/s interface four sources can easily saturate the link eg. High Definition Television(HDTV),desktop work stations with visual and graphics computing powers.However in this report the main focus is on voice sources which are bursty in nature.

Computer simulation techniques are used to analyse the cell loss probabilities ,conditional cell loss probabilities for such type of bursty(voice sources),the conclusion is that ,both overall cell loss rate and conditional cell loss are much higher compared to the less bursty type of sources

Therefore to maintain the quality of service(QOS) and throughput techniques like adaptive control ,should be applied at the user network interface(UNI) of the link which dynamically controls the arrival rate by switching the coder to a different compression ratio (that is changing the coding rate)

TABLE OF CONTENTS

INTRODUCTION:

1.0 DESCRIPTION OF TRANSFER MODES

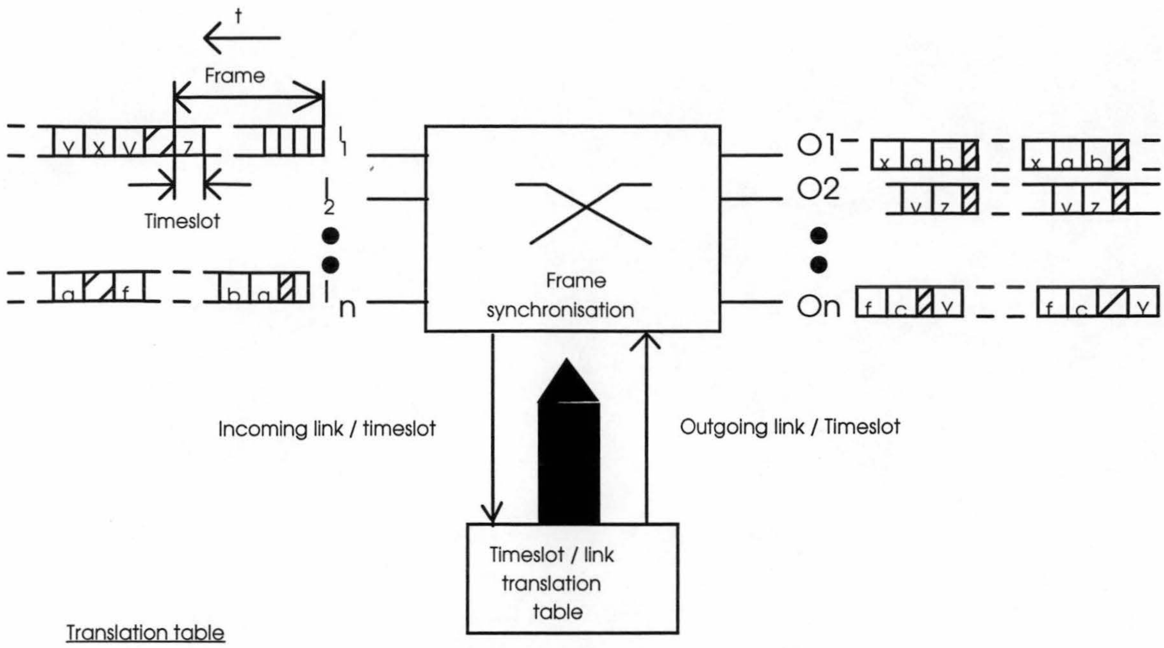
The words transfer modes are used by CCITT to describe technique which is used in a telecommunication network, covering aspects related to transmission, multiplexing and switching.Many different transfer m.odes exists in the telecommunication world. The most important transfer modes which have been considered for BISDN and their applicability to the BISDN requirements are being discussed, mainly with respect to time and semantic transparency.

2.0 Circuit switching

This transfer mode has long been in telephone networks, and is still applied in NISDN. In this classical approach, a circuit is established for the complete duration of the connection. This is based on the TDM (time division multiplexing) principle to transport the information from one node to another.

The information is transferred with a certain repetition frequency (e.g. 8 bits every 125 μs for 64 kbit/s or 1000 bits every 125 μs for 8 Mbit/s). The basic unit of this repetition frequency is called a time slot. Several connections are time multiplexed over one link by joining together several times slots in a frame, which is again repeated with a ceertain frequency. A connection will always use the same time slot in the frame during the complete duration of the session. The circuit switching can be performed internally by space switching, time switching or a combination of both.

The switching of a circuit of an incoming link to an outgoing link is controlled by a translation table which contains the relation of the incoming link and the slot number, to the outgoing link and the associated slot number. If reference is made to the following translation table:



<u>Translation table</u>

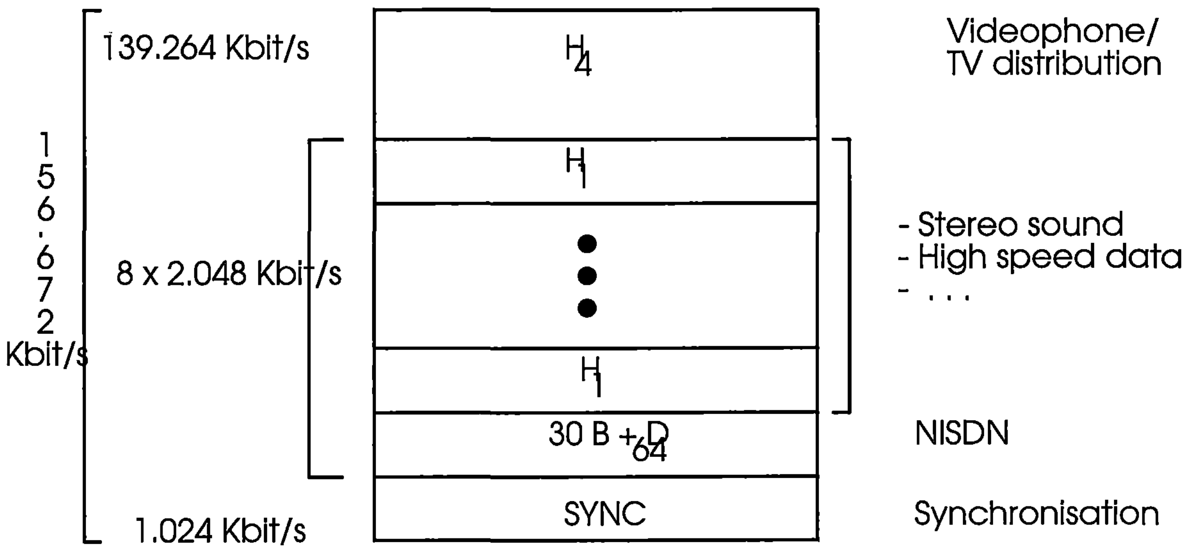
Incoming link	Time slot	Outgoing Link	Time slot
I_1	1 2 3 ... m	O_2 O_1 O_n ... O_2	2 3 m ... 1
I_2	1 2 3 ... m		
I_n	1 2 3 ... m	O_1 O_1 ... O_n	2 1 ... 2

As can be seen in the above diagram, connection V on the incoming link I_1, occupies time slot 1, and will always be transferred to outgoing link O_2 and time slot 2. Only when the connection finishes, can time slot 1 of link I_1 be switched to another slot on another link. This relation is unchanged for the complete duration of the connection, i.e. this relation determines the "circuit". The relations between all slots on all incoming links and the outgoings links are determined in a translation table. This translation table is only modified when a connection is set up or released.

3.0 Multirate circuit switching

To overcome the flexibility of only a single bit rate as is in pure circuit switching, a more enhanced version was developed, called MultiRate Circuit Switching (MRCS).

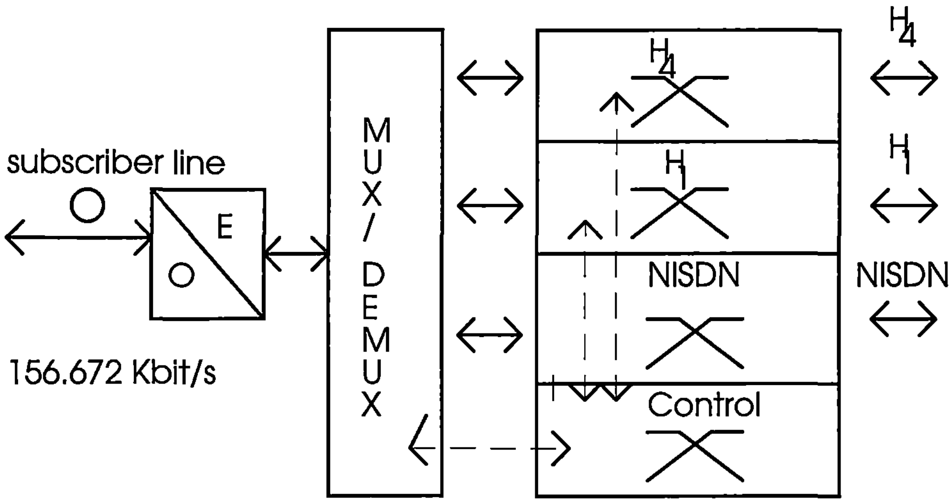
The transmission system of MRCS networks uses the same TDM(Time Division Multiplexing) format as in pure circuit switching with a fixed basic channel rate. However, one connection can now allocate n (n>= 1) basic channels. So every connection can be built as a multiple of the basic channel rate. This option is currently retained for instance for videophony in NISDN.



In the above diagram, a channel of 139.264 Kbit/s is multiplexed with 7 channels of 2.048 kbit/s, 30 B channels and a D channel of 64 kbit/s for signalling. In addition, a synchronisation channel is defined which allows the receiving side of the link/terminal to determine the frame boundary.

The H4 channel of 139.264 kbit/s was defined to be used for videophone and TV distribution services. The H1 channels (2.048 kbit/s) could be used for HIFI sound, high speed data and other medium speed services. The NISDN channel could be used for voice and other low speed data applications.

The switching systems developed for such a solution consist of overlay switches, each switch individually tailored to one specific basic channel rate. A possible switch architecture is given in the following diagram.



We can see that the information coming from (going to) the subscriber access line is de multiplexed/multiplexed into the different channels which are then connected to the different switches. Each individual switch (H4, H1, NISDN) has to be designed and manufactured individually. Control, operation and maintenance can be common to the different switches.

This architecture will inefficiently use its resources. Suppose that all H1 channels are occupied, then no additional H1 connections can be established, even when the H4 switch is completely free.

4.0 Fast circuit switching

In order to extend the concepts of circuit switching to sources with fluctuating and bursty nature, fast circuit switching (FCS) has been proposed. The resources in the FCS network are only allocated when information is sent, and released when no information is sent.

One can state that FCS relates to circuit switching, as connectionless packet switching (datagram switching) relates to connection-oriented packet switching, in the sense that the resources are not permanently allocated, but will only be allocated when needed. This allocation is performed per burst, as is the case in data gram switching, but now under control of fast "associated" signalling, whereas this "signalling" information is present in the header of a data gram for data gram switching.

At call set-up, users request a connection with a bandwidth equally some integer multiple of the basic rate; the system will not allocate the resources, but store inside the switch information on the required bandwidth and the selected destination, and allocate a header in the signalling channel, identifying that connection. When the source starts sending information, the header will indicate that the source has information, requiring from the switch to allocate the necessary resources immediately.

5.0 Packet switching

In packet switching networks, user information is encapsulated in packets which are containing additional information used inside the network for routing, error correction and flow control.

These networks such as X.25, were designed in the sixties, at the time when only poor to medium quality transmission links were available.

n order to offer an acceptable end-to-end performance on each link of the network, complex protocols were therefore necessary basically performing error and flow control on every link of the connection. This link-by-link error control was required because of the low quality of the links to ensure that the traffic increase was not too large to guarantee the required semantic transparency.

These packets have a variable length and thus require a rather complex buffer management inside the network. The operating speed was not high, so software buffer control was very well possible. This low speed caused a large delay. However, since real time services did not have to be transmitted over these networks, the lack of time transparency was not a problem.
The higher complexity protocols substantially increase the processing requirements and switching delay inside the network. This makes it very difficult to apply the packet switching technique for real time services and for high speed services at ten/hundreds of Mbit/s. However, it must be admitted that packet switching is still very efficient and successful for low speed data transfer.

6.0 Fast Packet Switching:

Fast packet switching is a common concept that covers several alternatives ,all with the same characteristics ,that is packet switching with minimal functionality in the network.

The name fast packet switching is applicable since it allows a the systems to operate at higher speeds.ATM(Asynchronous Transfer Mode) is another name for fast packet switching.The following are the advantages of ATM:

1.Flexible and future safe

2.Efficient in the use of its available resources

3.One universal network.

7.0 ATM General Introduction:

Motivation for ATM

In order to understand what ATM is all about, a brief introduction to STM is in order. ATM is the complement of STM which stands for "Synchronous Transfer Mode". STM is used by telecommunication backbone networks to transfer packetized voice and data across long distances. It is a circuit switched networking mechanism, where a connection is established between two end points before data transfer commences, and torn down when the two end points are done. Thus the end points allocate and reserve the connection bandwidth for the entire duration, even when they may not actually be transmitting the data. The way data is transported across an STM network is to divide the bandwidth of the STM links (familiar to most people as T1and T3 links) into a fundamental unit of transmission called time-slots or buckets. These buckets are organized into a train containing a fixed number of buckets and are labeled from 1 to N. The train repeats periodically every T timeperiod, with the buckets in the train always in the same position with the same label. There can be up to M different trains labelled from 1 to M, all repeating with the time period T, and all arriving within the time period T.

The parameters N, T, and M are determined by standards committees, and are different for Europe and America. The timeperiod T is a historic legacy of the classic Nyquist sampling criteria for information recovery. It is derived from sampling the traditional 3 to 4Khz bandwidth of analog voice signals over phone lines at twice its frequency or 8Khz, which translates to a timeperiod of 125 usec. This is the most fundamental unit in almost all of telecommunications today, and is likely to remain with us for a long time.

On a given STM link, a connection between two end points is assigned a fixed bucket number between 1 and N, on a fixed train between 1 and M, and data from that connection is always carried in that bucket number on the assigned train. If there are intermediate nodes, it is possible that a different bucket number on a different train is assigned on each STM link in the route for that connection. However, there is always one known bucket reserved a priori on each link throughout the route. In otherwords, once a time-slot is assigned to a connection, it generally remains allocated for that connections sole use throughout thelife time of that connection.

To make the ATM concept clear , imagine the same train arriving at a station every T tampered. Then if a connection has any data to transmit, it drops its data into its assigned bucket(time-slot) and the train departs. And if the connection does not have any data to transmit, that bucket in that train goes empty. No passengers waiting in line can get on that empty bucket. If there are a large number of trains, and a large number of total buckets are going empty most of the time (although during rush hours the trains may get quite full), this is a significant wastage of bandwidth, and limits the number of connections that can be supported simultaneously. Furthermore, the number of connections can never exceed the total number of buckets on all the different trains(N*M).

The telecommunications companies are investigating fiber optic cross country and cross oceanic links with Gigabit/sec speeds,and would like to carry in an integrated way, both real time traffic such as voice and hi-res video which can tolerate some loss but not delay, as well as non real time traffic such as computer data and file transfer which may tolerate some delay but not loss. The problem with carrying these different characteristics of traffic on the same medium in an integrated fashion is that the peak bandwidth requirement of these traffic sources may be quite high as in high-res full motion video, but the duration for which the data is actually transmitted may be quite small. In other words, the data comes in bursts and must be transmitted at the peak rate of the burst, but the average arrival time between bursts may be quite large and randomly distributed. For such bursty connections, it would be a considerable waste of bandwidth to reserve them a bucket at their peak bandwidth rate for all times, when on the average only 1 in 10 bucket may actually carry the data.

It would be nice if that bucket could be reused for another pending connection. And thus using STM mode of transfer becomes inefficient as the peak bandwidth of the link,peak transfer rate of the traffic, and overall burstiness of the traffic expressed as a ratio of peak/average, all go up. In thejudgement of the industry pundits, this is definitely the indicated trend for multimedia integrated telecommunications and data communications demands of global economies in the late 90's and early 21st century.

Hence ATM is conceived. It was independently proposed by Bellcore, the research arm of AT&T in the US, and several giant telecommunications companies in Europe, which is why there may be two possible standards in the future. The main idea here was to say, instead of always identifying a connection by the bucket number, just carry the connection identifier along with the data in any bucket, and keep the size of the bucket small so that if any one bucket got dropped enroute due to congestion, not too much data would get lost, and in some cases could easily be recovered. And this sounded very much like packet switching,so they called it "Fast packet switching with short fixed length packets." And the fixed size of the packets arose out of hidden motivation from the telecommunications companies to sustain the same transmitted voice quality as in STM networks, but in the presence of some lost packets on ATM networks.

Thus two end points in an ATM network are associated with each other via an identifier called the "Virtual Circuit Identifier"(VCI label) instead of by a time-slot or bucket number as in a STM network. The VCI is carried in the header portion of the fast packet. The fast packet itself is carried in the same type of bucket as before, but there is no label or designation for the bucket . The terms fast packet, cell, and bucket are used interchangeably in ATM literature and refer to the samething.

8.0 Statistical Multiplexing:

Fast packet switching is attempting to solve the unused bucket problem of STM by statistically multiplexing several connections on the same link based on their traffic characteristics. In other words, if a large number of connections are very bursty (i.e. their peak/average ratio is 10:1 or higher), then all of them may be assigned to the same link in the hope that statistically they will notal burst at the same time. And if some of them do burst simultaneously, that that there is sufficient elasticity that the burst can be buffered up and put in subsequently available free buckets. This is called statistical multiplexing, and it allows the sum of the peak bandwidth requirement of all connections on a link to even exceed the aggregate available bandwidth of the link under certain conditions of discipline. This was impossible on an STM network, and it is the main distinction of an ATM network. The discipline conditions under which statistical multiplexing can work efficiently in an ATM network are an active area of research and experimentation in both academia and industry. It has also been a prodigious source of technical publications and considerable speculations. Telecommunications companies in the US, Europe, and Japan as well as several research organisations and standards committees are actively investigating how BEST to do statistical multiplexing in such a way that the link bandwidth in an ATM network is utilised efficiently, and the quality of service requirements of delay and loss for different types of real time and non real time as well as bursty and continuous traffics are also satisfied during periods of congestion. The reason why this problem is so challenging is that if peak bandwidth requirement of every connection is allocated to it, then ATM just degenerates into STM and no statistical advantage is gained from the anticipated bursty nature of many of the futurebroadband integrated traffic profiles.

9.0 Types of User Network Interfaces (UNI) for ATM:

It is envisioned that the ATM network service providers may offer several types of interfaces to their networks. One interface It is envisioned that the ATM network service providers may offer several types of interfaces to their networks. One interface that is likely to be popular with companies that build routers and bridges for local area networks is a Frame based interface.One or more of the IEEE 802.X or FDDI frames may be supported at the UNI, with frame to ATM cell conversion and reassembly being done inside the UNI at the source and destination end points respectively. Thus a gateway host on a localarea network might directly connect its ethernet, token ring, fddi, or other LAN/MAN interface to the UNI, and thus bridge two widely separated LANs with an ATM backbone network. This will preserve the existing investment in these standards and equipments, and enable a gradual transition of the ATM networks into the market place.

An alternate interface likely to be more popular in the longer run, and for which the concept of Broadband-ISDN really makes sense, is direct interface at the UNI with standard ATM cells. Such a streaming interface can hook subscriber telecom,datacom, and computer equipment directly to the network, and allow orders of magnitude greater performance and bandwidth utilization for integrated multimedia traffic of the future. Thus it is by no accident that the IEEE 802.6 packet for the MAC layer of the Metropolitan Area Network (MAN) DQDB protocol (Distributed Queue Dual Bus) looks very much like an ATM cell.

It is quite likely that companies may crop up (if they have not already done so) to design ATM multiplexers for interface to theUNI of a larger ATM backbone network. Especially if the CCITT succeeds in standardizing an interface definition for UNI, it will be an additional boon to this market. The multiplexers with multiple taps on the user side can connect to one fat ATM pipe at the network side. Such a multiplexer would hide the details of ATM network interface from the user, and provide simple,easy to use, low cost ATM cell taps to hook the user equipment into.

Companies with investment in existing STM networks such as T1 and T3 backbones, are likely to want a direct T3 interface to the UNI, thus allowing them to slowly integrate the newer ATM technology into their existing one. Thus it is possible to see aflurry of small startups in the future rushing to make large T3 multiplexers for connecting several T3 pipes into one large ATMpipe at the UNI.

Typically, an ATM network will require a network management agent or proxy to be running at every UNI which can communicate and exchange administrative messages with the user attachments at the UNI for connection setup, tear down, and flow control of the payload using some standard signalling protocol. A direct user attachment at the UNI is likely to cost more and be more complex, then a user attachment to something which in turns interfaces to the UNI.

10.0 Connections on an ATM network:

As in STM networks, where a datum may undergo a time-slot-interchange between two intermediate nodes in a route, the VCI label in an ATM cell may also undergo a VCI label interchange at intermediate nodes in the route. Otherwise, the connections in the ATM network look remarkably similar to STM networks.

It may be possible to have certain reserved VCI labels similar in concept to "well known port definitions of UDP and TCP", as identifiers for special well known services that may be provided by the network. However very little can be assumed about the dynamically assigned VCI labels for most user related connections.

A service provider is unlikely to accede to any special request by any one service requester to allocate it a chunk of VCIlabels, unless the network itself is owned by the service requester. Furthermore, the address space of the VCI labels is limited to 24 bits and only designed to identify the connections between two points on a single link. The address space would disappear rather quickly if customers started to requisition portions of the VCI label for their own semantics.

If there is a specific need to assume semantics for the VCI label outside of the ATM network, i.e. require it to be within acertain range on the user attachments at the UNI, it is probably best to provide a lookup table in hardware inside the user attachments which can map the pretty much randomized VCI label assigned by the network to n bits of a new label to whichthe user attachment can assign its own semantics to its silicon's content.

11.0 Protocol Layer Of ATM:

As is probably evident by now, ATM is designed for switching short fixed length packets in hardware over Gigabit/sec links across very large distances. Thus its place in the protocol stack concept is somewhere around the data link layer. However it does not cleanly fit in to the abstract layered model, because within the ATM network itself, end-to-end connection, flowcontrol, and routing are all done at the ATM cell level. So there are a few aspects of traditional higher layer functions present in it. In the OSI reference model, it would be considered layer 2 (where layer 1 is the physical layer and layer 2 is the datalink layer in the internet protocol stack). But it is not very important to assign a clean layer name to ATM, so long as it is recognized that it is a hardware implemented packet switched protocol using 53 byte fixed length packets.

What is perhaps more relevant is how will all this interact with current TCP/IP and IP networks in general, and with applications which want to talk ATM directly in particular. A convenient model for an ATM interface is to consider it another communications port in the system. Thus from a system software point of view, it can be treated like any other data link layerport. Thus for instance, in IP networks connected via gateways to ATM backbones, the model would be no different then it presently is for a virtual circuit connection carried over an STM link except that an IP packet over an ATM network would get fragmented into cells at the transmitting UNI, and reassembled into the IP packet at the destination UNI. Thus a typical protocol stack might look like this: -------

Data

TCP

IP

ATM Adaptation Layer

ATM Datalink layer

Physical Layer (SONET STS-3c STS-12 STS-48)

Thus just like an ethernet port on a host is assigned an IP address, the ATM port may also be assigned an IP address. Thus the IP software in a router decides which port to send a packet to based on the IP address, and hands the packet to the port. The port then does the right thing with it. For an ethernet port, the ethernet header is tacked on and the Frame transmitted in ethernet style. Similarly, for an ATM port, the IP datagram is fragmented into cells for which an ATM adaptation layer is specified in the standards. The fragmentation and reassembly is done in hardware on the sending and receiving sides. A VCI label acquired via an initial one time connection establishment phase, is placed in the header of each cell, and the cells are drained down the fat ATM datalink layer pipe. On the receiving side, the cells are reassembled in hardware using the ATM adaptation layer, and the original IP packet is reformulated and handed to the receiving host on the UNI. The adaptation layer is not a separate header, but is actually carried in the payload section of the ATM cell as discussed earlier.

For direct interface to an ATM cell stream from an application, new interfaces have to be designed in the software that can provide the application with nice and fast mechanisms for connection establishment, data transfer, keep alive, tear down, and even application level flow control. In this case the software processing steps may look like this:

```
---------------------------
Application Streaming Data
---------------------------
OS interface to application
---------------------------
ATM virtual circuit management/signalling
---------------------------
Driver interface to ATM
---------------------------
ATM
---------------------------
```

where the ATM virtual circuit management represents software which understands the ATM header specifics, sets up and tears down connections, does demultiplexing of the payload to appropriate connections, and responds to whatever standard signalling protocol is employed by the ATM interface at the UNI for connection management.

The physical layer specification is not explicitly a part of the ATM definition, but is being considered by the same subcommittees. T1S1 has standardized on SONET as the preferred physical layer, and the STS classifications refer to the speeds of the SONET link. STS-3c is 155.5 Mbit/sec. STS-12 is 622 Mbit/sec, and STS-48 is 2.4 Gbit/sec. The SONET physical layer specifications chalk out a world wide digital telecommunications network hierarchy which is internationally known as the Synchronous Digital Hierarchy (SDH). It standardizes transmission around the bit rate of 51.84 Mbit/sec which is also called STS-1, and multiples of this bit rate comprise higher bit rate streams. Thus STS-3 is 3 times STS-1, STS-12 is 12 times STS-1, and so on. STS -3c is of particular interest as this is the lowest bit rate expected to carry the ATM traffic, and is also referred to as STM-1 (Synchronous Transport Module-Level 1). The term SONET stands for Synchronous Optical Network and is the US terminology for SDH .

The SDH specifies how payload data is framed and transported synchronously across fiber optic transmission links without requiring all the links and nodes to have the same synchronised clock for data transmission and recovery (i.e. both the clock Broad band ISDN network be a true international network. The fundamental frequency around which the SDH id done is 8khz.
However all of this sits below the ATM layer and the ATM cells are transported across the physical layer as opaque payload,also called the SONET payload or the Synchronous Payload Envelope (SPE). The physical layer is independent of the payload type, and can just as easily carry STM cells as ATM cells. Reference to the standards documents is suggested for more details.

12.0 Flow control in ATM:

Unlike the reactive end to end flow control mechanisms of TCP in inter networking, the gigabits/sec capacity of ATM network generates a different set of requirements for flow control. If flow control was left on end to end feedback, then by the time the flow control message was received at the source, the source would have already transmitted over several Mbytes of data into the ATM pipe exacerbating the congestion. And by the time the source reacted to the flow control message, the congestion condition might have disappeared altogether unnecessarily quenching the source. The time constant of end to end feedback in ATM networks (actually feedback_delay * link_bandwidth product) may be so large that solely relying on the user attachmentsto keep up with the dynamic network is impractical. The congestion conditions in ATM networks are expected to be extremely dynamic requiring fast hardware mechanisms for relaxing the network to steady state, and necessitating the network itself to be actively involved in quickly achieving this steady state. Thus a simplistic approach of end to end closed loop reactive control to congestion conditions is not considered sufficient for ATM networks.

The present consensus among the researchers in this field is to use a holistic approach to flow control. They recommend employing a collection of flow control schemes.Yes,. An ATM cell may encounter congestion and suffer variable delay due to buffering within the ATM switches, and may even be dropped either due to congestion control or due to header checksum error. However an ATM connection always obeys causality, the cells in a connection (i.e. cells with the same VCI label) arrive in order at the destination. This is so because there is no store and forwarding in the network, cells travel over a single virtual circuit path, the ATM switches do not switch the cells in the same VCI out of order, and no re transmissions is done at any point in the ATM network.

Connectionless services are also supported on ATM networks, but these are implemented as a higher layer service layered over the ATM datalink layer. Thus cells in a connectionless service may arrive out-of-order because there might be multipleVCIs over multiple paths setup to deliver the connectionless data grams and cells may arrive over different paths in different order. Thus the fragmentation reassembly engine which implements the connectionless data grams, and which is layered on top of the basic connection oriented service of the ATM layer, must carry sequence numbers in the adaptation layer in each cell and correct any reordering of the cells at reassembly time. This is what the IEEE 802.6 protocol for MAN does to support itsconnectionless service class.

13.0 ATM- Reliability In Delivery:

There is no end-to-end reliable delivery service at the ATM layer. The ATM layer does not do any re transmissions and there are no end-to-end acknowledgments for what has been received. Reliable delivery service can be implemented as a layer on top of the basic connection oriented ATM layer, where acknowledgment of received data and re transmission of missing data can be done for connections requiring reliable delivery. Thus a TCP type transport layer protocol (layer 4 in the OSI model) layered on top of the ATM layer is required for guaranteed delivery.

14.0 Performance of an ATM interface:

Unlike STM networks, ATM networks must rely on considerable user supplied information for the traffic profile in order to provide the connection with the desired service quality. There are some sources of traffic which are easier to describe than others, and herein lies the cost/performance challenge for best bandwidth utilisation in an ATM interface.

An ATM network can support many types of services. Connection oriented as well as connection less. It can support services which may fall in any of the four categories (loss sensitive, delay sensitive), (loss insensitive, delay sensitive), (loss sensitive,delay insensitive), and (loss insensitive, delay insensitive). It can further reserve and allocate a fixed bandwidth for a connection carrying a continuous bit stream for isochronous traffic (repeating in time such as 8khz voice samples), allocate a bandwidth range for a variable bit stream for plesiochronous traffic (variable frequency such as interactive compressed video), as well as allocate no specific amount of bandwidth and rely on statistical sharing among bursty sources. It may also provide multiple priorities in any of the above categories. The services can span the entire gamut from interactive such as telephony and on-line data retrieval, to distributed such as video and stereo Hi-Fi broadcasts and multicasts for conferencing and database updates.

Thus the performance that one might get from ones ATM connection is very much dependent on the parameters that respecified at connection setup time. Just because the link bandwidth may be an STS-12, does not necessarily imply that the end to end payload bandwidth that the ATM interface can sustain will also be STS-12. It will in fact be considerably lower based on connection setup parameters and the quality of service request, and whether bandwidth was reserved or statistically multiplexed, and the load on the ATM network.

Typically, the ATM network may not permit 100% loading of any link bandwidth, and in fact user available bandwidth may not be allowed to exceed more than 80% of the peak bandwidth of the link. The UNI may start policing and/or denying newconnection requests on the link if utilization exceeds this amount. Add the approx 10% overhead of the 5 byte header in the 53byte cell, and the max sustainable payload throughput on an ATM cell stream interface may peak at 72% of the peak link bandwidth. And this does not include any adaptation layer overhead if present, signalling overhead, or physical layer overheads of SDH SONET framing and inter-cell spacing gaps.

And of course, application to application bandwidth may be even less, unless the software data path from the interface driver through the OS to the application (and vice versa) is very carefully optimized. It would hardly be received very well if the end-to-end throughput from application to application would turn out to be no better for an ATM port than for an ethernet or FDDI(fiber distributed data interface) port due to software overheads.

15.0 ATM Transmitter And Reciever :

The transmitter side is constrained by the flow control of the simultaneous connection streams by pacing the injection rate according to the respective negotiated class of service and bandwidth requirements. The receiver side is constrained by asynchronous reception of cells at a variable rate, and with buffering capacity for a large number of simultaneous connections each of which can be receiving data simultaneously. And if an adaptation layer is used, then the reassembly of these cells into a higher layer protocol data unit (PDU) must also be done in hardware by the receiver side. Thus a lot of thought is required in designing an ATM interface to a host system, poor design of which can cripple the system performance.

CHAPTER2:

1.0 ATM ARCHITECTURE:

The two factors that have led to devolpment of broadband-ISDN(B-ISDN)

1 The emerging needs for high speed communications

2. The enabling technologies to support these services in an integrated fashion .The high transmission capacity offered by optical fibres has made wide area transmission possible with low error rates .The emerging needs for broadband stem from the end users with applications such as high quality video and bulk data transfer.The synchronous digltal hierarchy (SDH) unlike the existing digital hierarchy ,has been standardized as a world wide accepted broad band interface .The universal transport processing(multiplexing and routing) is performed by asynchronous transfer mode.

2.0 B-ISSDN AND THE NETWORK EVOLUTIONARY PATHS:

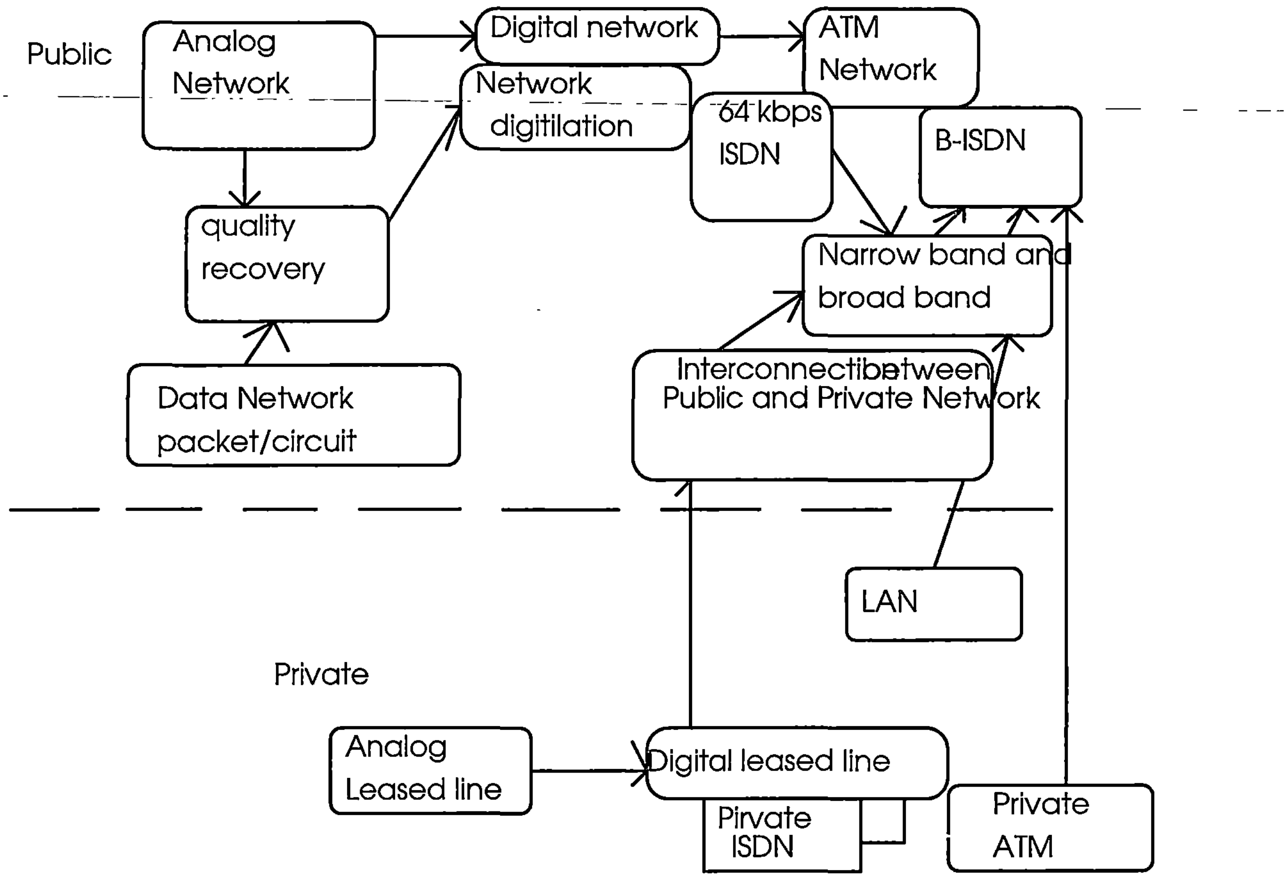
Telecommunication networks have long been based on analog technology dedicated to telephone service.The introduction of stored program control for switching and the common channel signalling system no7, for call control and signalling transport have enhanced the network capability to include the new services . The SS no 7 signalling network with access to service control points (SCP'S) which provide data base function ,supports enhanced services such as toll-free alternate billing services (ABS) ,CLASS(custom local area signalling service) mobile and personal communications .Furthermore ,digital technology improves the quality and economy of networks.In parallel ,packet switched networks and digital circuit switched networks have been developed as dedicated data networks in response to data service demand All of these activities have given impetus to a faster pace in the global development and deployment of narrow band ISDN. Recently ISDN offerings have enabled users to have integrated access to 64 kb/s and higher rate services, B-ISDN which is considered as an evolution of ISDN,should include

1.Broad band and multimedia communication capability .

2.intelligent network capability to provide enhanced network services

3.enhanced operation capability to achieve a highly reliable network performance even when there are failures and traffic fluctuations.

ATM is expected to play an important role in expanding the network capabilities towards B-ISDN .The following figure illustrates the significance of B-ISDN and ATM on network evolutionary path.



Network Evolution And B-ISDN

3.0 Information transfer

ATM is considered as a specific packet oriented transfer mode based on asynchronous time division multiplexing and the use of fixed length cells. Each cell consists of an information field and a header. The header is primarily used to identify cells belonging to the same virtual channel within the asynchronous time division multiplex, and to perform the appropriate routing. Cell sequence integrity is preserved per virtual channel.

The information field of ATM cells is carried transparently through the network. No processing like error control is performed on it inside the network. All services (voice, video, data, ...) can be transported via ATM, including connectionless services. To accommodate various services, several types of ATM Adaptation Layers (AAL) have been defined, depending on the nature of the service, to fit information into ATM cells, and to provide service specific functions (e.g. clock recovery, cell loss recovering, ...). The AAL specific information is contained in the information field of the ATM cell. Still a 1 bit indication in the header is also at the disposition of the user for this purpose.

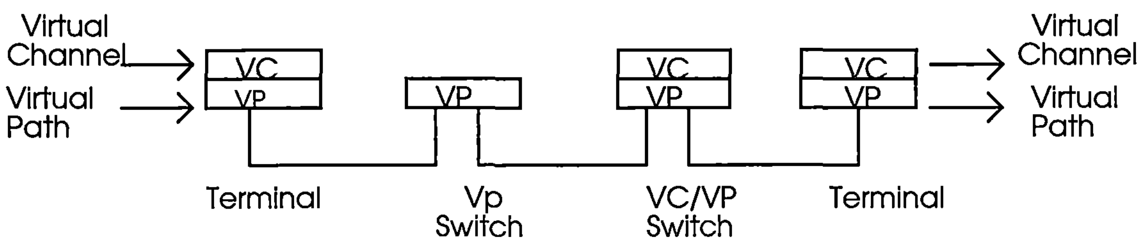
4.0 Routing

ATM is connection oriented. The header values are assigned to each section of a connection for the complete duration of the connection, and translated when switched from one section to another. Signalling and the user information are carried on separate virtual channels.

There are two possible types of connections:

(i) Virtual channel connections (VCC) and
(ii) Virtual path connections (VPC).

A VPC can be channel considered as an aggregate of VCCs. When switching on cells is to be performed, it must first be done based on the VPC, then on the VCC. This is shown in the diagram below.



There we see an entity which only performs VP switching, and another entity which performs both VP and VC switching. However, the VP switching part may be idle, resulting in a pure VC switch.

5.0 Resources

Since ATM is connection-oriented, connections are established either semi-permanently, or for the duration of a call, in case of switched services. This establishment includes the allocation of a VCI (virtual Channel Identifier) and/or VPI (Virtual Path Identifier), but also the allocation of the required resources on the throughput (bit rate) and Quality of Service. They may be negotiated between user and network for switched connections, during the call set-up phase and possible during the call.

6.0 ATM cell identifiers

ATM cell identifiers, both Virtual Path (VP) and Virtual Channel (VCI) identifiers, but also Payload Type (PTI) identifiers, support recognition of an ATM cell on a physical transmission medium. Recognition of the cell is the basis for all further operations. VPI and VCI are unique for cells belonging the same virtual connection on a shared transmission medium. As such, they are a limited resource although CCIT has recommended to make their number quite large (2^{28} - 16). Within a particular virtual circuit, cells may be further distinguished by their PTI, which cannot be allocated freely, but depends on the type of payload carried by the cell. This field indicates whether the cell is carrying user information to be delivered transparently through the network, or special network information. In case the field indicates network information, part of the information field indicates the type of network control whereas the remaining part of the information field indicates the type of network control whereas the remaining part of the information field may be processed inside the network.

A number of pre-assigned ATM cell identifiers have been chosen in the ATM layer for particular cell streams on the user-network interface and the node-network interfaces. They are necessary for enabling communication with the network, and to perform network management. Unassigned cell identifiers mark unused bandwidth. Other pre-assigned values define meta-signalling cells, point-to-point signalling cells, general broadcast signalling cells, physical layer OAM (operations and maintenance) cells and resource management cells.

7.0 Signalling

The negotiation between the user and network with respect to the resources, (VCI/VPI, throughput, QOS) is performed over a separate signalling virtual channel. The signalling protocol to be used over this signalling virtual channel will be an enhancement of those used in NISDN signalling.

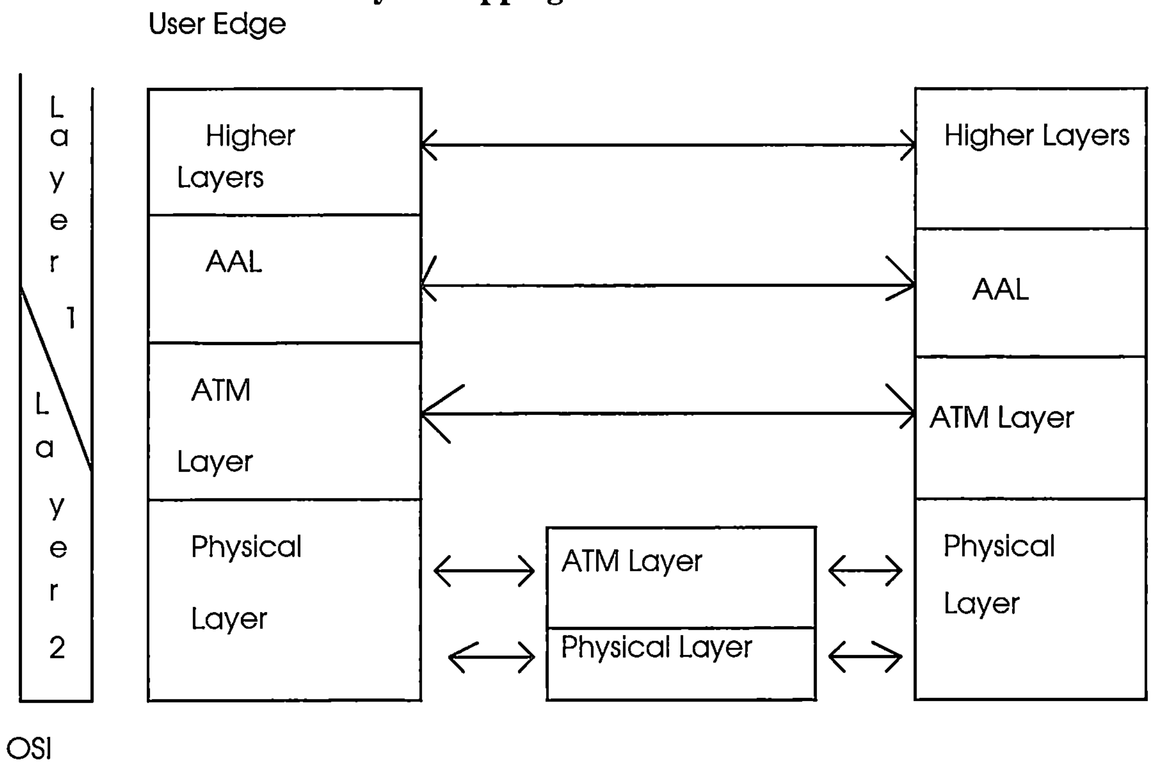
For point-to-point configurations on the UNI(User Network Interface), a pre-defined signalling channel exists. For point-to-multipoint configurations, where multiple terminals are connected to a single S_B interface via a shared medium, multiple signalling virtual channels (at least one per terminal) can be established via the meta-signalling channel. This meta-signalling channel is transported over a pre-assigned VCI/VPI defined on the user-network interface. the meta-signalling procedure performs the negotiation of VCI/VPI and required throughput for signalling with a terminal.

8.0 Operations and Maintenance:

CCITT has defined 5 levels of connectivity in the ATM transport network. The physical layer is composed of the lower 3: the regenerator section on the lowest level, then the digital section, and the transmission path. The ATM layer consists of the remaining 2: the virtual path, and above the virtual channel. Each of these levels has its own operations and maintenance flow, called F1 to F%, starting with F1 on the regenerator section level.

The transfer mode used for the information carried by these flows depends on the nature of the layer. For a physical layer based on SDH (Synchronous Digital Hierarchy), or for the emerging standards for mapping ATM and PDH(Plessiochronous Digital Hierarchy), flows F1 to F3 are carried in synchronous channels in the overhead of the physical layer. For a cell based physical layer, these flows are carried out by Physical Layer OAM(PL-OAM) cells. For the ATM layer itself, the F4 flows are carried in cells distinguished by pre-assigned VCIs in the virtual path, and the F5 flows are carried in cells distinguished by special PTI codes in the virtual circuit.

9.0 ATM And OSI Layer Mapping:



AAL-Cell segmentation and reassembly,timing control,flow control

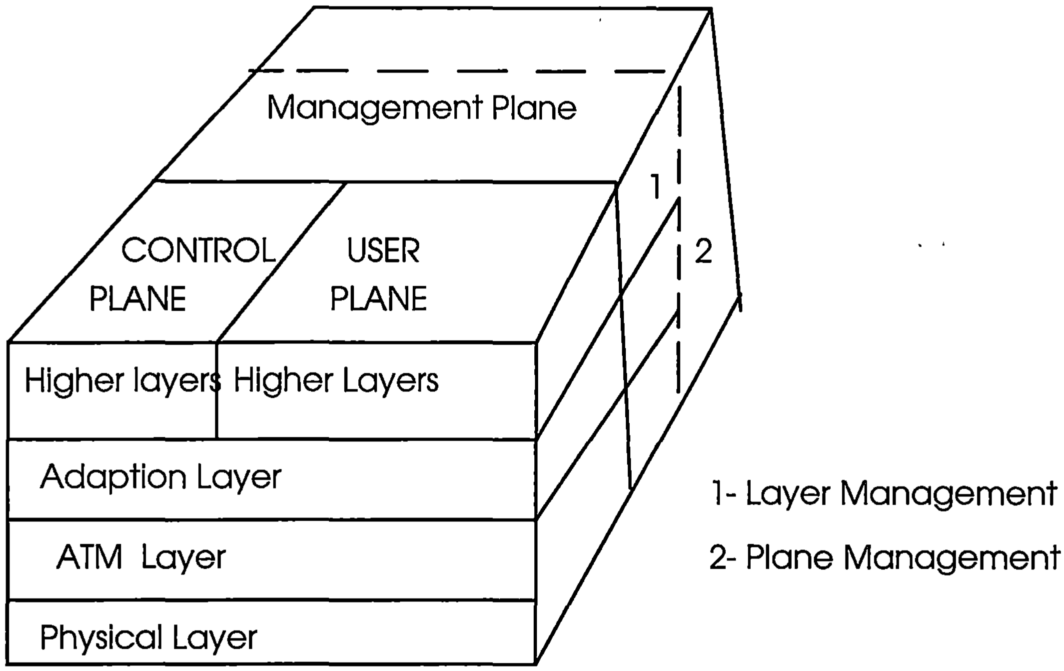
ATM Layer:Cell transport(VC/VP routing and multiplexing)

Physical Layer: Transmission payload to carry ATM cell stream

B-ISDN information transport protocol

10.0 B-ISDN Protocol Structure:

Protocol layering is essential to specify inter networking ,signalling and OAM(Operation And Maitainence).The protocol structure of B-ISDN is shown below .



B-ISDN ATM protocol reference model

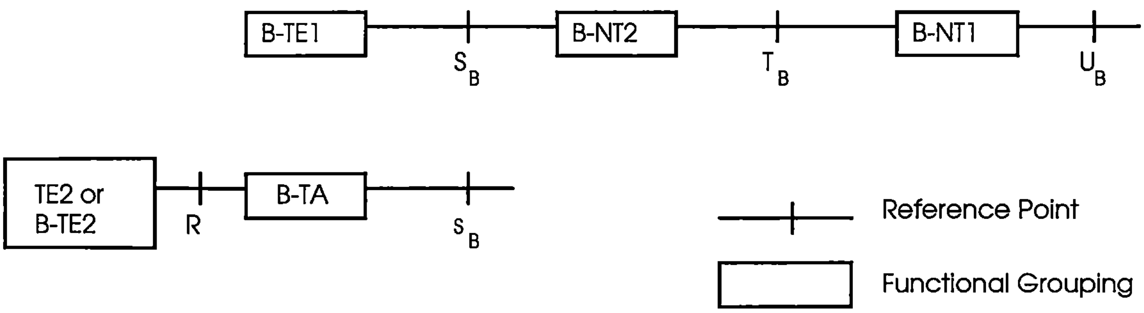
In the B-ISDN protocol structure ,two specific layers are related to ATM functions

1.An ATM layer that is common to all services and provides cell transfer capabilities
2.An ATM adaption layer(AAL)that is service dependent and supports higher layer functions of the user ,control and management planes

The boundary between the ATM layer and the AAL corresponds to the boundary between functions in the cell header and functions in the cell information field.

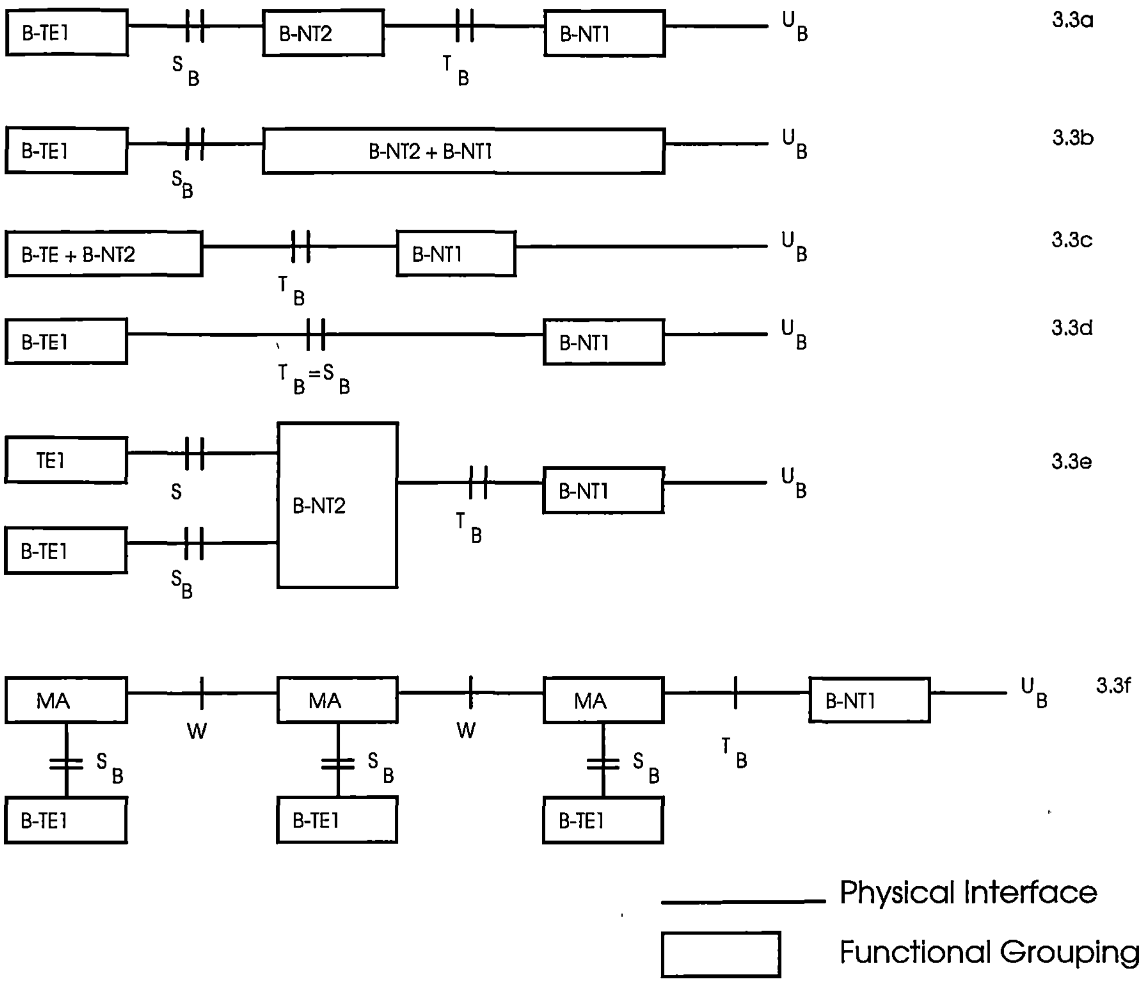
A reference configuration is a practical tool to define clear interfaces between different entities of the network and to define the functions of these different entities. The reference configurations for the user-network interface used for ISDN, defined in CCITT Recommendation I.411 are considered to be general enough to be applicable to all aspects of the BISDN accesses.

The reference configuration adopted by CCITT is given in the following diagram.



The Reference point s R, S, T and U as defined for NISDN are also valid in the BISDN case, as well as the functional groupings B-NT1, B-NT2 (Broadband Network Termination 1 or 2), B-TE1, B-TE2 (Broadband Terminal Equipment 1 or 2) and B-TA (Broadband Terminal Adaptor), as shown in the above diagram.

The reference configuration as described can be physically realised in different ways. A few examples for the user-network part are given in the diagram below, where the interfaces shown are now physically implemented, as well as the physical grouping of the functions.



In the first example (Fig 3.3a.), both the S_B and T_B interface are physically implemented resulting in a physical B-NT2 entity. This is different from the second example where only S_B is physically present (Fig 3.3b.(. In this example the B-NT2 and B-NT1 are physically co-located in a single entity. In Fig 3.3c. only TB will be physically implemented resulting in a combined B-TE and B-NT2 entity. In Fig. 3.3d., S_B and T_B are coinciding and therefore identical. No physical B-NT2 is present at the customer's premises. In fig. 3.3e., the physical interfaces between terminals and B-NT2 are both S_B and S (for narrowed ISDN), based on a centralised B-NT2 grouping. The B-NT2 can also be distributed, giving rise to a possible configuration shown in Fig. 3.3f. The Medium Adaptor(MA) will provide a medium access mechanism to ensure that all terminals get access to the network. These MA boxes are fully topology dependent and their functions will not be standardised by CCITT. In this example, the W interface may be topology dependent and non-standardised. Other implementations may have a solution in which W is identical to S_B.

11.0 Reference points

The basic characteristics of the S_B and T_B are described for 155.520 Mbit/s. This 155.520 Mbit/s is the physical bit rate provided at both T_B and S_B. With respect to the structuring of this physical bit/cell stream, CCITT has left 2 options possible and they are given as follows:-

(i) Cell based, and
(ii) Synchronous Digital Hierarchy.

The T_B and S_B interfaces are physical point-to-point, meaning that receiver and transmitter are always paired (i.e. one receiver receives information from one transmitter). On the higher layers a logical point-to-multipoint functionality may be supported.

12.0 Functional grouping

CCITT has not yet achieved a full description of all functional groupings. However, some major guidelines can already be described.

The B-NT1 mainly performs low layer functions such as line transmission termination, transmission interface handling and OAM functions.

The B-NT2 functional group performs adaptation functions such as cell delineation, concentration, buffering, multiplexing/de multiplexing, resource allocation, usage parameter control, signalling adaptation layer functions, signalling protocol handling, switching of local connections and OAM functions. B-TN2 implementations can be connected, or physically distributed. It may even consist of physical connections only.

The B-TE1 terminates the user interface (S_B or T_B), and performs the termination of all end protocols from the low layers up to the higher layers.

13.0 Lower Layer Functionalities:

The physical layer offers capability to carry ATM cells .The ATM layer offers a cell transfer capability common to all services .The AAL offers service dependent functioons on top of the ATM layer

Physical Layer Functionalities:

The physical layer offers the transmission payload which carries the ATM cell stream .At the transmitting side ,the ATM cell stream is mapped into the transmission payload .Cell delineation function and header error control function are needed at physical medium -level connection endpoints.In order to adapt the rate of valid cells to the capacity of the transmission payload ,idle cells are inserted and extracted at the endpoint's ,the following figure illustrates the ATM cell mapping into the SDH payload at a 155.56 Mb/s interface .The same ATM cell mapping will be applied at UNI.

Physical layer

The physical layer of the BISDN is further composed of 2 sublayers:-

(i) The Physical Medium (PM), and
(ii) The Transmission Convergence (TC).

The PM supports the pure medium dependent bit functions whereas the TC sublayer converts the ATM cell stream into bits to be transported over the physical medium.

Physical medium sublayer

This sublayer is responsible for the correct transmission and reception of bits on the appropriate physical medium. The functions to be performed are shown in the following table:-

Convergence	CS	AAL
Segmentation and assembly	SAR	
Generic flow control Cell VPI/VCI translation Cell multiplex and demultiplex		ATM
Cell rate decoupling HEC header sequence generation/verification Cell delineation Transmission frame adaptation Transmission frame generation/recovery	TC	PHY
Bit timing Physical medium	PM	

CS : Convergence sublayer

SAR : Segmentation and Reassembly

TC : Transmission Convergence

PM : Physical Medium

At the very lowest level this function is really medium dependent (optical, electrical, ...) and is called physical medium. In addition, this sublayer must guarantee a proper bit timing reconstruction at the receiver. Therefore, the transmitting peer entity will be responsible for the insertion of the required bit timing information and line coding.

Transmission convergence sublayer

In this sublayer, bits are already recognised, as they come from the PM sublayer. This sublayer performs basically 5 functions as show in the above table.

The first function after the bit reconstruction is the adaptation to the transmission system used. The cells are fit within the transmission system according to a standardised mapping.

This sublayer is also responsible for the generation of the HEC (Header Error Check) syndrome of each cell at the transmitter, and its verification at the receiver. For a start, this permits the recognition of the cell boundary, i.e. proper cell delineation at the receiver. The mechanism to perform cell delineation is based on the HEC algorithm. This means that if a correct HEC syndrome is recognised for a number of consecutive cells, it is assumed that the correct cell boundary is found. To avoid malicious or erroneous cell delineation on user information, the information field of each cell is scrambled at the transmitting side and descrambled at the receiving side. This ensures that the probability of finding a correct HEC syndrome in the information field of an ATM cell is very low.

Once the cell delineation has been found, an adaptive mechanism uses the HEC syndrome for correction or detection of cell header errors, depending on the situation. Isolated single bit errors are corrected, but as soon as multiple consecutive cells show header errors, correction is given up for higher precision detection and elimination of cells with errors, to avoid slipping through of cells with undetected multiple header errors during the period of bit error bursts. Such errors may not be detected by the correction algorithm.

Finally, this sublayer must ensure insertion and suppression of unassigned cells to adapt the useful rate to the available payload of the transmission system. This function is called cell rate uncoupling.

In addition, Operations and Maintenance information must be exchanged with the Management plane.

14.0 Synchronous digital hierarchy based interface

Physical medium characteristics

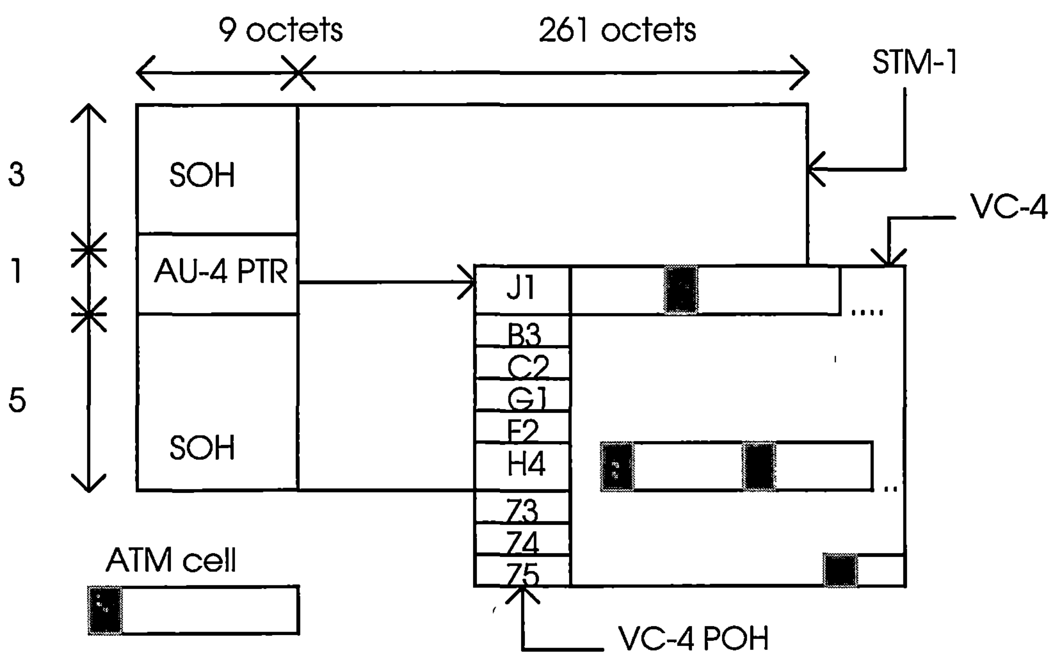
The preferred physical medium is optical fibre, other transmission media such as coaxial cable are also considered.

At the T_B reference point, a bit rate of 155.520 Mbit/s is selected in both directions. Both electrical and optical interfaces are possible, depending on the requirements in terms of distance, reliability and cost. Maximum range depends on the specific attenuation of the transmission medium used, and lies in the range of 100-200 meters. The optical solution should cover up at least 800 meters, and possibly up to 2000 meters.

Transmission convergence characteristics.

The bit rate available for information cells, signalling cells and OAM cells excluding physical layer frame structure octets is 146.760 Mbit/s on a 155.520 Mbit/s transmission system, and 599.040 Mbit/s on a 622.080 Mbit/s transmission system.

In this option, ATM cells are carried in an SDH frame as shown in the following diagram, for an STM-1 signal.



The SOH (Section Overhead) and the POH (Path Overhead) comply fully with the SDH. The Transmission convergence Sublayer performs frame generation and recovery, scrambling and descrambling for improving clock extraction, multiplexing of containers, frequency justification of individual containers to the transmission frequency by pointer processing, path signal identification, OAM and 125 μs clock recovery. Especially for ATM cell payloads, cell delineation by using the HEC, cell scrambling and descrambling, and HEC generation and checking are added.

The OAM implementation is in accordance with the general SDH specifications. This OAM allows frame alignment, error monitoring, error reporting and so on. Transmission performance is monitored and reported per section and per path, using the SDH overheads octets. Only the contents of the C2 byte is particular for an SDH frame transporting ATM cells: it contains an indication that the payload consists of ATM cells, an ATM payload construction indication.

15.0 Cell based interface

Physical medium characteristics

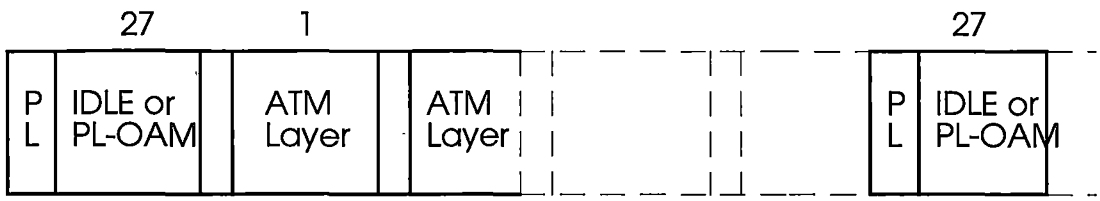
The physical medium characteristics of a cell based are identical to the ones for an SDH based interface.

Transmission convergence characteristics

In this option, cells are transported continuously, without any regular framing related to a time frame. Since no external clock is available at the receiver, this clock may either be derived from the signal received from the local node, or be provided by the clock of the customer equipment.

The Transmission Convergence Sublayer performs cells delineation, HEC generation and checking, cell rate adaptation between the ATM layer and physical layer, and OAM functions. The bit rate available for user information cells, signalling cells and OAM cells is 149.760 Mbit/s on a 155.520 Mbit/s transmission system, and 599.040 Mbit/s on a 622.080 Mbit/s transmission system. These values are identical to the payload of the respective SDH frames.

In order not to exceed the allowed maximum payloads on an interface with a nominally higher physical bit rate, the physical layer carries special Physical Layer (PL) cells, which are neither passed on to, nor received from, the ATM layer. They are generated and interpreted in the physical cell based layer. The maximum spacing between successive PL cells is 26 ATM layer cells. They can either be IDLE cells, or Physical Layer OAM cells (PL-OAM). PL cells are identified by a pre-defined header. IDLE cells merely perform cell rate adaptation, PL-OAM cells convey OAM information concerning the physical layer itself. The following table shows the pre-assigned header values for PL cell types.



PL-OAM cells carry regenerator level (F1) and transmission path (F3) level information. The need to be inserted in the ATM layer cell flow on a recurrent basis. Minimum periodicity for each type is one in 513 cells. The digital section level flow (F2) is not used, as its functions are supported by F3 flow, by lack of a transmission frame on the cell based UNI.

Functions to be supported are the monitoring of the performance, and detection and reporting of transmission errors. Performance checking includes counting, and calculating an error code over the ATM layer and IDLE cells between two subsequent PL-OAM cells. The results are conveyed in the information field of the PL-OAM cell, together with maintenance signalling, and a CRC on the Pl-OAM cell information field itself.

16.0 Plesiochronous digital hierarchy based interface

Physical medium characteristics

Carrying ATM cells in PDH frames has the advantage of using on the existing transmission network infrastructure, instead of having to rely on the deployment of new transmission equipment.

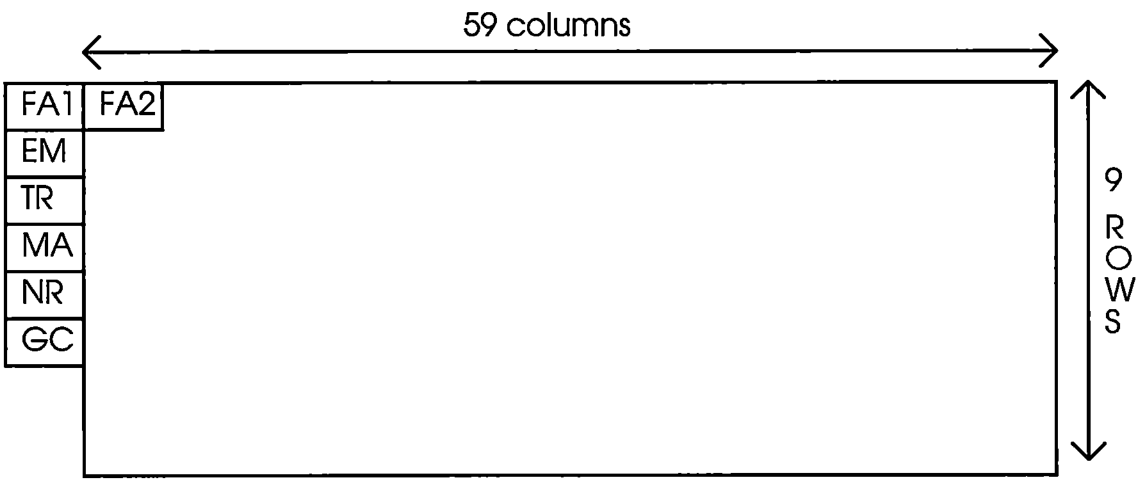
Transmission convergence characteristics

Several ways for mapping ATM cells in PDH frames of different bit rates have been proposed. Some early proposals implemented a kind of cell based transmission on top of the existing PDH frame, using PL-OAM cells for maintenance.

These methods have since been left, in favour of a more SDH-like approach, in which maintenance, performance monitoring and reporting are based on the use of special octets which are added to the frame. The remaining payload of the PDH frame is filled with ATM cells, which are octet aligned to the octet structure of the PDH frame payload area. The ATM cells are delineated by using the HEC, and their payload is scrambled to avoid false frame and cell synchronisation.

As an example the figure below shows the frame for a 34.368 Mbit/s PDH interface. The following Path Overhead functions are defined:-

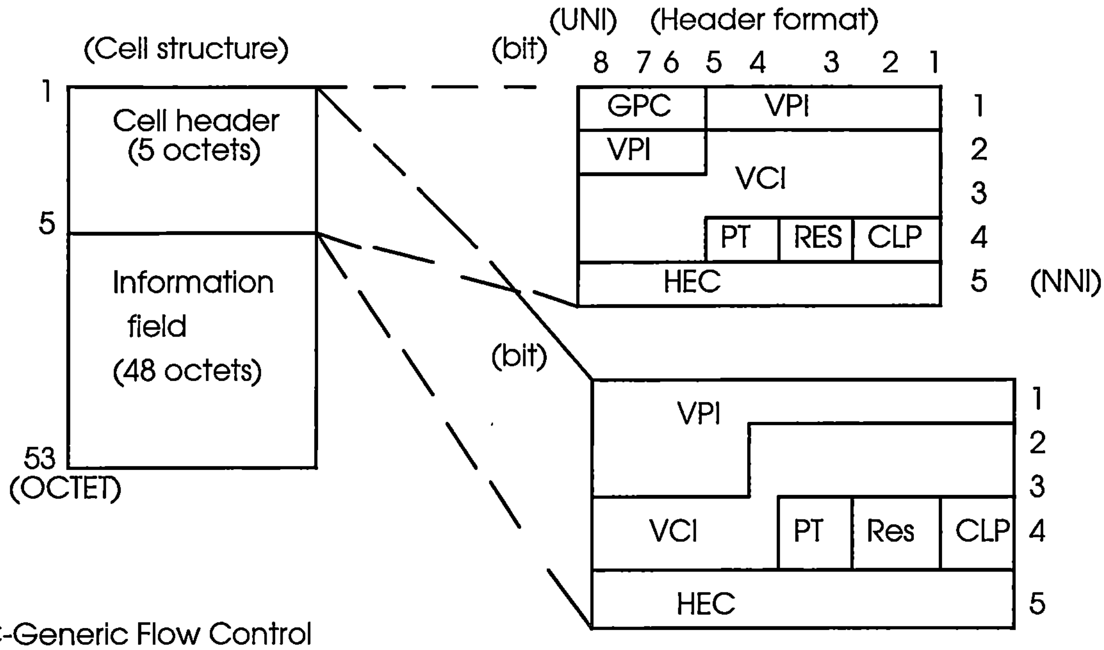
FA : Frame Alignment
EM : Bit Interleaved Parity (BIP-8_
TR : Trail Trace
MA : Far End Receive Failure (FERF), Far End Block Error (FEBE), Payload Type.
NR : Network Operator byte
GC : General purpose Communications channel (e.g. data or voice for maintenance)



The method makes use of a Physical Layer Convergence Protocol (PLCP) which is a subset of the one used in DQDB(Distributed Queue Dual Bus) networks. A PCLP frame is used, which contains an integer number of ATM cells, and which has no direct relationship with the DS3 frame in which it is carried. In the PLCP frame, each ATM cell is preceded by PLCP framing octets

17.0 ATM Layer Functionalities:

The ATM layer functions are those provided by the ATM cell header .The cell format at UNI and NNI is shown below :



FC-Generic Flow Control

PI-Virtual Path Identifier

CI-Virtual Channel Identifier

T-Payload Type

HEC-Header Error Control

CLP-Cell Loss Priority

ATM cell header structure

Res-Reserved bits

In the cell header ,the connection identification field ,composed of VPI and VCI, is essential for switching and multiplexing .As the connection identification field, 24 bits are allocated at UNI and 28 bits are allocated at NNI.

In addition to these connection identification functions, some additional fields are included to improve performance an eight -bit error control(HEC), a two -bit payload type(PT)field,and a one-bit cell loss priority(CLP) field.At UNI ,a four bit generic flow control(GFC) field is defined.

The ATM layer specifications in terms of the ATM cell header are standardized except for the coding of GFC and PT. The GFC requirements include guarantee of capacity ,fairness,efficiency ,and directionality of operation.The best criteria for GFC protocol evaluation was established ,which include test configuration and test load

Another outstanding feature in the ATM layer is whether VCI/VPI assignment should be directional or unidirectional .Comparison of these two schemes was performed under various VC/VP configurations but no agreement has not been reached .

3AAL Functionality's : AAL is a layer between the ATM layer and the service user layer.The functions performed in the AAL depend upon the higher layer requirements.The AAL supports multiple protocols to fit the needs of different AAL service users .Examples of services provided by the AAL include handling quantization effect due to cell information field size ,transmission errors, lost and misdelivered cells,flow control and timing control

The AAL services are classified based on a timing relation between source and destination ,bit rate(constant and variable) ,and connection mode.As a result, four AAL classes are distinguished as shown below:

To provide four AAL service classes (class A ,B,C, and D),four AAL protocol types (Type1,2,3 and 4) are prepared .To simplify the realzation of the AAL functions and to avoid protocol proliferation ,maximum commanality between the protocol. elements of the AAL protocols is preferred.

AAL protocol specification :

	CLASS A	CLASS B	CLASS C	CLASS D
Timing between Source and Destination	Related		Not Related	
Bit Rate	Constant	Variable		
Connection Mode	Connection Oriented		Connectionless	
Example of Services	Circuit Emulation Constant Bit Rate video	Variable Bit Rate Video and Audio	Connectionless Data Transfer	

ATM adaptation layer services

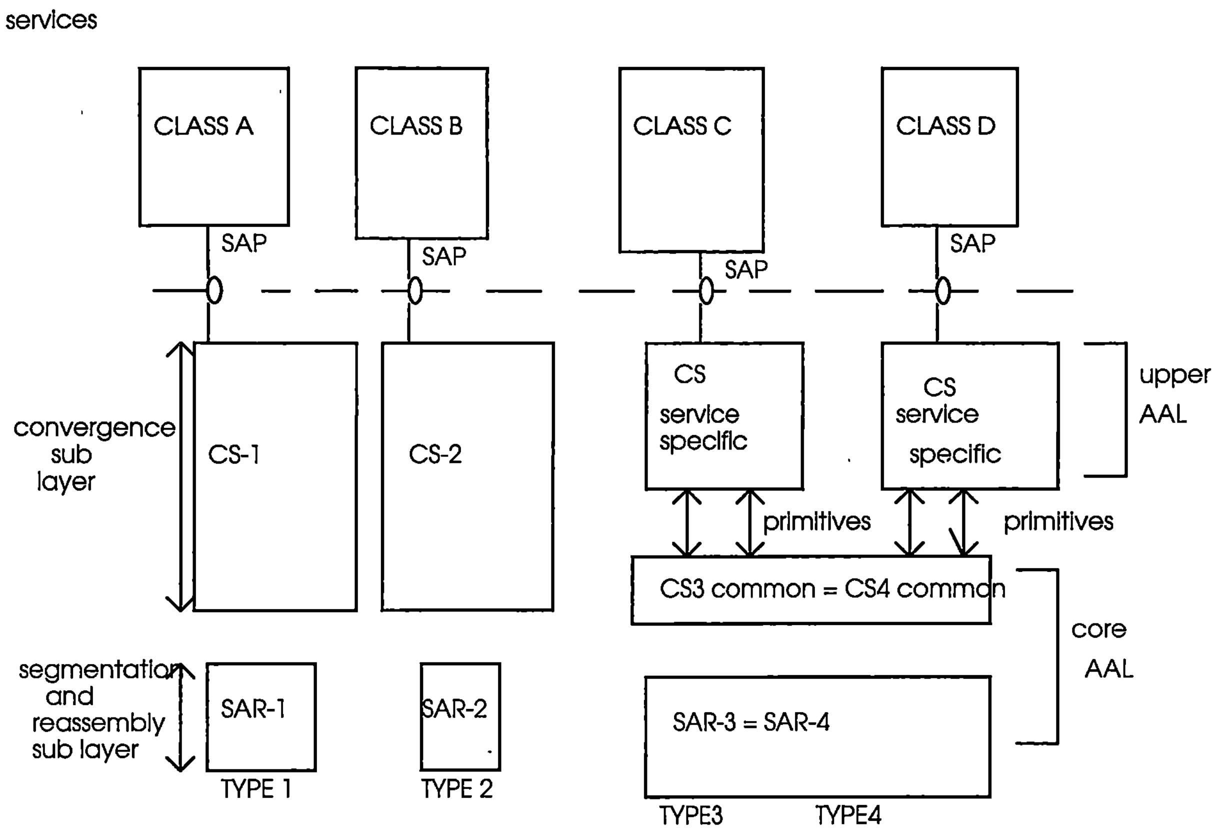
CCITT recommendation specifies these four types of AAL protocols .The current status is described below .

AAL TYPE1: AAL TYPE 1 provides constant bit rate services such as traditional voice transport .Functions which are provided by TYPE1 include segmentation and reassembly of user information ,handling of lost and misinserted cells ,handling of cell delay variation ,and source clock frequency recovery at the reciever .

AAL TYPE2: AAL TYPE 2 transports variable bit rate video and audio information ,keeping the timing relation between source and destination .It provides functions similar to those of AAL type 1.

AAL TYPE 3: AAL TYPE 3 supports connection oriented data service and signalling .This functions include segmentation and reassembly of variable length user data and error handling

AAL TYPE4: AAL TYPE 4 supports connectionless data services .The same functions as AAL TYPE 3 are provided ,in addition to the multiplexing - demultiplexing function



AAL Protocol Architecture

The functions of AAL are organized in two sublayers :1.segmentation and reassembly (SAR) sublayer and 2.CONVERGENCE sublayer (CS) service access points (SAP'S) are not defined between sublayers .Different combinations of SAR and CS provide different SAP'S to the layer above the AAL .The key issues relevant to the AAL protocol architecture are:

1.Relationship between service class and protocol type,
2.Independence between SAR and CS
3.Method of supporting multiple services by one service class

There are basically two alternatives in terms of protocol types supporting class C and D. The main difference between SAR type3 and SAR TYPE4 is the necessity of multiplexing functions in the AAL .One alternative makes use of AAL TYPE3 for connection oriented and type4 for connection's services .A new AAL protocol architecture has been established to resolve the differences.

The agreed framework of the AAL protocol architecture for the service classes C and D is shown below .Service primitives are defined inside the CS which separate the core AAL and the upper AAL .The core AAL protocol is common to both service class C and D,where as upper AAL is service dependent

The service provided by the core AAL is equivalent to the core service in the frame mode bearer service(FMBS) Most of the specifications of the SAR sub layer and the CS in core AAL ,which are agreed to.The common CS protocol is based on the specifications of IEEE 802.6

CHAPTER3:

RESULTS SECTION1

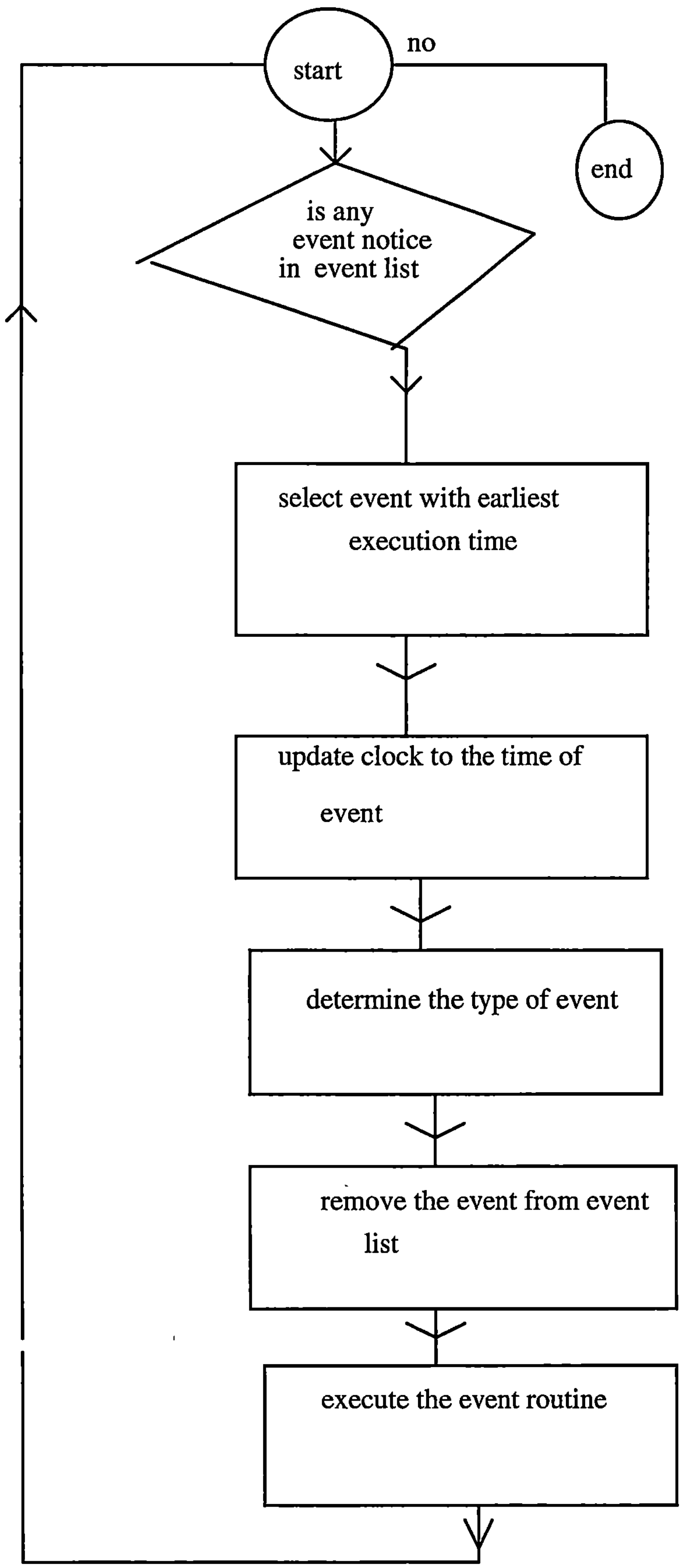
1.0 AN OVERVIEW OF SIMULATION RESULTS:

Simulation programs are written in simscript 2.5 version .All the programs are written using EVENTS which are activated in specific instants of time .A separate routine for arrival process and departure process are written as events and whenever these routines are called all the steps that directed to are done then each routine returns the control to timing routine which determines the next event that is to be performed and is picked accordingly and performed

Essentially the program consists of PREAMBLE where the global variables are declared .Next is MAIN in which the first event to called is written using a schedule statement .The total amount of time the program is to be run is also specified using the schedule statement .In all the programs written to perform simulation will have an arrival statement ,a stop statement asking the program to stop at specified instant .

Next the arrival event routine calls itself at inter arrival times of the packets.Also the event departure is scheduled in service time whenever the packet has to be serviced. In the departure routine again the channel(queue) is tested whether empty or not ,if not empty the departure routine calls itself at regular intervals (service times) .

The following is the **timing routine** for **simscript**:



The following programs are written and the results investigated

M/M/1/K with markov(random) arrivals ,markov service times ,one server and finite buffer space of K . ***M/D/1/K*** queue which is the queue with minimum markov arrivals ,constant (deterministic) service times ,one server and finite buffer .

The switched Poisson arrivals queue(SPP/D/1/K) which is constant service time ,one server and finite buffer size.Next interrupted Poisson arrival process queue where each source is active for certain amount of time and inactive for certain time ,the active and the inactive times are modelled as exponentially distributed with mean values

2.0 *M/M/1/K* QUEUE ANALYSIS

The following are the characteristics considered.

1. Buffer size is limited to 4 .
2. The arrival times are generated using inbuilt exponential random generator,similarly the service time is generated using the inbuilt exponential random generator
3. Starting from utilisation 0.4 to .9 are considered for the analysis
4. Burst length analysis that is determining the probability that a single cell is lost and the consequent cells lost in a row . That is probability of burst lengths or row lengths are determined from simulation and analytical formulae
5. Histogram distribution as a function of packets and their wait times are also plotted to gain insight into the delay suffered by the packets

THEORETICAL ANALYSIS:

The important characteristic of exponential distribution is its memory less nature .

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

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

The arrivals and the services are independent of each other .For stastically independent random arrivals ,the joint probability density function is their product[Hayden] ,that is

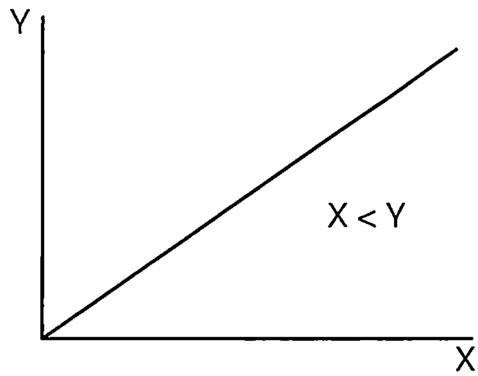
$p_{xy} = a(x).s(y)$

where s(y) is the service time exponential probability density function obtained by differentiating the c.d.f of the service time distribution and is given by
$s(y) = \mu e^{-\mu y}$

for our case y can be taken to be zero when the cell is lost
Now the probability of one service in the interval in (0,A) and m arrivals in the interval (0,B) is:

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

To calculate the probability of buffer overflow we want M(arrivals) and N(services) such that
$0 \le X \le Y$ and $0 \le X \le Y$
This defines an infinite triangular area on the M,N plane as shown in the following figure



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

Remembering always that a single cell is lost ,the above probability is identical that the burst of lost cells is m+1 long

$$\text{pr(burst of length m+1)} = \int_0^\infty \int_0^y \frac{\lambda(\lambda x)^{m-1} e^{-\lambda x} \mu e^{-\mu y}}{(m-1)!} dx dy$$

The above integral is calculated using the integral tables(Gradshteyn) and can be shown to be:

$$\text{pr(burst of length m+ 1)} = \left(\frac{\rho}{\rho+1}\right)^m \quad \text{---- 1}$$

where ρ is the server utilisation equal to $\dfrac{\lambda}{\mu}$

The above formulae can be used to derive exactly the burst length m from the theory of conditional probabilities[Haykin] Bayes rule
pr(burst of length m) = pr(m arrivals before one service)/pr(1 arrival before 1 service)
then using the above equation (1)we have

$$\text{pr(burst of length m)} = \left(\frac{\rho}{\rho+1}\right)^{m} \Big/ \left(\frac{\rho}{\rho+1}\right)$$

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

Derivation Of average cell loss run for ***M/M/1/K*** QUEUE:

Important characteristics in studying the cell loss are the mean and the variance of the cells lost in a burst

The average cell loss length is given by

$$\bar{x} = \sum_{m=1}^{\infty} m\, p_r(mcellslost / 1celllost)$$

Now the probability that the cell length is exactly m cells is:

pr(number of cells lost due to buffer overflow is exactly m)

$$= \text{pr(length} \geq m) - p_r(length \geq m+1)$$

$$= \left(\frac{\rho}{1+\rho}\right)^{m-1} - \left(\frac{\rho}{\rho+1}\right)^{m}$$

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

Now again consider the equation $\sum_{m=1}^{\infty} \frac{m\rho^{m-1}}{(1+m)^{m}} = 1+\rho$

Now again consider L.H.S $= \frac{1}{\rho} \sum_{M=1}^{\infty} m\left(\frac{\rho}{1+\rho}\right)^{m}$

$$= \frac{1}{\rho}\left(\frac{\rho}{1+\rho}\right)\sum_{m=1}^{\infty} m\left(\frac{\rho}{1+\rho}\right)^{m-1}$$

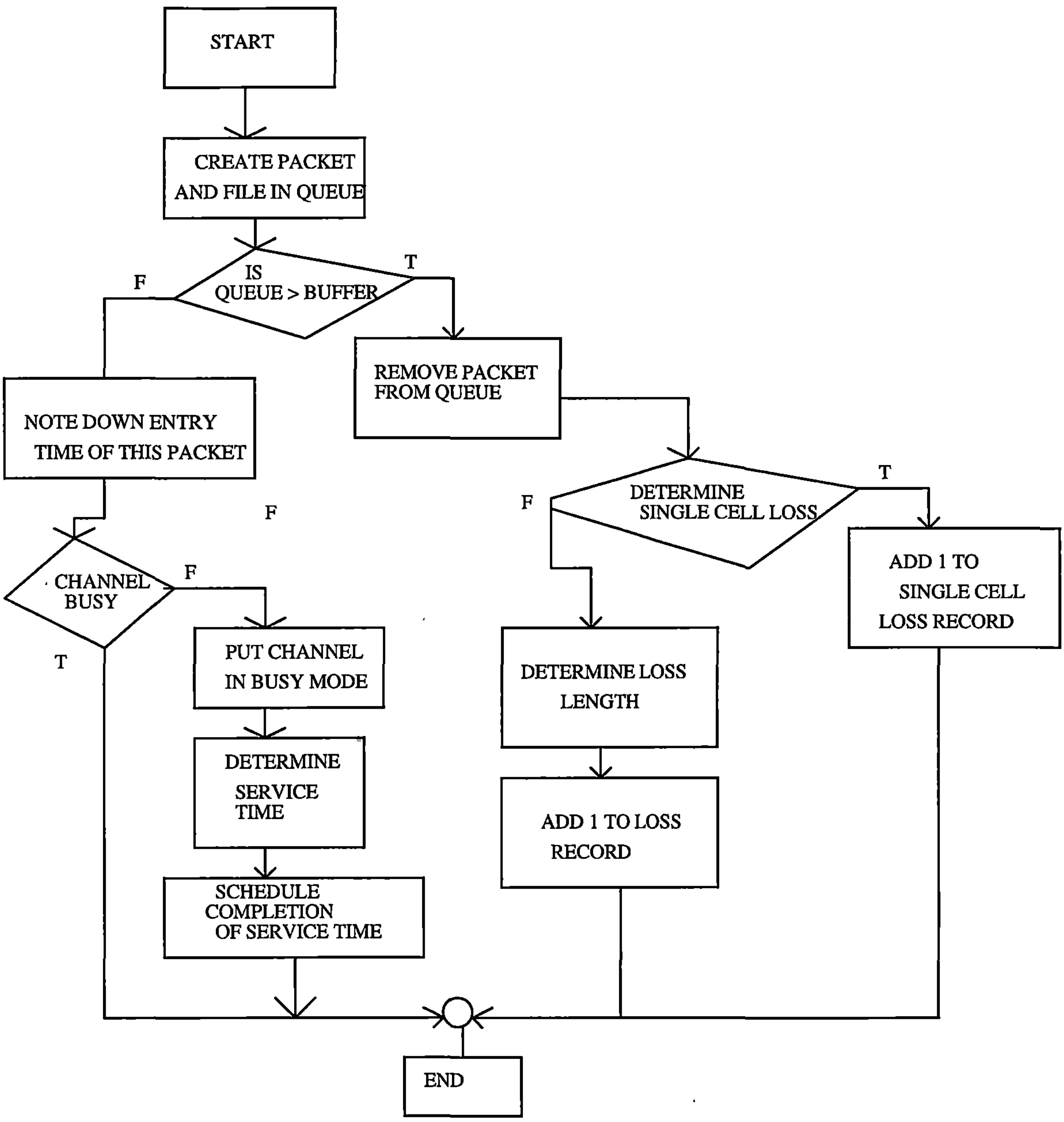
$$= \left(\frac{1}{1+\rho}\right)\frac{1}{\left(1-\left(\frac{\rho}{1+\rho}\right)\right)^{2}}$$

$$= \left(\frac{\rho}{1+\rho}\right)(1+\rho)^{2}$$

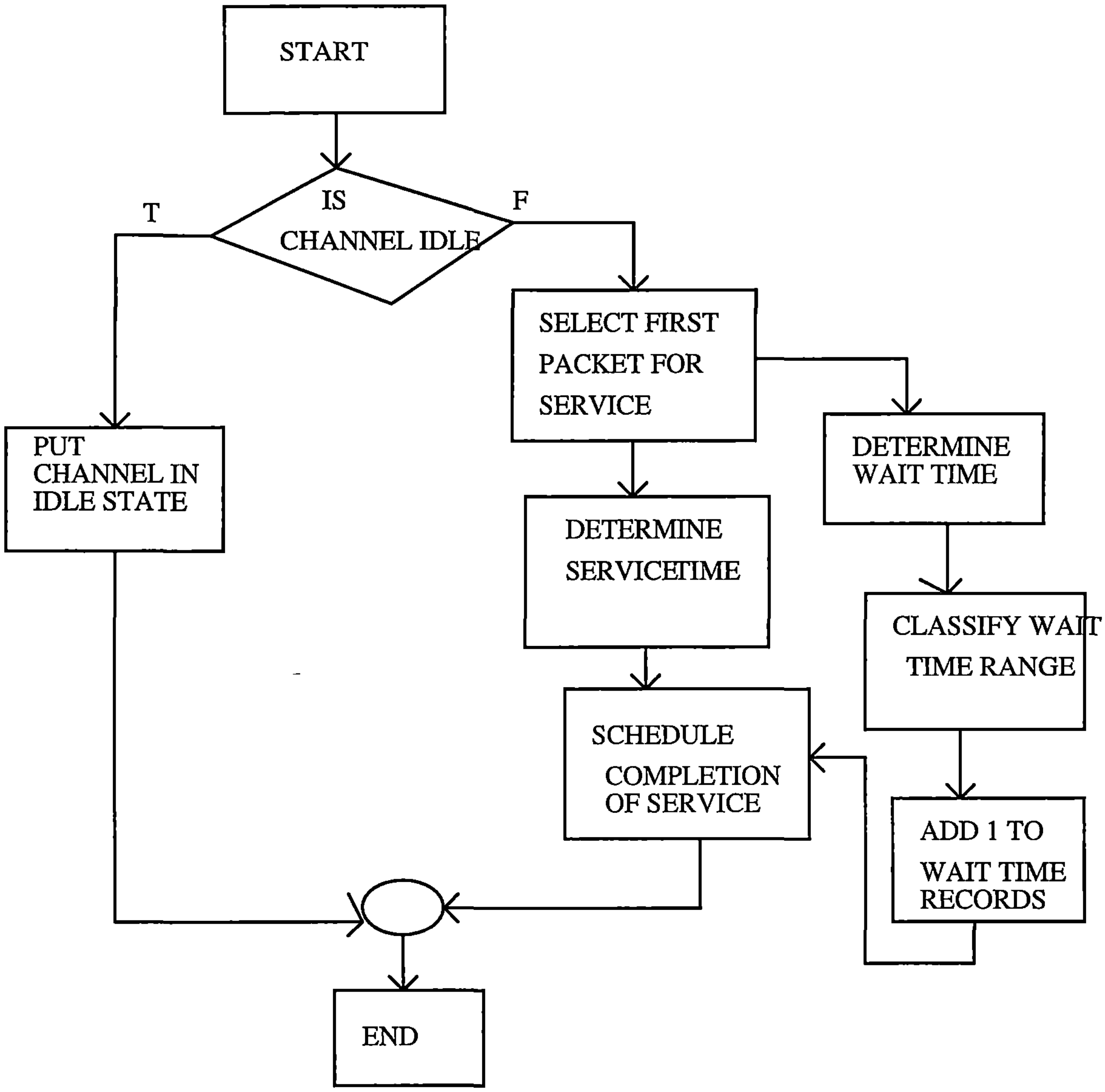
$$= 1+\rho \quad \text{---------} \quad (3)$$

Now the equation (2) is analytical formulae for calculating the burst length probability ,to test the agreement between simulation and analytical result a program is written which collects the queue stastics from which the various burst length probabilities are calculated ,the following are the flow charts for the arrival routine and event departure routine

EVENT ARRIVAL ROUTINE



EVENT DEPARTURE



3.0 PROGRAM SOURCE CODE *M/M/1/K* QUEUE

```
  preamble   " declaration of all the global variables that are used throughout
"the programm
  the system owns the queue
  event notices include arrival,departure, and stop.sim
 to the queue
    define arr, svc ,wait.tim as real variables
     define times,channel,m  as integer variables
        define idle to mean 0
     define busy to mean 1
   accumulate aniq as the average and max.length as the maxi
      tally mat as the mean of arr
   tally mst as the mean of svc
  tally mwt as the mean of wait.tim
  define destroy.customer as integer variable
  define n.departures as integer variable
  define n.arrivals as integer variable
 accumulate xbusy as the average   of channel
  define n,times1,i,count as integer variable
  define a,k,lamda as real variables
  define milliseconds to mean units
 define list,list1 as 1-dimensional integer arrays
   tally max.wait as the maximum of wait.tim
 end
  main
  read S       " buffer size
  read T      " service time
  read A       " arrival time
   reserve list(*)  as 32  "list 1-dimensional array to collect burst cell lengths
     reserve list1(*) as 32   "list1 1-dimensional array to collect wait times and
"plot histogram
    let count = 1
   let n.arrivals = 0
  let n.departures = 0
        let arr = exponential.f(A.,1)
   schedule an arrival in  arr milliseconds
 let destroy.customer = 0
  start simulation
   end
  event arrival
 let n.arrivals = n.arrivals + 1 " counter to keep track of the number of arrivals
    let arr  = exponential.f (A,1)
 schedule an arrival in arr milliseconds
   create a  packet
```

```
    let entry.time(packet) = time.v
    file this packet in the queue
     if n.queue   >  S   "condition  to test whether queue of packets is greater
"than buffer
        add 1 to destroy.packet " counter to accumulate the loss of packets due
to buffer overflow
        remove this  packet from the queue
    destroy this packet
       add 1 to count
       else
          if n.queue <=  S  " loop condition to determine the burst cell length
in "other words finding out whether two cells are lost in a row ,3 cells lost in
row   "accordingly  and adding 1 to list(2) or list(3) so on
       if count > 0
         if count <= 10
         add 1 to list(count)
           else
        add 1 to list(11)
            endif
             endif
             let count = 0
             endif
              endif
     if channel = idle
    let channel = busy
    remove first  packet from the  queue
    let wait.tim = 0
     if wait.tim = 0
     add 1 to list1(22)
     always
     let svc =   exponential.f(t,1)
   schedule a departure in svc milliseconds
destroy this packet
    always
   return
  end
  event departure
let n.departures = n.departures + 1
  if n.departures = 5000000
   print 1 line with n.queue thus
   n.queue is ***********
   schedule a stop.sim now
   always
  if queue is empty
  let channel = idle
```

```
 jump ahead
  else
  remove the first packet from the queue
  let wait.tim = time.v - entry.time(packet)
     if wait.tim > 0 and wait.tim <= .06
      add 1 to  list1(1)
       always
         if wait.tim > .06
        for i = 2 to 21
destroy this packet
    always
   return
 end
 event departure
let n.departures = n.departures + 1
  if n.departures = 5000000          " a total of 5 million cells are transmitted
"through the channel  and then programm is asked to stop and display queue
"stastics
   print 1 line with n.queue thus
   n.queue is ***********
   schedule a stop.sim now
   always
  if queue is empty
  let channel = idle
 jump ahead
  else
  remove the first packet from the queue
  let wait.tim = time.v - entry.time(packet)
     if wait.tim > 0 and wait.tim <= ..325  " after classifying the differnt  wait
"times according to the range (5 % increments) and then accordingly
"incremmenting the corresponding  total
      add 1 to  list1(1)
       always
         if wait.tim > .325
        for i = 2 to 21
 for i = 2 to 21
          do
         let k = k + .325
        if wait.tim > .325 and wait.tim <= (.325) * (i)
          add 1 to list1(i)
        always
      loop
     let k = 0
       always
   let svc = exponential.f(t,1) " service time exponentially distributed
```

```
schedule a departure in svc milliseconds
  destroy this packet
  here
 return
end
event stop.sim
  print 5 lines with list1(1),list1(2),list1(3),list1(4),
list1(5),list1(6),list1(7),list1(8),list1(9),list1(10),list1(1
list1(12),list1(13),list1(14),list1(15),list1(16),list1(17),
list1(18),list1(19),list1(20),list1(21),list1(22) thus
   stop
******** ********* ********* ********* ******** *******
******** ********* ******** ********* ******* ********
******* ********** ******** ******** ******* *******
*******    *********    **********    ********
 print 2 lines with list(1),list(2),list(3),list(4),list(5
 list(6),list(7),list(8),list(9),list(10),list(11),list(12),
  list(13),list(14),list(15),list(16) thus
***** ***** ******* **** ***** ****** ****** ***** ******
***** ******** ****** ******* ******* ****** ******


  print 13 lines with
   aniq,
  xbusy,
 mat,
 mst,
mwt,n.arrivals,n.departures, destroy.customer,arr,and max.le
mwt,n.arrivals,n.departures, destroy.customer,arr,and max.le
times1,count,max.wait thus
   average queue length **.******* minutes
    utilisation of server ***.******
    mean arrival time of packet is ****.*****************
     mean service time of packet is **.*******
      average wait time is **.********
   number of arrivals *****************
    no of departures ***************
   destroy customer ***************************
    b = ***.********
  max.length = *********.***********
times1 = ***********.***
 count = **********
  maximum wait = ************.**********
   add 1 to times
  if times = 1
 stop
```

```
return
end
```

Now the programm output file is copied here and the queue stastics that are printed out are explained
Now the queue stastics for M/M/1 queue with buffer size limited to 4
is collected the following program results

```
 n.queue is        0
  stop
 693502   534703   372346    233213  133327  70993
  35575     17233    8143      3589    1634      717
   271       129       45       22       6      4
    1           2               0      894545
90217 39602  17824 7910  3516  1541   684  316   134
   66     63    0     0     0    0    0
   average queue length  1.1360058 minutes
    utilisation of server   .728245
    mean arrival time of packet is    .37530093954264343
     mean service time of packet is  .2998504
       average wait time is  .46774325
   number of arrivals         3291309
    no of  departures       3000000
   destroy .packets           291309
    b =    .59143025
  max.length =        5.00000000000
times1 =         0.
 count =        0
  maximum wait =          6.4948976394
~
~
"bar1" 21 lines, 905 characters
```

The above output file was obtained for utilisation of 81% but as is evident from the output stastics results the utilisation is only 72% it is obvious because we have limited the buffer size to 4 and the cells are lost again this can be verified since the no of departures are 3 million (transmitted packets) and the destroyed two hundred and ninety one thousand and three hundred nine which is again approximately 8%,maximum length of the queue is 5 ,since from 5th arrival a packet is made to loose.

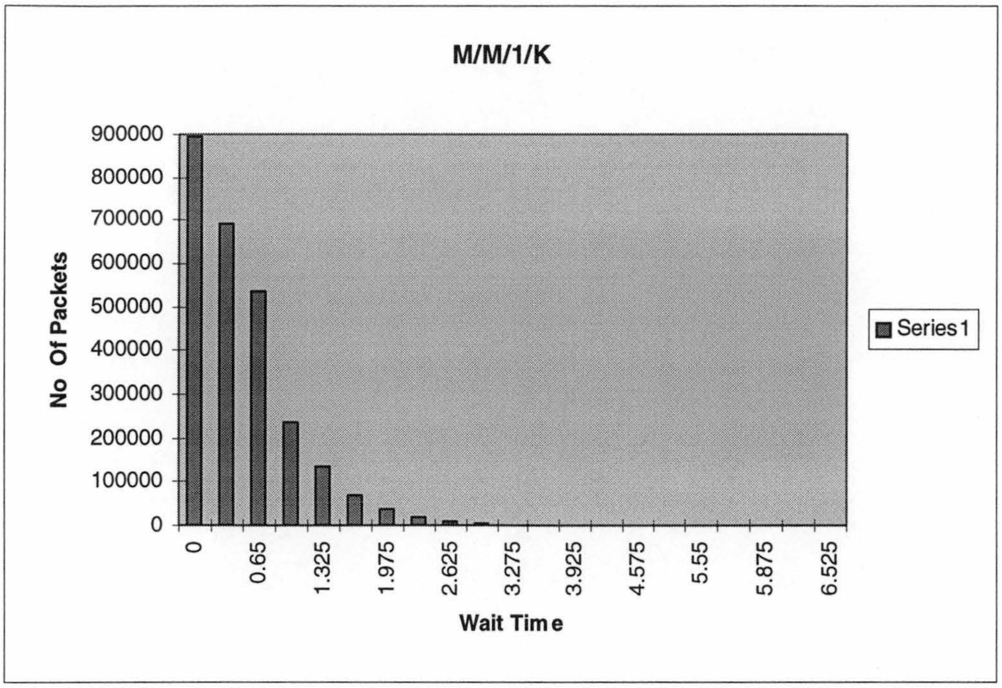
Now the maximum wait time is 6.49 milliseconds(all times are in terms of milliseconds) which can be approximated to 6.5 ,now 5% of 6.5 is given by = 6.5/20 = .325

In the above output file the first row immediately after 'stop' (identification purpose only) gives how many packets had 0 to .325,.325 to .65,.65 to 0.975 and so on upto 6.5 ,thus with these stastics the histogram for wait time vs no of packets is plotted and is given as follows
utilisation = 80

TABLE1
WAIT TIME v/s NUMBER OF PACKETS

0	894545
0.325	693502
0.65	534703
0.975	233213
1.325	133327
1.65	70993
1.975	35575
2.3	17233
2.625	8143
2.95	3589
3.275	1634
3.6	717
3.925	271
4.25	129
4.575	45
4.9	22
5.55	6
5.55	4
5.875	1
6.2	2
6.525	0

HISTOGRAM 1 *M/M/1/K (.8 UTILISATION)*



Also the fourth row(from output file) gives (starting from 90217) gives the no of single cells lost,the next reading gives no of two cells lost in a row ,three cells lost in a row etc .now the burst length probability that it is of length m is :

pr(burst is lengthm)= prob(m arrivals before one service)/prob(1arrival before one service)

for example pr(burst length 2) is = 39602/90217=.438 (at utilisation = .8)

pr(burst length3) is = 17824/90217 = .197

pr(burst length 4) is = 7910/90217 = .087

pr(burst length 5) is = 3516/90217 = .038

similarly all the burst length probabilities are collected and also calculated analytically using equation (2) and are tabulated as follows also the graphs for burst probabilities vs utilisation for 0.4 and 0.8 are plotted

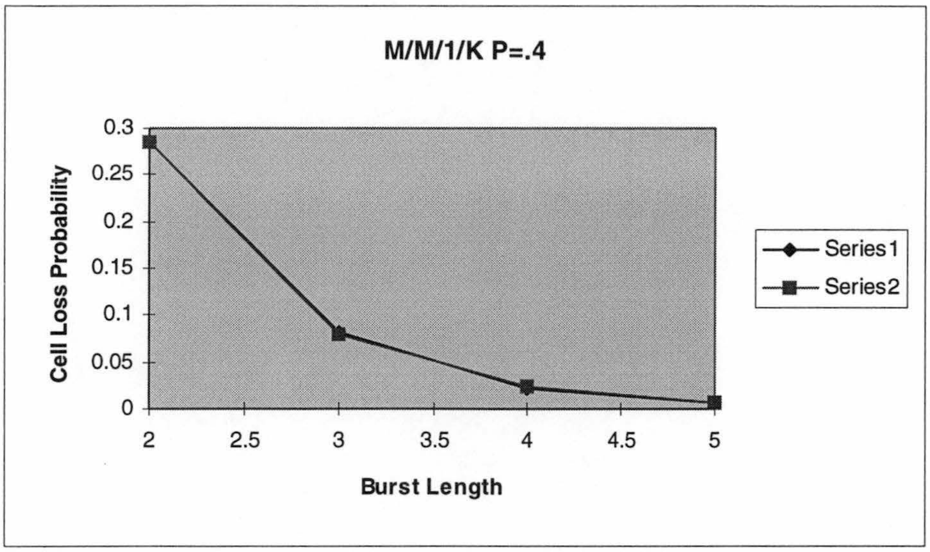
TABLE2. Burst Lengths Cell Loss *M/M/1/K*

TABLE 2

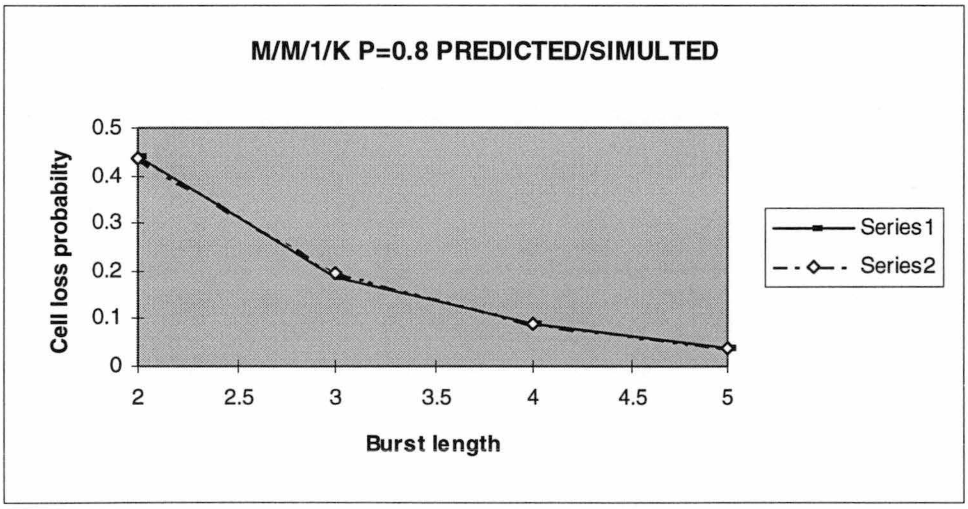
Burst length(simulated/predicted)

p	2		3		4		5	
0.4	0.284	0.283	0.081	0.083	0.023	0.024	0.006	0.007
0.5	0.333	0.333	0.116	0.112	0.036	0.037	0.012	0.012
0.6	0.37	0.374	0.14	0.142	0.053	0.054	0.02	0.02
0.7	0.43	0.414	0.17	0.18	0.075	0.07	0.027	0.029
0.8	0.443	0.444	0.185	0.198	0.09	0.088	0.038	0.037
0.9	0.474	0.474	0.223	0.224	0.106	0.106	0.054	0.051

Graph for simulated /predicted
GRAPH 1



GRAPH 2



From the above graphs it is clear that analytical and simulation results agree quite well and note that two graphs are running hand in hand.

Average Cell Loss

M/M/1/K

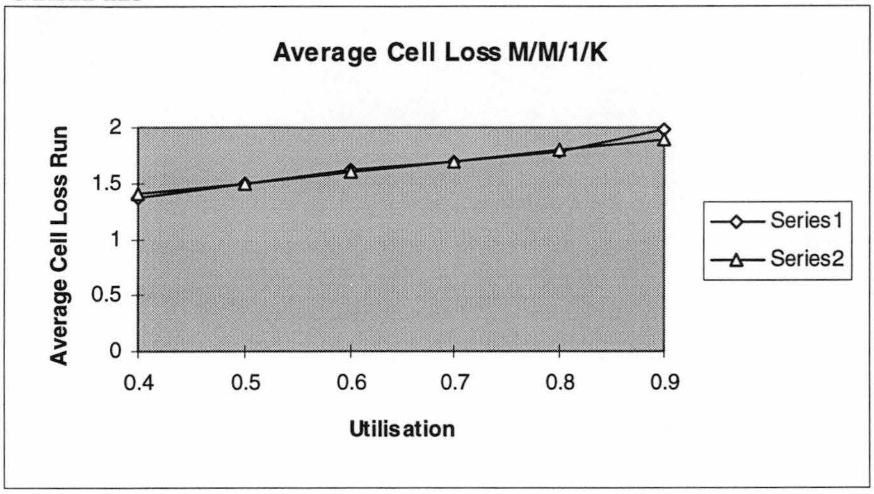
The theoretical values are calculated using the equation 3 and also simulated results are tabulated as follows

TABLE 3 Average Cell loss
Simulated and Predicted

0.4	1.378
0.5	1.489
0.6	1.613
0.7	1.7
0.8	1.788
0.9	1.99

M/M/1/K

GRAPH3



4.0 *M/D/1/K* Analysis

Theoretical Analysis Of Conditional Cell Loss for *M/M/D/1/K* QUEUE

The actual model that is suitable for modelling of a ATM multiplexer is *M/M/D/1/K* since all the cells are of 53 octets size (payload 48 and header 5) ,the time a multiplexer takes to process the cell is constant as the line speed is constant .

Now from the theory of the conditional probabilities analytical formulae for burst length is derived as well as the simulation is performed to match with it.Consider any M cell which is about to arrive ,and will be lost because the buffer is full .What is the probability that cell M+1 is lost as well ? From the theory of conditional probabilities we have
pr(A/B) = PR(A and B)/pr(B)
Considering the buffer overflow problem,then:
pr(cell M+ 1 is lost/cell M is lost) = pr(cell M+1 is lost and cell M is lost)/pr(cell M is lost) ------------ (4)

Now the R.H.S of 4 can be written in terms of service and arrival rates for the two probabilities
pr(cell M is lost) = pr(1 arrival before one service and queue is full) -------- (5)
pr(cell M+1 is lost and cell M is lost) = pr(2 arrivals before one service and queue is full) ----------- (6)
Substituting (5) and (6) into (4) we have
pr (cell M+1 is lost /cell M is lost) = pr(2 arrivals before one service and queue is full)/pr(1 arrival before 1 service and queue is full)
Since the arrival process are independent of each other we have
For two independent events A,B:
pr(A and B) = pr(A) .pr(B)

Therefore (6) becomes :
pr(cell M+1 is lost/cell M is lost) = pr(2 arrivals before 1 service)pr(queue is full)/pr(1 arrival before 1service) pr(queue is full)
=pr(2 arrivals before 1 service)/pr(1 arrival before 1 service) -----------(7)
(7) can be generalised to K arrivals before one service
pr(cell M+1,M+2,.........M+K are lost/cell M is lost)
= pr(K arrivals before one service and queue is full)/pr(1 arrival before 1 service and queue is full)
= pr(K arrivals before one service)/pr(1 arrival before one service) ---------(8)
Now equation 8 is applied to *M/M/D/1/K* queue
Since the arrivals occur at random and all the service times are costant ,the amount of time any arrival sees until the next service has a uniform distribution

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

$s(y) = \mu \quad \text{for } y \in (0, \frac{1}{\mu}) \quad \text{and zero elsewhere}$ ---------(9)

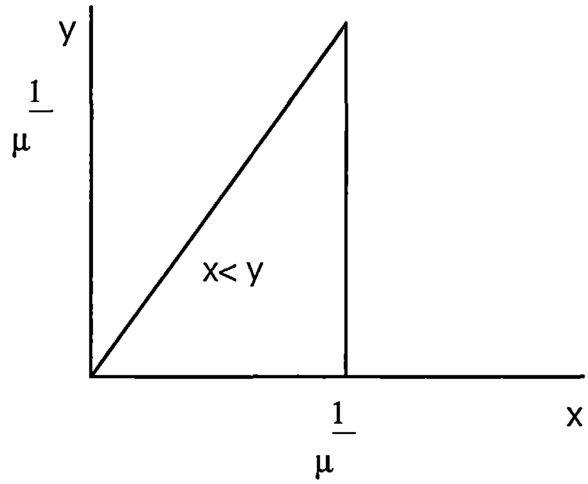
For the M/D/1/K queue arrivals are exponentially distributed.The probability density function of K arrivals in time x is(Kleinrock):

$$a(x) = \frac{\mu\lambda(\lambda x)^{k-1} e^{-\lambda x}}{(k-1)!} \quad \text{----------(10)}$$

Using the assumption of independence,the joint probability density function is the product of (9) and (10)

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

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



That is :

$$\text{pr(k arrivals before 1 service)} = \int_0^{\frac{1}{\mu}} \int_0^{y} \mu\lambda(\lambda x)^{k-1} e^{-\lambda x} dx dy \quad \text{---------(11)}$$

The above integral can be evaluated with the help of integral tables [Gradshteyn] and the 'Mathematica' package[Wolfram].

Finally using (11) and the gamma function ,the probability that the burst will be of length k is:

$$\text{pr(burst is K or more cells long)} = \frac{1 - \frac{1}{p} \sum_{i=0}^{k-1} \frac{\Gamma(i+1,0,\rho)}{i!}}{1 - \frac{1}{\rho}\Gamma(1,0,\rho)} \quad \text{--------- (12)}$$

Now equation (12) is the analytical formulae used to calculate the predicted burst length probability and also simulated results

5.0 Now the source code for ***M/M/D/1/K* program** is produced here

```
preamble
 the system owns the queue
    temporary entities
   every packet has a entry.time and
  may belong
  event notices include arrival,departure, and stop.sim
    temporary entities
   every packet has a entry.time and
  may belong to the queue
     define arr, svc ,wait.tim as real variables
      define times,channel,m  as integer variables
         define idle to mean 0
      define busy to mean 1
    accumulate aniq as the average and max.length as the maxi
       tally mat as the mean of arr
    tally mst as the mean of svc
   tally mwt as the mean of wait.tim
   preamble   " declaration of all the global variables that are used throughout
"the programm
   the system owns the queue
   event notices include arrival,departure, and stop.sim
 to the queue
     define arr, svc ,wait.tim as real variables
      define times,channel,m  as integer variables
         define idle to mean 0
      define busy to mean 1
    accumulate aniq as the average and max.length as the maxi
       tally mat as the mean of arr
    tally mst as the mean of svc
   tally mwt as the mean of wait.tim
   define destroy.customer as integer variable
   define n.departures as integer variable
   define n.arrivals as integer variable
  accumulate xbusy as the average   of channel
   define n,times1,i,count as integer variable
   define a,k,lamda as real variables
   define milliseconds to mean units
  define list,list1 as 1-dimensional integer arrays
    tally max.wait as the maximum of wait.tim
 end
   main
   read S      " buffer size
   read T      " service time for one packet
```

```
read A        " arrival time(single inter arrival time)
 reserve list(*)  as 32  "list 1-dimensional array to collect burst cell lengths
  reserve list1(*) as 32   "list1 1-dimensional array to collect wait times and
"plot histogram
  let count = 1
 let n.arrivals = 0
let n.departures = 0
    let arr = exponential.f(A.,1)
 schedule an arrival in  arr milliseconds
let destroy.customer = 0
start simulation
 end
event arrival
let n.arrivals = n.arrivals + 1 " counter to keep track of the number of arrivals
  let arr  = exponential.f (A,1)
schedule an arrival in arr milliseconds
 create a  packet
  let entry.time(packet) = time.v
  file this packet in the queue
   if n.queue    >  S    "condition  to test whether queue of packets is greater
"than buffer
       add 1 to destroy.packet " counter to accumulate the loss of packets due
to buffer overflow
       remove this  packet from the queue
  destroy this packet
     add 1 to count
     else
        if n.queue <=  S  " loop condition to determine the burst cell length
in "other words finding out whether two cells are lost in a row ,3 cells lost in
row   "accordingly  and adding 1 to list(2) or list(3) so on
     if count > 0
      if count <= 10
      add 1 to list(count)
        else
      add 1 to list(11)
         endif
          endif
          let count = 0
          endif
           endif
  if channel = idle
 let channel = busy
  remove first  packet from the  queue
  let wait.tim = 0
  if wait.tim = 0
```

```
      add 1 to list1(22)
      always
      let svc =   T   " the service time is constant(since it is M/D/1/K queue)
    schedule a departure in svc milliseconds
destroy this packet
     always
    return
  end
  event departure
let n.departures = n.departures + 1
   if n.departures = 5000000   " programm asked to stop and print out results
    print 1 line with n.queue thus
    n.queue is ***********
    schedule a stop.sim now
    always
   if queue is empty
   let channel = idle
  jump ahead
   else
   remove the first packet from the queue
   let wait.tim = time.v - entry.time(packet)   "classfying the wait time ranges to
"obatain  histogram
       if wait.tim > 0 and wait.tim <= .06
        add 1 to  list1(1)
         always
           if wait.tim > .06
          for i = 2 to 21
destroy this packet
     always
    return
  end
  event departure
let n.departures = n.departures + 1
   if n.departures = 5000000          " a total of 5 million cells are transmitted
"through the channel  and then programm is asked to stop and display queue
"stastics
    print 1 line with n.queue thus
    n.queue is ***********
    schedule a stop.sim now
    always
   if queue is empty
   let channel = idle
  jump ahead
   else
   remove the first packet from the queue
```

```
  let wait.tim = time.v - entry.time(packet)
     if wait.tim > 0 and wait.tim <= .06   " after classifying the differnt  wait
"times  according  to  the  range  (5  %  increments)  and  then  accordingly
"incremmenting the corresponding  total
      add 1 to  list1(1)
       always
         if wait.tim > .06
        for i = 2 to 21
 for i = 2 to 21
          do
         let k = k + .06
        if wait.tim > k  and wait.tim <= (.06) * (i)
          add 1 to list1(i)
        always
      loop
     let k = 0
        always
   let svc =  T " service time is constant
  schedule a departure in svc milliseconds
    destroy this packet
   here
  return
 end
 event stop.sim
    print  5 lines with list1(1),list1(2),list1(3),list1(4),
list1(5),list1(6),list1(7),list1(8),list1(9),list1(10),list1(1
list1(12),list1(13),list1(14),list1(15),list1(16),list1(17),
list1(18),list1(19),list1(20),list1(21),list1(22) thus
     stop
 ******** ********* *********  ********* ******** *******
 ******** ********** ********  *********  ******* ********
 *******  **********  ********  ********  *******  *******
 *******     *********       **********     ********
  print  2  lines with list(1),list(2),list(3),list(4),list(5
 list(6),list(7),list(8),list(9),list(10),list(11),list(12),
   list(13),list(14),list(15),list(16) thus
 ***** ***** ******* ****  ***** ****** ****** ***** ******
  ***** ******** ****** ******* ******* ****** ******

   print  13 lines with
    aniq,
   xbusy,
  mat,
  mst,
 mwt,n.arrivals,n.departures,  destroy.customer,arr,and max.le
```

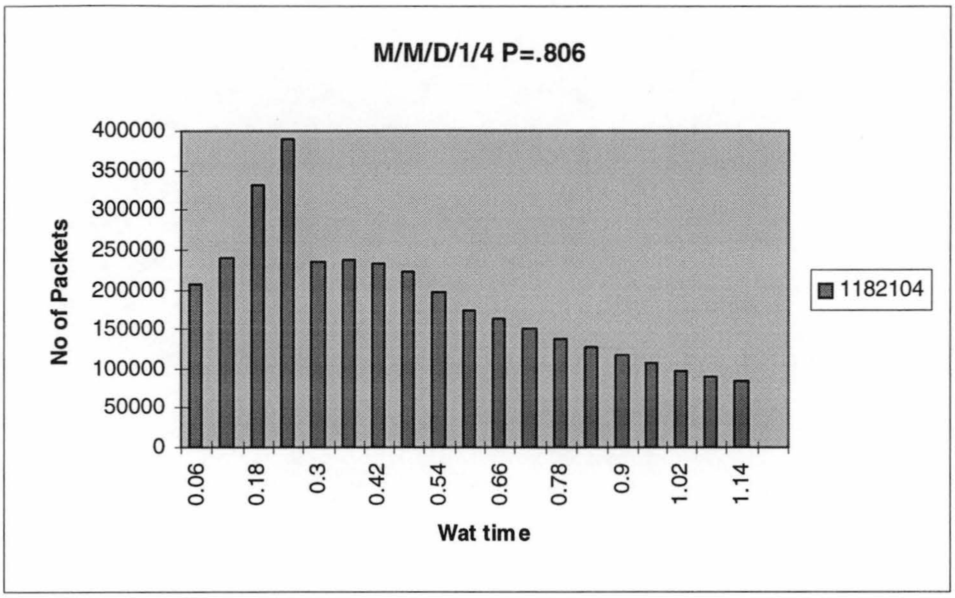
```
mwt,n.arrivals,n.departures, destroy.customer,arr,and max.le
times1,count,max.wait thus
    average queue length **.******* minutes
     utilisation of server ***.******
     mean arrival time of packet is ****.*****************
      mean service time of packet is **.*******
        average wait time is **.********
    number of arrivals *****************
     no of departures ***************
    destroy customer ***************************
     b = ***.********
   max.length = *********.***********
times1 = ***********.***
  count = **********
   maximum wait = ************.**********
     add 1 to times
   if times = 1
  stop
```

The following is the above **programms outputfile** which printed out the various queue stastics which are of interst

```
     n.queue is      2
     stop
  205473   240760   282354   331434  388693  234382
  237311   232232  220568   195966  174493  163693
 151504   138831   126383  116195  107006   97970
  89843        82806              0      1182104
111939 26180   4852  792    92    13    3    0    0
    0     0    0    0    0    0    0
    average queue length   .9346095 minutes
     utilisation of server    .771934
     mean arrival time of packet is    .37494287663808298
      mean service time of packet is  .3000000
        average wait time is  .36322138
    number of arrivals          5182583
     no of departures        5000000
    destroy customer                  182581
    max.length =        5.00000000000
   maximum wait =           1.1999992993

"bar2" 21 lines, 905 characters
```

From the above output file we can observe that the maximum wait time is 1.199999(milliseconds simscript time units) no of packets transmitted are 5 million Now the first row immediately gives the no of packets interms of wait times 0 to 0.06,0.06 to 0.12 and so on the buffer length is 4 the loss is 3 to 4 % Now the histogram is plotted for this utilisation that is 80.6%



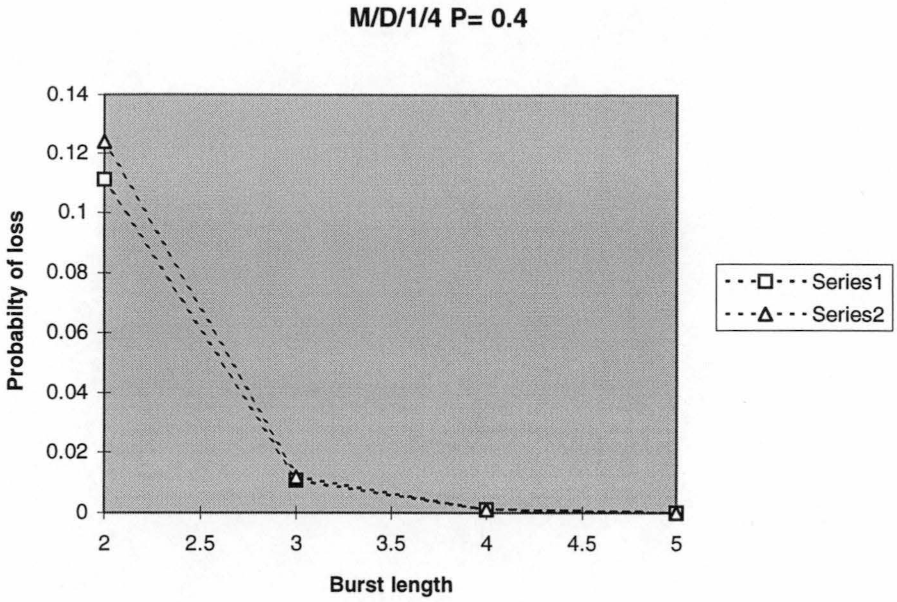
Now 5th row on wards after stop are the readings for burst length distributions
first reading = 111939(single cell loss total no)
second reading = 26180(two cells in a row total no)
and so on

Now burst is of length 2 = total no of two cells in a row loss/total single loss(From conditional loss theory) At utilisation = .8
Therefore pr(burst length 2) = 26180/111939= .2338
pr(burst length 3) = 4852/111939 = .043
pr(burst length 4) = 792/111939 = .00707
pr(burst length 5) = 0.000821

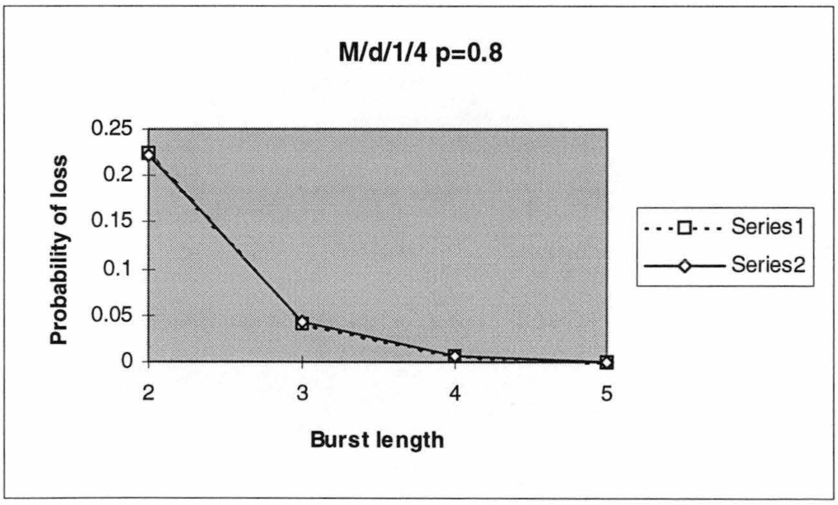
TABLE4 (Simultaed/Predicted)

P	2		3		4		5	
0.4	0.111	0.124	0.0108	0.0119	0.0012	0.001	0.0001	0.0001
0.5	0.138	0.154	0.0151	0.0182	0.0014	0.0016	0.0001	0.0002
0.6	0.166	0.181	0.0225	0.0256	0.0026	0.003	0.0003	0.0003
0.7	0.195	0.207	0.0303	0.0337	0.004	0.00406	0.0005	0.0005
0.8	0.224	0.223	0.04	0.043	0.006	0.007	0.0007	0.0008
0.9	0.253	0.258	0.0513	0.0528	0.008	0.09	0.0012	0.0017

GRAPH3



GRAPH4



6.0 Switched Poisson Process(*SPP/D/1/K*) :

The switched Poisson process has been suggested as a good model for bursty traffic such as video and data ,where the channel switches between periods of high and low activity [Heffes and Lucantoni]:

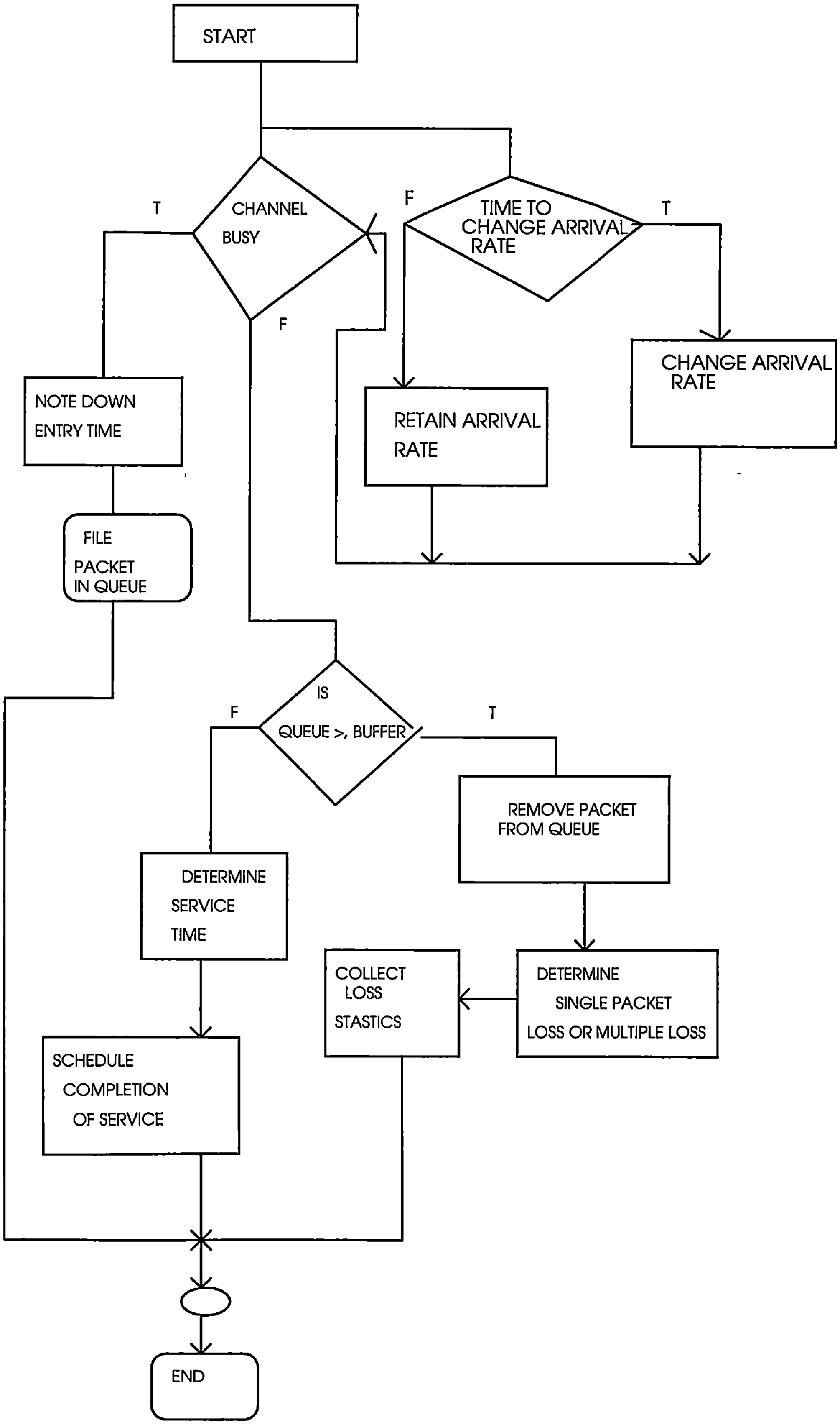
The switched Poisson process is characterised by four parameters $\lambda 1, \lambda 2, \gamma, \omega$, .The process spends exponentially distributed with mean $1/\gamma$ milliseconds in an on mode where arrivals occur with an inter arrival time described by a Poisson distribution with mean $1/\lambda 1$; and spends $1/\omega$ milliseconds in an off mode where arrivals occur with mean $1/\lambda 2$,where $\lambda 1 > \lambda 2 \geq 0$.When $\lambda 2=0$ the process becomes Interrupted Poisson Process(IPP)

Conditional cell loss probabilities can be approximated by linearly interpolating the probabilities in the 'on' and the 'off' mode

$$\text{pr(k cell/1 cell loss)} = \frac{\gamma B(\lambda 1) + \omega B(\lambda 2)]}{(\omega + \gamma)}$$

where $B(\lambda)$ is the conditional probability of cell loss when the arrival rate is λ
The following is the flow chart for the arrival event,the logic is same for departure event process as in M/M/1/K and M/M/D/1/k queues hence not reproduced here

EVENT ARRIVAL



7.0 The following is the source code program for **SPP/D/1/K** program

```
 preamble
  "ISDN statistical multiplexer Rama Mohan Anne 1995
    "Use Simscript to model the behaviour of buffer in particular
     "to bursty traffic
   the system owns the queue
  permanent entities
   every source has an act.time and an inact.time
  temporary entities
every  packet has an entry.time and may belong to the queue
   event notices include  act,arrival,departure and stop.simulat
   "event notices include arrival1, .. arrival8
   define queue as a fifo set
    define switch.flag as integer variable
  define idle to mean 0
   define busy to mean 1
   define channel,n.arrival,n.departure,times,times1,i as intege
   define arr, svc,wait as real variables
   tally avg.wait as the mean and max.wait as the maximum of  wa
   accumulate utilization as the mean of channel
     define m as real variable
     define milliseconds to mean units
     accumulate avg.length as mean and max.length as maximum of
    define  destroy.packet as integer variable
define  destroy.packet as integer variable
    define count,count1 as integer variables
    define p,r,s,k as real variables
    define list,list1,list2,list3 as 1-dimensional double arrays
     every arrival has a source.no
    every departure has a source.no
    every act has a source.no
    define count2 as real variable
   define a,b,c,d as real variables
    define buffer.size,count4 ,z  as integer variable
        end
    main
  READ A " this is active period duration with mean A (exponent
" distributed )
 READ B "this is the inactive period duration with mean B(expon
"distributed)
 READ C   "this corresponds to single interarrival time  with m
  "exponentially distributed
 READ D "this corresponds to single interarrival time with mean
 "exponentially distributed
```

```
 READ BUFFER.SIZE  "the size of the buffer
  read s "service time
   reserve list(*) as 32
create every source(150) " no of sources

   let n.arrival = 0
    let switch.flag = 1
   let n.departure = 0
    let time.v = 0
   let channel = idle
    n.source = 1 "this variable can be varied if more than 1 source
 "that  is for n.source = 1 to z
      add act.time(n.source) to count1
    "print 1 line with act.time(n.source) thus to debug
     "act.time = *********.***********
 let act.time(n.source) = exponential.f(a,1)
   schedule an arrival(n.source) now
     "loop
   start simulation
    end
  event arrival(n.source)

   let arr = exponential.f(c,1)
  schedule an arrival(n.source) in arr milliseconds
    if time.v > act.time(n.source) "time to be in inactive state
    let inact.time(n.source)   = exponential.f(b,1)
    add   1  to count  " to keep track of no of times it was in active
     let c = d  "swapping arrival time
 schedule an act(n.source) in  inact.time(n.source) milliseconds
    always
  create a packet
  let n.arrival = n.arrival + 1
     let entry.time(packet) = time.v
      file this packet in the queue
      if n.qUeue  > buffer.size
          remove this packet from the queue
          add 1 to count4 "counter to count no of cells lost
        destroy this packet
        let destroy.packet = destroy.packet + 1
       else
      if n.queue <= buffer.size
       if count4 > 0
      if count4 <= 10
    add 1 to list(count4)
      else
```

```
 add 1 to list(11)
endif
 endif
 let count4 = 0
endif
endif

    if switch.flag = 1 "to calculate the time the queue is
"found in a particular state
     if n.queue = 8 "this  can be varied for any size
      let switch.flag = 0
        let p = time.v
         always
        always
       if switch.flag = 0
        if n.queue ne 8
         let r = time.v
       let k = (r - p)
      add k to count2
       let switch.flag = 1
       always
      always
    if channel = idle,
      let channel = busy
remove first packet from the queue
                    let wait = 0
                 let svc = s
      schedule a departure(n.source) in  svc milliseconds
           destroy this packet
        always
     return
   end
  event departure(n.source)
   let n.departure = n.departure + 1
       if n.departure >= 30000 "directing programm to stop a
 "this number of departures
     schedule a stop.simulation now
       always
    if queue is empty,

   let channel = idle
    if channel = idle
     add 1 to list1(0)
    always
     jump ahead
```

```
   else
 remove first packet from the queue
let wait = time.v - entry.time(packet)
    let svc =  s
  schedule a departure(n.source) in svc milliseconds
  destroy this packet
   here
  return
  end
event stop.simulation
  print 4 lines with list(1),list(2),list(3),list(4),
list(5),list(6),list(7),list(8),list(9),list(10),list(11),
list(12),list(13),list(14),list(15),list(16),list(17),
list(18),
  list(19),list(20),list(21),list(22) thus
 ******** ********** ********* ********* *********
 ********* *********** ********** ********* *************
********* ********** ********** ******** ********** ********
********* *********** ******** ********* ********* ************
  print 14 lines with n.arrival,n.departure,avg.wait,avg.length
  max.length,  utilization,times, m,destroy.packet,max.wait,
   times1,
    count,count1,count2 thus
  number of arrival *********.********
  number of departures ****************.******
let wait = time.v - entry.time(packet)
    let svc =  s
  schedule a departure(n.source) in svc milliseconds
  destroy this packet
   here
  return
  end
event stop.simulation
  print 4 lines with list(1),list(2),list(3),list(4),
list(5),list(6),list(7),list(8),list(9),list(10),list(11),
list(12),list(13),list(14),list(15),list(16),list(17),
list(18),
  list(19),list(20),list(21),list(22) thus
 ******** ********** ********* ********* *********
 ********* *********** ********** ********* *************
********* ********** ********** ******** ********** ********
********* *********** ******** ********* ********* ************
  print 14 lines with n.arrival,n.departure,avg.wait,avg.length
  max.length,  utilization,times, m,destroy.packet,max.wait,
   times1,
```

```
  count,count1,count2 thus
 number of arrival *********.********
 number of departures ****************.******
avg.wait ******.*******
  avg.length *********.*********
  max.length *********.********
   utilization of channel *****.*******
   times *******.*******
    m ******.******
  destroy packets = ************
   max.wait = ********.**************
    times1 = *********.**********
    count = ***************.*******
   count1 = **************.*******
   count2 = *******************.***************
 print 1 line thus
 these are results for     20 sources
    stop
  end
 Event act(n.source)
 let s = exponential.f(a,1)  " again generating active period
   add  1 to count1
  let act.time(n.source) =  s + time.v
let d = c "to swap arrival time
    return
   End
~
```

The predicted cell loss for two cells with gamma and omega equal is 0.168

The table shows that if switching times are very small the simulated value deviates from the predicted this is because of hangover effect of switching between states

TABLE 5

Switched Poisson Process

Equal switching times

$\lambda 1 = .8$ **and** $\lambda 2 = .4$

γ	ω		
		2	3
0.0001	0.0001	0.162	0.02
0.01	0.01	0.1536	0.02
0.1	0.1	0.169	0.023
1	1	0.182	0.027
10	10	0.21	0.038
100	100	0.215	0.039
1000	1000	0.218	0.038

GRAPH 5

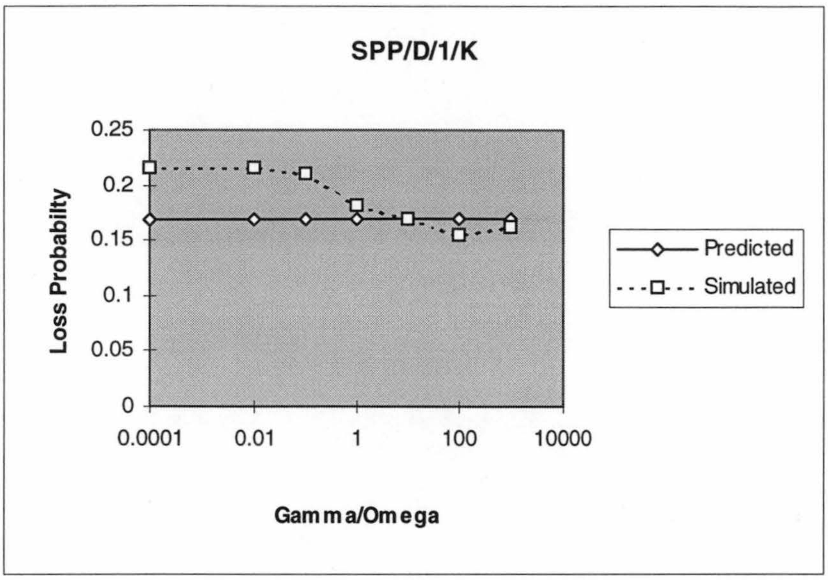


TABLE 6

Switched Poisson Process
$\lambda 2 = .4, \lambda 1 = .8$
Predicted and Simulated

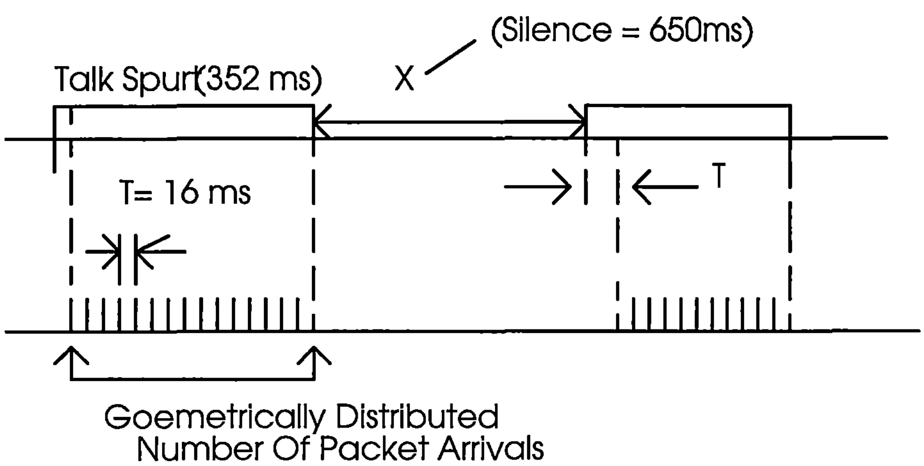
γ	ω						
		2		3		4	
5	2	0.143	0.156	0.0123	0.0198	0.0026	0.0022
2	1	0.149	0.177	0.0214	0.0215	0.0028	0.0032
1	0.5	0.174	0.178	0.025	0.0215	0.0028	0.0023
1	1	0.1667	0.181	0.0254	0.0271	0.0034	0.0035
0.1	1	0.213	0.224	0.046	0.0411	0.0059	0.0062
0.01	1	0.223	0.197	0.04	0.041	0.006	0.006
0.5	1	0.187	0.199	0.0206	0.0318	0.0043	0.0044
0.1	0.01	0.213	0.224	0.0373	0.034	0.0056	0.005
0.4	1	0.134	0.164	0.034	0.023	0.005	0.0026
1	0.01	0.111	0.113	0.0112	0.009	0.0012	0.0018

It is clear that the simulated and predicted values are matching not quite closely as the predicted values are obtained from interpolation of the ***M/M/D/1/K*** **QUEUE** ,also the hangover effect is more if the switching times are very small .

RESULTS SECTION2:

1.0 Bursty Source Model 1:

To investigate the characteristics of the bursty sources ,a voice model is first considered .A single voice source model alternates between an active state and idle state which are exponentially distributed .The numerical value(mean) for active period is 352 milliseconds and the inactive period (mean) is 650 milliseconds ,in the active mode the single source generates a packet every 16 milliseconds ,the following figure illustrates the assumption for a single voice source



Single voice Source Model

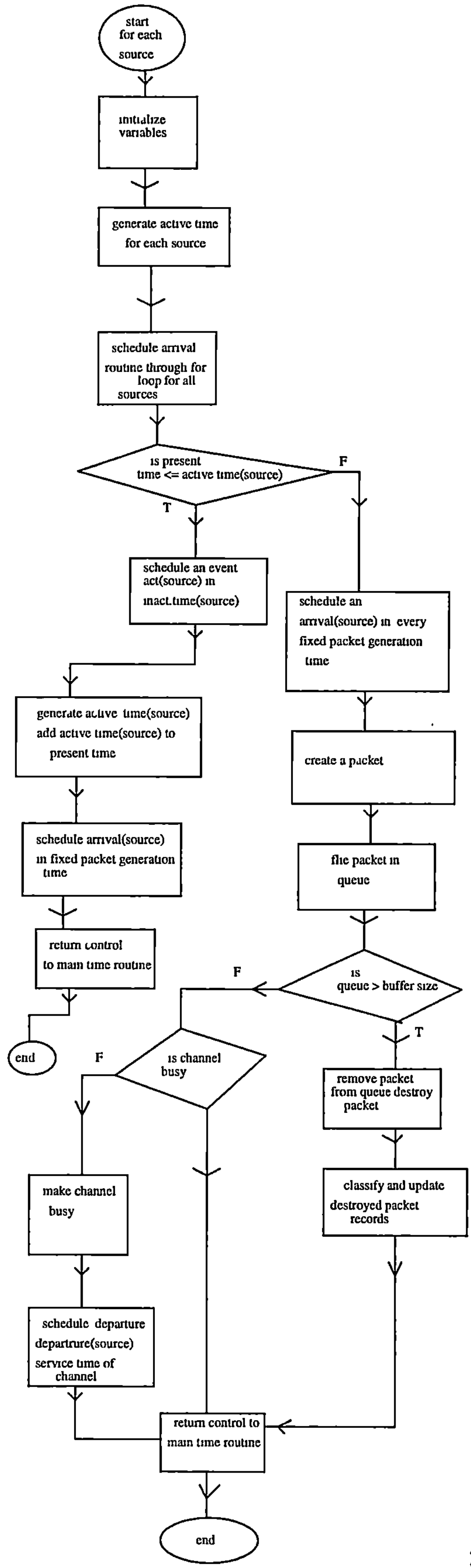
Each voice generates packets at fixed intervals [Sriram and Whitt IEE TRANS sept 1986] during talk spurt and no packets during the silence period ,thus there are 22 mean number of packets(352/16) during the talk spurt period.

The probability of finding a packet with 16 ms inter arrival time is 21/22 and the probability of finding a packet with 666ms(silence period + 22ms) since the next packet does not arrive until after 16ms of start of a next talk spurt period.,The packet is assumed to be of 64 bytes and the link is T1 rate link 1.536 MB/S .Now to saturate this link approximately 136 voice sources of the above described voice sources are required (22*512*136 = 1.536 MB/S) .

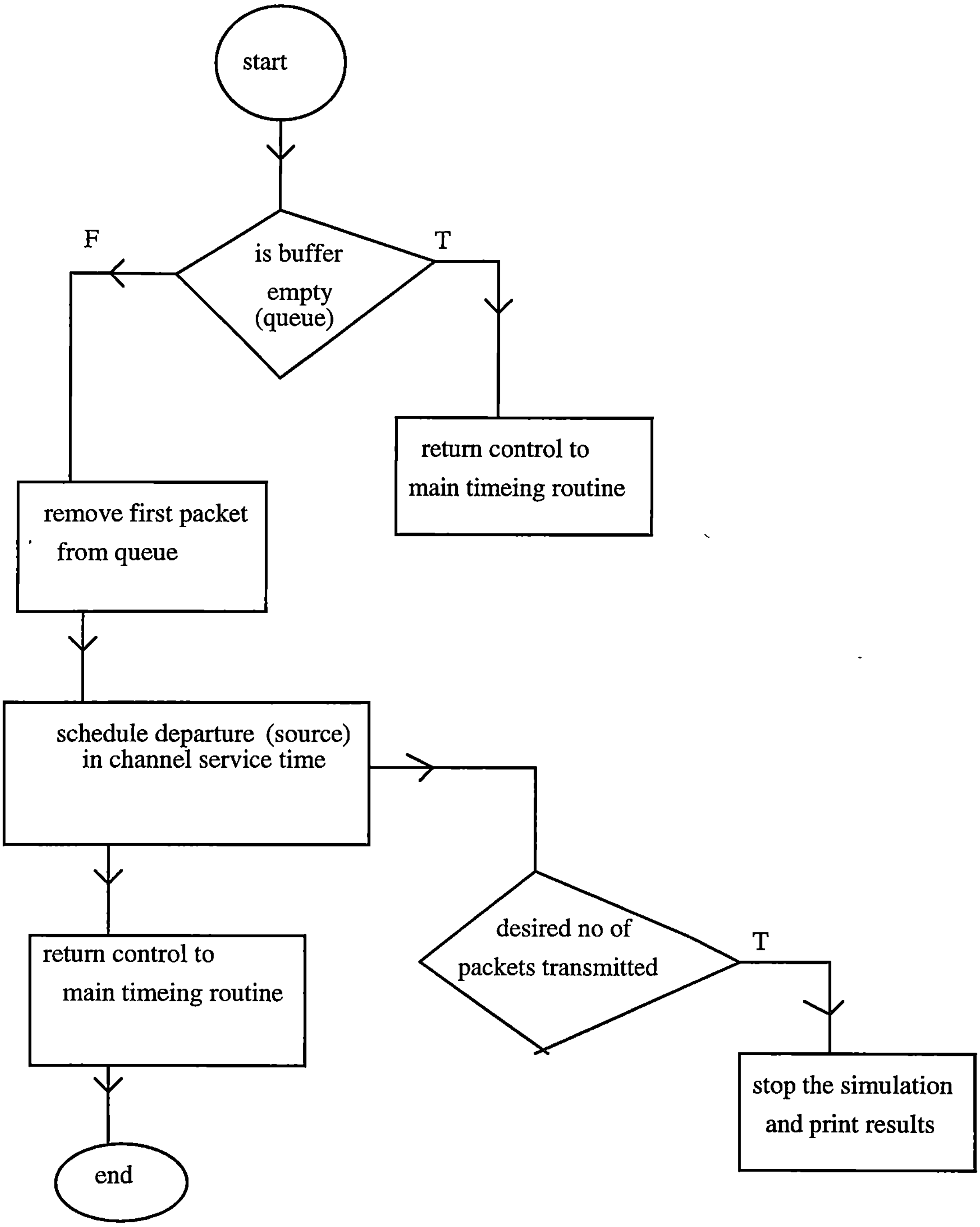
Now the program is written to model these sources .The number of voice sources taken to plot the graph of probability of cell loss vs no of sources is from 110 to 135 voice sources .The service time is .33 milliseconds since the ATM cells are of fixed size .To investigate where actual the normal Poisson approximation fails to model the Bursty sources.Since the Mathematics becomes difficult to suggest a suitable analytic model Simulation is carried out to investigate the characteristics. Now the following is the source code program ,the preamble contains all the global variables declared ,each source is declared with an active time and inactive time ,a loop is written in the MAIN to enable arrival process for more than one source .The buffer length starting from 4 to 61 is varied . It is noted that at only buffer length = 61 there is marked difference in predicting the cell loss probabilities compared to the ***M/D/1/K*** queue (Ramesh nagarajan,James F.Kurose IEEE journal on commun,apr1991 vol9) .The reason is that buffer length is more the queue is long the inter arrival times can interact from the superposition of the sources compared to the smaller buffer lengths .This is proved by the graph plotted with the program output data..

The following are the Flow chart diagrams for the **arrival routine** and **departure** routine

Event Arrival:



EVENT DEPARTURE



2.0 Bursty Source Program Code:

```
preamble
  "ISDN satastical multilexer Rama Mohan Anne 1995
    "Use Simscript to model the behaviour of buffer in particu
     "to bursty traffic
   the system owns the queue
  permanent entities
   every source has an act.time and an inact.time
  temporary entities
every  packet has an entry.time and may belong to the queue
   event notices include  act,arrival,departure and stop.simulat
   "event notices include arrival1, .. arrival8
   define queue as a fifo set
    define switch.flag as integer variable
  define idle to mean 0
   define busy to mean 1
   define channel,n.arrival,n.departure,times,times1,i as intege
   define wait as real variables
   tally avg.wait as the mean and max.wait as the maximum of  wa
   accumulate utilization as the mean of channel
     define m as real variable
     define milliseconds to mean units
     accumulate avg.length as mean and max.length as maximum of
    define  destroy.packet as integer variable
 define count,count1 as integer variables
    define p,r,s,k as real variables
    define list,list1,list2,list3 as 1-dimensional double arrays
     every arrival has a source.no
    every departure has a source.no
    every act has a source.no
    define count2 as real variable
     define b,a,svc,arr as real variables
     define buffer.size as integer variable
        end
    main
     read a "type in the mean active period
      read b  "type in the mean inactive time
      read buffer.size "type in the buffer size
      read svc   "type in the service time
      read arr  "type in the arrival time
      read z "type in the no of sources
    reserve list(*),list1(*),list2(*),list3(*) as 150
    create every source(z)
    let n.arrival = 0
```

```
  let switch.flag = 1
 let n.departure = 0
  let time.v = 0
define count,count1 as integer variables
 define p,r,s,k as real variables
 define list,list1,list2,list3 as 1-dimensional double arrays
  every arrival has a source.no
 every departure has a source.no
 every act has a source.no
 define count2 as real variable
  define b,a,svc,arr as real variables
  define buffer.size as integer variable
    end
 main
  read a "type in the mean active period
   read b  "type in the mean inactive time
   read buffer.size "type in the buffer size
   read svc   "type in the service time
   read arr  "type in the arrival time
   read z "type in the no of sources
 reserve list(*),list1(*),list2(*),list3(*) as 150
 create every source(z)
 let n.arrival = 0
  let switch.flag = 1
 let n.departure = 0
  let time.v = 0
let channel = idle
  for n.source = 1 to z
  do
    add act.time(n.source) to count1
  "print 1 line with act.time(n.source) thus
   "act.time = *********.***********
let act.time(n.source) = exponential.f(a,1)
 schedule an arrival(n.source) now
   loop
 start simulation
  end
 event arrival(n.source)
  if time.v <= act.time(n.source)
 schedule an arrival(n.source) in  arr milliseconds
   always
  if time.v > act.time(n.source)
  let inact.time(n.source)   = exponential.f(b,1)
  add   inact.time(n.source) to count
```

```
schedule an act(n.source) in  inact.time(n.source) millisecond
   always
 create a packet
 let n.arrival = n.arrival + 1
let entry.time(packet) = time.v
     file this packet in the queue
     if n.qUeue  >=  buffer.size
        remove this packet from the queue
      destroy this packet
      let destroy.packet = destroy.packet + 1
      else
      if switch.flag = 1
      if n.queue = 8 " code to determine how long the queue
         "is 8
       let switch.flag = 0
         let p = time.v
          let b = 0
          always
         always
        if switch.flag = 0
         if n.queue ne 8
          let r = time.v
        let k = (r - p)
       add k to count2
        let switch.flag = 1
        always
       always
always
      if channel = idle,
       let channel = busy
         remove first packet from the queue
                    let wait = 0
       schedule a departure(n.source) in  svc milliseconds
            destroy this packet
         always
      return
   end
  event departure(n.source)
   let n.departure = n.departure + 1
       if n.departure >= 30000    "the programm is directed to stop
      schedule a stop.simulation now
        always
      if queue is empty,

    let channel = idle
```

```
      jump ahead
    else
  remove first packet from the queue
     let wait = time.v - entry.time(packet)
   schedule a departure(n.source) in svc milliseconds
 destroy this packet
    here
   return
   end
 event stop.simulation  "these arrays are used when histogram
   " plot is needed
   print 4 lines with list2(0),list2(1),list2(2),list2(3),list2(
 list2(5),list2(6),list2(7),list2(8),list2(9),list2(10),list2(11
 list2(12),list2(13),list2(14),list2(15),list2(16),list2(17),
list2(18),
  list2(19),list2(20),list2(21),list2(22) thus
 ********* ******** ********** ********* ********* *********
  ********* *********** ********** ********* ************
 ********* ********** ********** ******** ********** ********
 ********* *********** ******** ********* ********* ************
   print 3 lines with list3(0), list3(1),list3(2),list3(3),list3
 list3(6),list3(7),list3(8),list3(9) thus
  ************* ************* ************* ************
  ************ ************** ************* *************
   ***********  *****************************
 print 5  lines with list1(0),list1(1),list1(2),list1(3),list1(4
 list1(5),list1(6),list1(7),list1(8),list1(9),list1(10),
 list1(11),list1(12),list1(13),list1(14),list1(15),list1(16),
list1(17),list1(18),list1(19),list1(20) thus   "list used when histogram is
"required  to plot
  ************* ************ ************ ********* *********
  ************* ************* *********** *********** *******
 ************* ************* ************ ********** **********
  *********** **************** ************* **************
  ******************* *********************
   print 14 lines with n.arrival,n.departure,avg.wait,avg.length
   max.length,  utilization,times, m,destroy.packet,max.wait,
    times1,
     count,count1,count2 thus
   number of arrival *********.********
   number of departures ****************.******
    avg.wait ******.*******
    avg.length *********.*********
    max.length *********.********
     utilization of channel *****.*******
```

```
times *******.*******
 m ******.******
destroy packets = ************
 max.wait = ********.**************
  times1 = *********.**********
  count = ***************.*******  " counter to add up total inactive times
 count1 = **************.*******  "counter to add up total active time
"periods

  stop
 end
Event act(n.source)
let s = exponential.f(a,1)
 add s to count1
 let act.time(n.source) = s + time.v  "time to generate one more active period
schedule an arrival(n.source) in arr milliseconds
  return
 End
```

Now the above programm is run with buffer size limited to **61** and service time is .33 milliseconds(**T1 link** assumed) , the cell size is **64 bytes.**The following are the output files starting from 110 sources

Ourtput1

```
number of arrival   3001094.00000000
  number of departures        3000000.000000
   avg.wait      .8525300
   avg.length       2.085372456
   max.length       61.00000000
    utilization of channel      .809658.
   destroy packets =        1093
    max.wait =       19.86000001430511
         count =        86937844.0000000  " total time spent in inactive state
    count1 =       46925869.0000000   "total time in active state
    count2 =              674.381469726562521  "the amount of time queue is
"having 61
  these are results for 110 sources
```

Output2

```
number of arrival   3002624.00000000
  number of departures        3000000.000000
   avg.wait    1.4173937
   avg.length       3.618839712
   max.length       61.00000000
    utilization of channel      .8450972
```

destroy packets = 2622
max.wait = 19.86000001430511
times1 = 0.
count = 87132995.0000000
count1 = 46941313.0000000
these are results for 115 sources

Output 3

number of arrival 3009693.00000000
number of departures 3000000.000000
avg.wait 2.8144123
avg.length 7.505315643
max.length 61.00000000
utilization of channel .8826918

destroy packets = 9691
max.wait = 19.86000001430511
times1 = 0.
count = 86884815.0000000
count1 = 47051629.0000000

these are results for 120 sources

Output 4

number of arrival 3025718.00000000
number of departures 3000000.000000
avg.wait 4.6006646
avg.length 12.724565892
max.length 61.00000000
utilization of channel .9154828
destroy packets = 25714
max.wait = 19.86000001430511
times1 = 0.
count = 87235438.0000000
count1 = 47308002.0000000
these are results for 125 sources

Output5

number of arrival 3047271.00000000
number of departures 3000000.000000
avg.wait 6.6259197
avg.length 18.817970930

max.length 61.00000000
utilization of channel .9400567
destroy packets = 47261
max.wait = 19.86000001430511
count = 88619830.0000000
count1 = 47639198.0000000
these are results for 130 sources

Output6

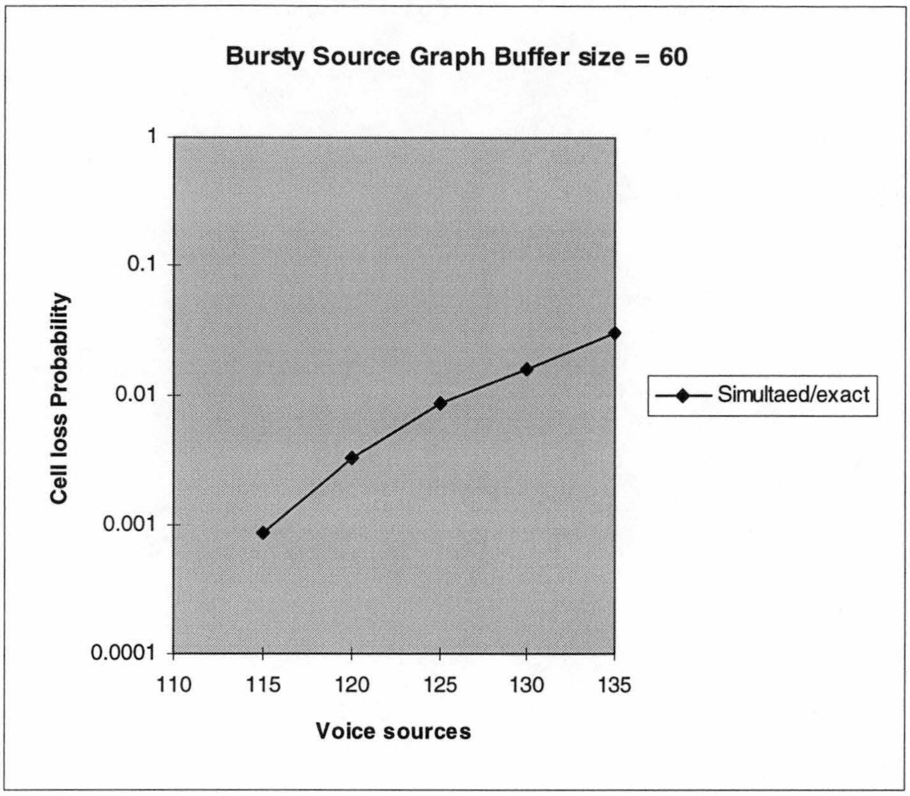
number of arrival 3096807.00000000
number of departures 3000000.000000
avg.wait 9.6381895
avg.length 28.052835766
max.length 61.00000000
utilization of channel .9633869
destroy packets = 96748
max.wait = 19.86000001430511
times1 = 0.
count = 89646584.0000000
count1 = 48426558.0000000
these are results for 135 sources

From the above output files the ratio of total packets destroyed to total number of packets is evaluated(**Blocking probability**) ,the following table gives the voice sources and the respective blocking probabilities

Table 7
Voice Source/Loss probabilty

110	
115	0.000874
120	0.00323
125	0.00857
130	0.01575
135	0.03

Graph 6
Scale: Log Scale



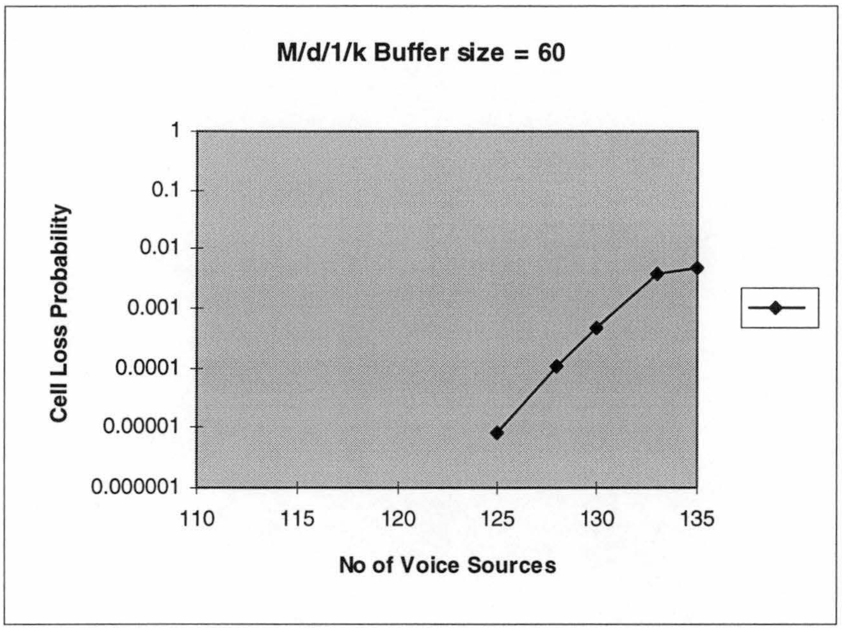
M/D/1/K queue program is run for the equivalent utilisation that is for example 110 sources utilise (110*22*512= 1.239 MB/S approximately 80.6%) ,so the arrival time is .33/.806 =.409 milliseconds ,similarly the other equivalent arrival times are given as input to the program and the following is the graph plotted

Table7

Voice sources/loss probability

110	
115	
120	
125	0.000008
128	0.000103
130	0.000464
133	0.003693
135	0.0048

Graph 7
Scale:Log Scale



From the above two graphs it is clear that ***Poisson*** approximation clearly underestimates the cell loss under bursty traffic ,thus alternative models are called for to predict cell loss under these conditions.Also the important conclusion is that ,this deviation of Poisson approximation comes into effect for only large buffer sizes(typically >= 60) for the above mentioned link and the cell size .The reason for this at large bufer size the number of inter arrival times interacting in the queue will be more ,and also the small term positive co variances (Sriram and Whitt I.E.E.E jr on Commun sept 1986) is more and the conventional Poisson approximation for the arrival process fails.

3.0 Bursty Source Model 2

Next the above parameters are varied that is cell size now is only **53** octets(**48 octets pay load and 5 octets header**) Three sets of bursty sets are considered Bursty set1,Bursty set2,Bursty set3

BURSTY SET1:["Simulation Analsis Of a Communication Link With Stastically Multiplexed Bursty Voice Sources" Mohsen A.saleh,Ibrahim W.Habib and Tarek N. Saadwai,I.E.E.E journal on select area in commun,april 1994)

The active period for each source is exponentially distributed with mean **352 ms** and the inactive or idle period is exponentially distributed with mean **650 ms** ,the link considered is again **T1 link** the service time is **.256** milliseconds ,considering that 150 sources saturate the link ,the arrival time is calculated as follows

Utilisation = 150 *(average bit rate)/link capacity
Therefore average bit rate = link capacity/150 (since utilisation = 100%)

Average bit rate = 1544000/150
= 10,293 bits/sec

Peak bit rate = average bit rate/0.35 (0.35 is activity fcator)
= 10,293/.35
29,409.25 bits/sec

Inter arrival time = No of bits in a cell(packet)/Peak bit rate
= 384/29,409.52 = .013 sec (Pay load = 384)
= 13 milliseconds (approx)

Bursty set2

1. The mean active period is 1200 milliseconds exponentially distributed
2.The mean silence period is 2200 milliseconds exponentially distributed
3.The number of sources that saturate the link are 130 sources
4.The Inter arrival time for Bursty set2 is calculated similar to the Bursty set1 and it is 11.3 millisecods

Bursty set3

1.The mean active period is 450 millisecods(exponentially distributed)
2.The mean silence period is 550 milliseconds(exponentially distributed)
3.The number of sources that saturate the link are 120 sources
4 .The Inter arrival time is 13.4 milliseconds

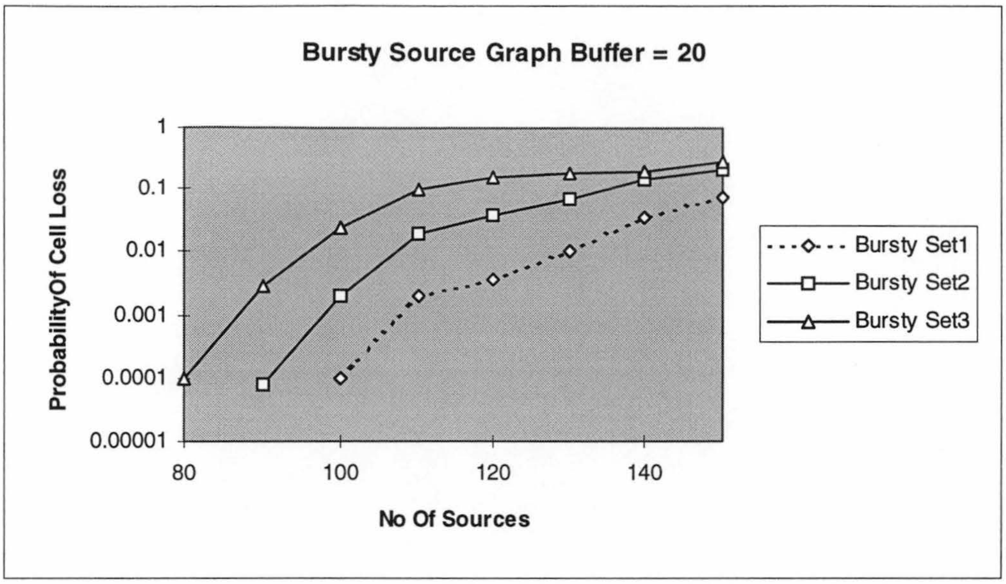
The following table gives the cell loss probabilities for the above described three sets from 80 to 150 voice sources

Table 8
Bursty Sources/Set1/Set2/Set3
Cell Loss Probability

x	y	y1	y2
80			0.0001
90		0.00008	0.003
100	0.0001	0.002	0.025
110	0.00199	0.02	0.1
120	0.0036	0.04	0.16
130	0.01	0.07	0.18
140	0.036	0.15	0.2
150	0.074	0.216	0.28

Graph8

Scale:Log Scale



Observation2:

From the above it can be observed that Bursty set3 has higher cell loss compared to Bursty Set2 and Bursty Set1 .This is because the activity factor of Bursty Set3 (0.45) is high ,therefore the average bit rate put on to the channel is high therefore utilisation is high ,and the cell loss is high .Again Bursty Set2 has higher cell loss compared to Bursty Set1 as it is more active (mean active period 1200 milliseconds) compared to Bursty Set1 (mean active period 352 milliseconds) ,even though the activity factors of the two sets are same(0.35)

Investigation:

To gain an insight into the study of cell loss and to meet the prescribed quality of service ,the Histogram plot of buffer length(queue length) vs the amount of time the length of the queue is in a particular state is ploted .

Buffer Occupancy/Set1/Set2/Set3
Percentage Of Time

0	3	2	1
1	3.6	2	1
2	2.5	1	0.8
3	1.6	1	0.8
4	1.2	0.8	1
5	1	0.8	0.8
6	1	0.7	0.8
7	0.5	0.7	0.7
8	0.5	0.9	0.9
9	0.5	0.8	0.8
10	0.8	1.1	0.9
11	0.7	1	0.7
12	0.8	0.9	0.8
13	1	1	1
14	2.7	1.5	1.5
15	3	2.8	2.7
16	5	2.8	3.7
17	8	11.8	12.5
18	12	12.8	13.04
19	16	21.9	22.9
20	33.9	36.9	37.4

Bursty Set1 = 164 Voice Sources Bursty Set2 = 142 Voice Sources Bursty Set3 = 133 Voice Sources Buffer Length = 20

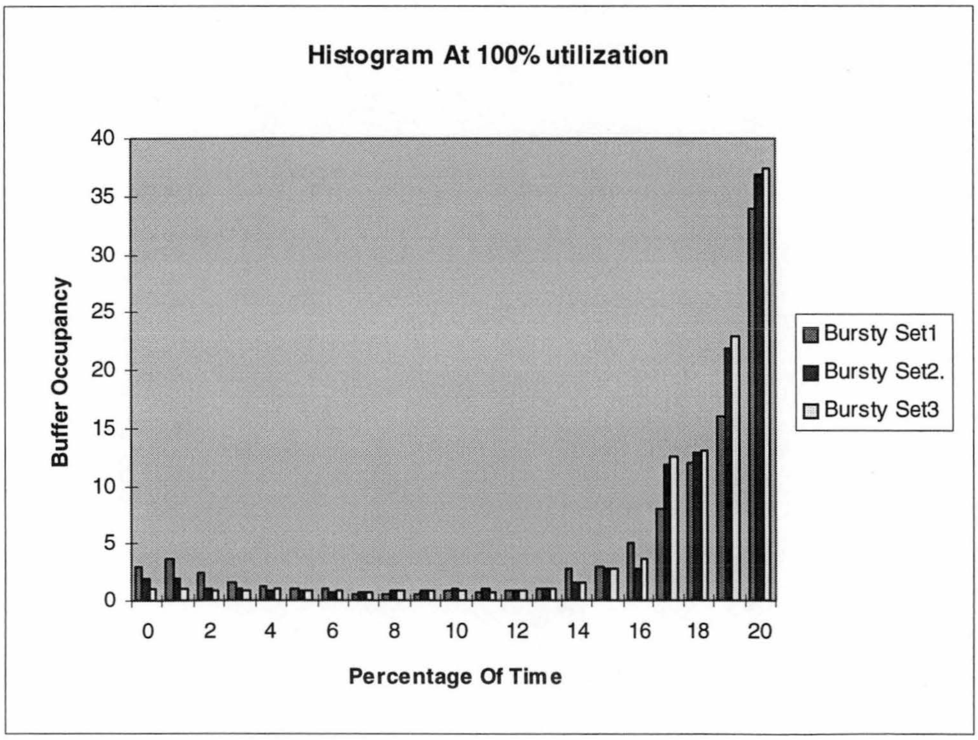
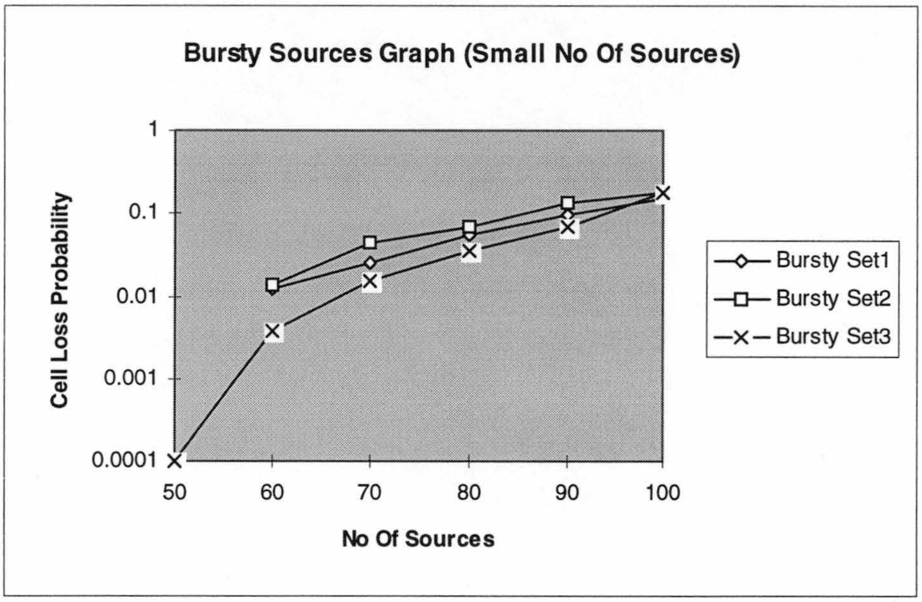


Table9
No Of Sources/Set1/Set2/Set3
(Cell Loss Probability)

50			0.0001
60	0.012164	0.0134	0.003861
70	0.02464	0.043277	0.015109
80	0.053546	0.068063	0.03476
90	0.09359	0.135	0.07048
100	0.15	0.177355	0.17708

Graph 9

Scale : Log Scale



Observation3:

The shape of the above graph is same as for large scale voice sources ,but the deviation of the graphs between each other is not much though the graph for Bursty Set3 starts loosing cells at 50% utilisation ,the graph of Bursty Set2 leads overall (more cell loss) ,it may be due to small inter arrival times interacting for a larger mean active time(1200 milliseconds) .Therefore a thorough investigation is called for to analyse the effects of smaller Inter arrival times in a mean active period.

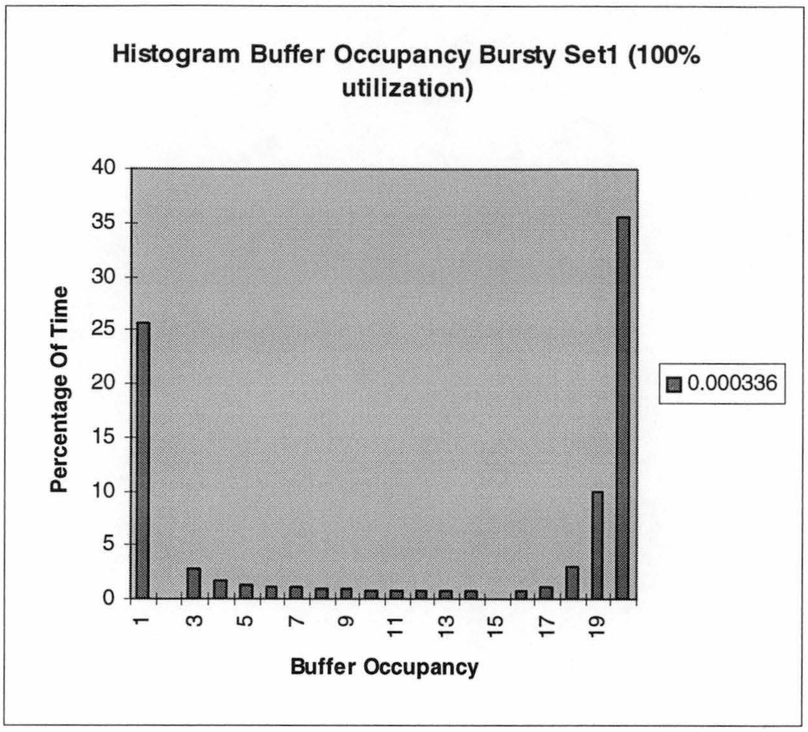
Investigation:

To study the cause of such high cell loss ,again a Histogram is plotted in a similar way as it was plotted for large number of sources .Bursty Set1 is considered and the utilization is at 100%

X1 = Buffer Occupancy
X2 = Total Time In That State(Simscript Time Units)
X3 = Total Simulation Time(Simscript Time Units)
X4 = (X2/X3)* 100

X1	X2	X3	X4
0	168	500000	0.000336
1	128120	500000	25.6
2	42947	500000	0.0858
3	17401.43	500000	2.8
4	8655.97	500000	1.73
5	7012	5000000	1.4
6	5820.7	500000	1.16
7	5010	500000	1.03
8	4502	500000	0.9
9	4270	500000	0.85
10	3976	500000	0.79
11	3917	500000	0.78
12	3925	500000	0.78
13	3951	500000	0.79
14	3894	500000	0.778
15	3890	500000	.0.778
16	4016	500000	0.8023
17	5244	500000	1.048
18	14742	500000	2.9
19	50380	500000	10
20	178160.3	500000	35.6

X4 along the Y-axis and X1 along the x-axis is plotted as histogram



Observation4:

It can be observed that 25.6% of the time it is in state1 (queue =1) and 35% of the time in state 20 (queue = 20) ,in other words queue is idle for .035% of time as the Inter arrival times are small [1.74 milliseconds(set1),1.47 milliseconds (set2),1.119 milliseconds(set3)] and the service time is .256 milliseconds, so during the mean active time the channel is utilised more compared to larger no of sources Inter arrival times more than 10 milliseconds.

RESULTS SECTION3:

1.0 Bursty Source Model3:

The following is the program source code written to evaluate Burst length probabilities for 6 Bursty Sources .The following are the parameters that are plugged into the program:

1.Buffer length = 20
2.Mean active time period = 450 milliseconds (exponentially distributed)
3.Mean silence period = 550 milliseconds(exponentially distributed)
4.Activity factor = 0.45
5.Number of Bursty Sources varied from 2 to 6

2.0 Source Code Program for Bursty Model3:

```
preamble
  "ISDN satastical multilexer Rama Mohan Anne 1995
   "Use Simscript to model the behaviour of buffer in particular
    "to bursty traffic
  the system owns the queue
 permanent entities
  every source has an act.time and an inact.time
 temporary entities
every  packet has an entry.time and may belong to the queue
  event notices include  act,arrival,departure and stop.simula
  "event notices include arrival1, .. arrival8
  define queue as a fifo set
   define switch.flag as integer variable
 define idle to mean 0
  define busy to mean 1
  define channel,n.arrival,n.departure,times,times1,i as integ
  define arr, svc,wait as real variables
  tally avg.wait as the mean and max.wait as the maximum of  w
  accumulate utilization as the mean of channel
    define m as real variable
    accumulate avg.length as mean and max.length as maximum of
   define  destroy.packet,switch.flag1 as integer variable
   define count,count1 as integer variables
define p,r,s,k,s1,s2 as real variables
   define list,list1,list2,list3 as 1-dimensional double array
    every arrival has a source.no
   every departure has a source.no
   every act has a source.no
   define count2,a as real variable
    define b,count4,count5,k1,k2,k3,k4,k12 as integer variable
  define milliseconds to mean units
 define k11,k13,k14,k15,k16,k17 as integer variables
 define list4,list5,list6 as 1-dimensional integer arrays
      end
   main
   reserve list(*),list1(*),list2(*),list3(*),list4(*),
    list5(*),list6(*)  as 10
   create every source(6)  "number of sources to be created
   let n.arrival = 0
    let switch.flag = 1
   let switch.flag1 = 0
   let n.departure = 0
    let time.v = 0
```

```
  let channel = idle
     for n.source =  1 to 3   "loop  to initialise the arrivals
       do
let act.time(n.source) = exponential.f(450,1)  " mean iactive time period
   add act.time(n.source) to count1
    schedule an arrival(n.source) in .349759 milliseconds "fixed interarrival time
"(utilization kept at .96 %)
      loop
    start simulation
     end
   event arrival(n.source)  " each source has an arrival associated with it
     if time.v <= act.time(n.source)
   schedule an arrival(n.source) in   .349759 milliseconds
      always
     if time.v > act.time(n.source)
     let inact.time(n.source)   = exponential.f(550,1)  " mean inactive time period
     add   inact.time(n.source) to count

  schedule an act(n.source) in  inact.time(n.source) millisecon
     always
   create a packet
   let n.arrival = n.arrival + 1
      let entry.time(packet) = time.v
       file this packet in the queue
      if n.qUeue  >= 21
          remove this packet from the queue
        destroy this packet
add 1 to list(n.source)
        let destroy.packet = destroy.packet + 1
     add 1 to count4
     if count4 = 1   "code to determine the burst length
     let k1 = n.source
   always
  if count4 = 2
    let k2 = n.source
   always
  if count4 = 3
   let k3 = n.source
  always
   if count4 = 4
  let k4 = n.source
  let count4 = 0
if k1 eq k2 eq k3 eq k4
 let k11 = k1
 add 3 to count5
  else
```

```
 if k1 eq  k2 eq k3
  let k12 = k1
 add 2 to count5
  else
if k2 eq k3 eq k4
 let k13 = k2
  add 2 to count5
  else
 if k1 eq k2
  let k14 = k1
 add 1 to count5
  else
 if k2 eq k3
 let k15 = k2
 add 1 to count5
 else
 if k3 eq k4
 let k16 = k3
  add 1 to count5
   else
  if count4 = 1
 if k1 eq k4
 let k17 = k1
 add 1 to count5
  always
 always
 always
always
 always
 always
 always
 always
 always
 else
   if count5 > 0  "code to test the burst lengths and assign them to arrays
    if count5 le 10
   if k12 = 1
   add 1 to list1(count5)
   always
  if k12 = 2
  add 1 to list2(count5)
  always
  if k12 = 3
  add 1 to list3(count5)
  always
  if k12 = 4
```

```
add 1 to list4(count5)
always
if k12 = 5
add 1 to list5(count5)
always
if k12 = 2
add 1 to list2(count5)
always
if k12 = 3
add 1 to list3(count5)
always
if k12 = 4
add 1 to list4(count5)
always
if k12 = 5
add 1 to list5(count5)
always
if k12 = 6
add 1 to list6(count5)
always
if k11 = 1
add 1 to list1(count5)
always
if k11 = 2
add 1 to list2(count5)
always
if k11 = 3
add 1 to list3(count5)
always
if k11 = 4
add 1 to list4(count5)
always
if k11 = 5
add 1 to list5(count5)
always
if k11 = 6
add 1 to list6(count5)
always
IF K13 = 1
add 1 to list1(count5)
always
if k13 = 2
add 1 to list2(count5)
always
if k13  =  3
add 1 to list3(count5)
```

```
always
if k13 = 4
add 1 to list4(count5)
always
if k13 = 5
add 1 to list5(count5)
always
if k13 = 6
add 1 to list6(count5)
always
if k14 = 1
add 1 to list1(count5)
always
if k14 = 2
add 1 to list2(count5)
always
if k14 = 3
add 1 to list3(count5)
always
if k14  = 4
add 1 to list4(count5)
always
if k14 = 5
add 1 to list5(count5)
always
if k14 = 6
add 1 to list6(count5)
always
if k15 = 1
add 1 to list1(count5)
always
if k15 = 2
add 1 to list2(count5)
always
if k15 = 3
add 1 to list3(count5)
always
if k15 = 4
add 1 to list4(count5)
always
if k15 = 5
add 1 to list5(count5)
always
if k15 = 6
add 1 to list6(count5)
always
```

```
if k17 = 1
add 1 to list1(count5)
always
if k17 = 2
add 1 to list2(count5)
always
if k17 = 3
add 1 to list3(count5)
always
if k17 = 4
add 1 to list4(count5)
always
if k17 = 5
add 1 to list5(count5)
always
if k17 = 6
add 1 to list6(count5)
always
if k16 = 1
add 1 to list1(count5)
always
if k16 = 2
add 1 to list2(count5)
always
if k16 = 3
add 1 to list3(count5)
always
if k16 = 4
add 1 to list4(count5)
always
if k16 = 5
add 1 to list5(count5)
always
if k16 = 6
add 1 to list6(count5)
always

   always
 let count5 = 0
let k11 = 0
let k12 = 0
let k13 = 0
let k14 = 0
let k15 = 0
let k16 = 0
let k17 = 0
```

```
 always
 always
      if channel = idle,
        let channel = busy
 remove first packet from the queue
                    let wait = 0
                 let svc =  .256
        schedule a departure(n.source) in  svc milliseconds
             destroy this packet
          always
       return
   end
   event departure(n.source)
    let n.departure = n.departure + 1
        if n.departure >= 2000000
      schedule a stop.simulation now
         always
      if queue is empty,

     let channel = idle
       jump ahead
    else
 remove first packet from the queue
     let wait = time.v - entry.time(packet)
     let svc = .256  "service time fixed T1 link assumed
  schedule a departure(n.source) in svc   milliseconds
  destroy this packet
 here
  return
  end
event stop.simulation
  print 1 line thus
These are results for cell loss (each source)
 print 1 line with list(1),list(2),list(3),list(4),
list(5),list(6) thus                                   " printing out the   total  cell
loss
 "for each source
********** ******** ******** ******* ****** *******
 print 1 line thus
These are results for cell loss (burst length) first source  "printing the individual
"burst lengths of each source
 print 1 line with list1(1),list1(2),list1(3),list1(4),
 list1(5) thus
******** ******** ******* ****** ********
print 1 line thus
 These are results for bursty length cell loss(second source)
```

```
print 1 line with list2(1),list2(2),list2(3),list2(4),
 list2(5)  thus
 ******** ******* ********** ******** *************
 print 1 line thus
 these are results for source number 3
print 1 line with list3(1),list3(2),list3(3),list3(4),
 list3(5)  thus
******* ********* ********* ********* **********
 print 1 line thus
 These are results for source number 4
 print 1 line with list4(1),list4(2),list4(3),list4(4),
  list4(5) thus
 *******  ********  *******  *******  **************
 print 1 line thus
 These are results for source number5
 print 1 line with list5(1),list5(2),list5(3),list5(4),
 list5(5)  thus
******* ******** ******** *************** ************
 print 1 line thus
 These are results for source number6
 print 1 line with list6(1),list6(2),list6(3),list6(4),
 list6(5)  thus
 ******** ********** ********* ******** ********
  print 12 lines with n.arrival,n.departure,avg.wait,avg.lengt
  max.length,  utilization,destroy.packet,max.wait,
    count,count1,count2,count5 thus
  number of arrival *********.********
  number of departures ****************.******
   avg.wait ******.*******
   avg.length *********.*********
max.length *********.********
    utilization of channel *****.*******
   destroy packets = ************
    max.wait = ********.**************
     count = ***************.*******
    count1 = **************.*******
    count2 = *******************.***************
   count5 = ************
  stop
   end
  Event act(n.source)
  let s = exponential.f(450,1)  "again generating the mean active period
    add s to count1
   let act.time(n.source) =  s + time.v "adding present time and updating the "
"active time
  schedule an arrival(n.source) in .349759  milliseconds
```

return
End

3.0 The following are the **programm output files** produced here

Output2:(Bursty Sources = 2)

Next keeping the utilisation = 96% and sources = 2 act.factor = .45
Interarrival time = .2331606 milliseconds
% a.out
These are results for cell loss (each source)
312550 299213 0 0 0 0
These are results for conditinal cell loss (burst length) source1
11699 18844 47691 0 0
These are results for conditional cell loss(burst length) source2
12044 17700 44962 0 0
these are results for conditional cell loss(burst length) source3
0 0 0 0 0
These are results for conditional cell loss(burst length) source4
0 0 0 0 0
These are results for conditional cell loss(burst length) source5
0 0 0 0 0
These are results for conditional cell loss(burst length) source 6
0 0 0 0 0
number of arrival 2111783.00000000
number of departures 1500000.000000
avg.wait 4.8081270
avg.length 13.441781435
max.length 21.00000000
utilization of channel .7156782
destroyed packets = 611763
max.wait = 5.12000012985664
count = 580729.0000000 "mean inactive time
count1 = 492975.0000000 "mean active time

Results Calculation:

First Source

Total number of cells lost due to first source = 312550
Total number of cells lost in burst length 2 = (11699)*(2) = 23398
Total number of cells lost in burst length3 = (18844)* (3) =56,532
Total number of cells lost in buffer length4 = (47691)*(4) = 190764
Therefore total number of isolated cells lost = 41865
Total blocking probability = total cells lost/total number of cells transmitted
Total blocking probability = 611763/1500000=.407
Similarly the calculations are performed for **second source**

Output3:(Bursty Sources = 3)

These are the results for three sources and utlisation = .96% and activity factor = .45 ,Interarrival time = .349759

These are results for cell loss (each source)
241807 241064 226074 0 0 0
These are results for cell loss (burst length) first source
15122 2566 25195 0 0
These are results for cell loss (burst length) first source
15122 2566 25195 0 0
These are results for bursty length cell loss(second source)
16351 2943 22543 0 0
these are results for source number 3
15112 2819 20505 0 0
These are results for source number 4
0 0 0 0 0
These are results for source number5
0 0 0 0 0
These are results for source number6
0 0 0 0 0
number of arrival 2708965.00000000
number of departures 2000000.000000
avg.wait 2.9707183
avg.length 8.504402743
max.length 21.00000000
utilization of channel .7328562
destroyed packets = 708945
max.wait = 5.11999992589699
count = 1148645.0000000
count1 = 948338.0000000

Results:

Total Blocking Probability 708945/2000000 = .3544725

Output4:(Bursty Sources = 4):

These are results for 4 sources utilisation = .96% and activity.factor = .45
Inter arrival time = .4663338

These are results for cell loss (each source)
131126 121057 126572 132097 0 0
These are results for cell loss (burst length) first source
8647 8357 8393 0 0
These are results for bursty length cell loss(second source)
8220 7173 7112 0 0
these are results for source number 3
8047 7387 8023 0 0
These are results for source number 4
9064 8055 8041 0 0

These are results for source number5
0 0 0 0 0
These are results for source number6
0 0 0 0 0
number of arrival 2510872.00000000
number of departures 2000000.000000
avg.wait 3.6103943
avg.length 10.870421962
max.length 21.00000000
utilisation of channel .7707771
destroyed packets = 510852
max.wait = 5.12000004507718
count = 1486881.0000000
count1 = 1170643.0000000

Result

Total blocking probability = 510852/2000000=.2554260

Output 5:(Bursty Sources = 5)

These are the results for 5 sources utilisation = 96% inter arrival time = .5829
% a.out
These are results for cell loss (each source)
41483 42285 48703 57279 55580 0
These are results for cell loss (burst length) first source
2169 393 2085 0 0
These are results for bursty length cell loss(second source)
2333 242 1876 0 0
these are results for source number 3
2746 354 2364 0 0
These are results for source number 4
3661 488 3288 0 0
These are results for source number5
3600 522 3287 0 0
These are results for source number6
0 0 0 0 0
number of arrival 1245346.00000000
number of departures 1000000.000000
avg.wait 2.7454093
avg.length 8.663392867
max.length 21.00000000
utilisation of channel .8078212
destroyed packets = 245330

Results Calculation:

TOtal blocking probability = .24533

Output 6:(Bursty Sources = 6)

These are results for 6 sources ,with utilisation = 0.98% and activity factor = .45,Interarrival time = .7 milliseconds
% a.out

These are results for cell loss (each source)
33984 33410 40536 32894 30319 34488
These are results for cell loss (burst length) first source
2962 1539 1207 0 0
These are results for bursty length cell loss(second source)
2626 1206 1008 0 0
these are results for source number 3
3375 1945 1424 0 0
These are results for source number 4
2833 1433 984 0 0
These are results for source number5
2521 949 876 0 0

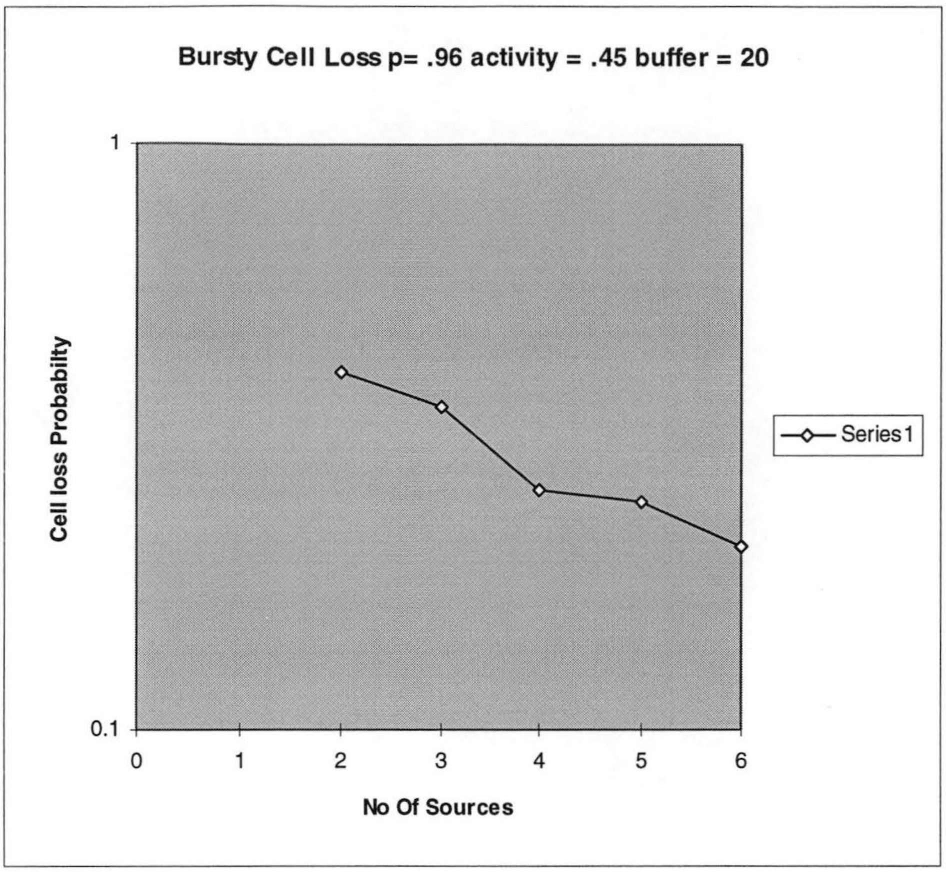
number of arrival 1205650
number of departures 1000000
avg.wait 3.2735240
max.length 21
utilisation .8269
destroyed packets = 205361

Results Calculation:
Total blocking probability = 205361/1000000 =.2053610
All the above results are tabulated and the graph is plotted

Table 10
No of Sources/Cell loss
probability

2	0.407
3	0.354473
4	0.25542
5	0.24533
6	0.205

Graph10
Scale : Log



Observation 4:

From the above graph it is clear that as we move towards leftside on x-axis that is decreasing the number of sources ,the cell loss probability increases as the Inter arrival time is less(Utilisation kept constant) ,therefore more average bit rate put on to the channel,that is more saturation of the channel in mean active time period and hence the high cell loss

Conditional Cell Loss(Burst length Probabilities):

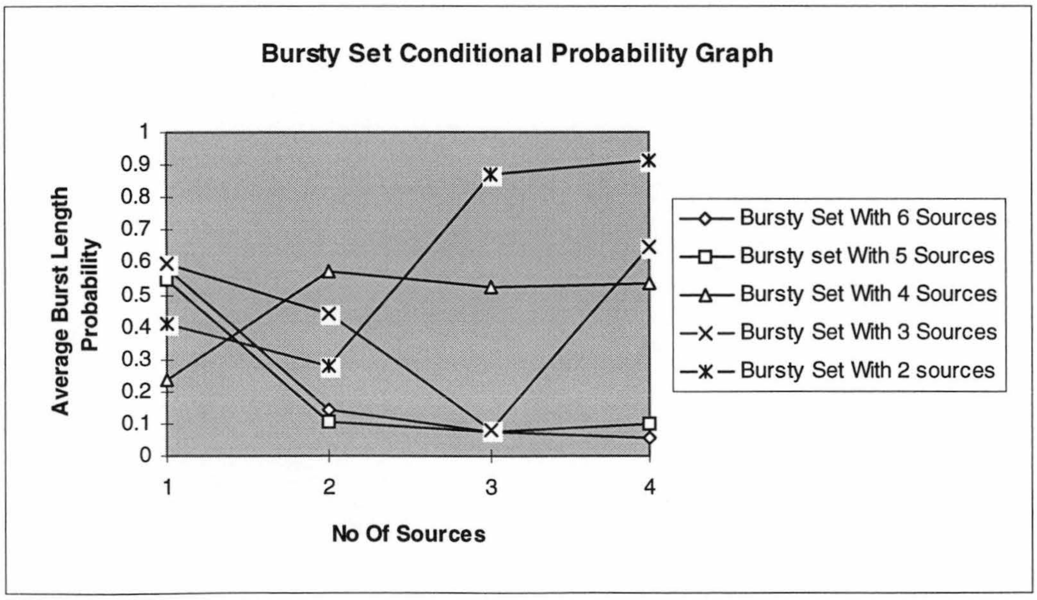
From the above output files ,the average burst length probabilities are calculated for sources 2 to 6 and the graph is plotted.

Table
X = No Of Sources
Y1 = Average Burst Length Probability (Bursty Set With 6 Sources Utilisation = 96%)

Y2 = Average Burst Length Probability (Bursty Set With 5 Sources Utilisation = 96%)
Y3 = Average Burst Length Probability (Bursty Set With 3 sources Utilisation = 96%)
Y4 = Average Burst Length Probability (Bursty Set With 2 Sources ,Utilisation = 96%)
Table11

X	Y1	Y2	Y3	Y4	Y5
1	0.581	0.544	0.2375	0.597	0.411
2	0.145	0.10858	0.571	0.439	0.2826
3	0.07279	0.0748	0.5207	0.078611	0.87
4	0.05445	0.09654	0.5367	0.644	0.91

Graph11:



Observation:

From the above graph it can be observed that Bursty Sets with sources 5 and 6 have a smooth decreasing slope ,this is expected since as the sources are more the Inter arrival times are large and the link getting saturated in the mean active periods is less .In other the average bit rate is less.If the sources are few say 1to 3 but utilisation of the link is same as large sources ,the Interarrivl times in this case are small (note that service time is constant in this case .256 milliseconds) .where as the Inter arrival time may be .233 milliseconds(assuming link utilisation is 96%),however the link is more than saturated (.256/.233) ,in other words for that period it is in instable state,however the next consecutive silence period may be small(exponential distribution) and then again an active period (bigger)may start (exponentially distributed) . ,this makes the queue to switch between momentary stable and instable states ,and this may be the reason that for smaller sources average burst length probability for 3,4 or more Burst lengths is more than 1 and 2 Burst lengths.This explains the clinks (irregularity) in the graph for smaller sources

Thus if such a real world environment exists the cell loss is heavy and the quality decreases.

Bibliography And References

1.Gross D.Harris., *Fundamentals Of Queuing Thoeory.*, Wiley 1974.

2.Gradshteyn I.,*Tables Of Intetgrals,Series And Products*.,Academic Press

3.Jain R.,*Congestion Control In Computer Networks: Issues And Trends,*may 1990,IEEE Network Magazine.,pp 24-30

4.Kleinrock L., QUEUEING SYSTEMS.VOL1.John Wiley and Sons 1975

5.Ramaswami V.,Willinger,W.,*Efficient traffic Performance For Packet Multiplexers.,*Proceedings Of ITC Specialist Seminar Adelaide,1989

6.Stallings W.*Data And Computer Ccommunications(Macmillan)* 1988

7.Ramesh Nagarajan,James F.Kurose,and Don Towsley Loss in *Finite-buffered Voice Multiplexers IEEE selected areas in communications april 1991*

8.Moshen A.Saleh,Ibrahim W.Habib, and Tarek N.Saadwai *Simulation Analysis Of A Communication Link With Stastically Multiplexed Bursty Voice Sources* IEE journal on selected areas in commun apr 1994

9.Sriram and Whitt *Characterising Superposition Arrival* , *Processes* IEEE Journal on selected ares in commun sept 1986

10.Telecom Australia, *A course in Teletraffic Engineering.,1978*

11.Wolfram S., Mathematica:*A System for Doing Matheamatics by computer Addison-Wesley,1991*

12.Stallings W .,*ISDN and Broadband ISDN.,* Macmillan

13.Haykin S., Digital Communications.,John Wiley and Sons 1988

14.Nagle.J .,On Packet Switches With bInfinite Storage., IEEE Transactions on Communications april-1987 pp435-438

15.Heffes H,Lucantoni D.,A Markov Modulated characterisation of packetized Voice and Data Traffic and Related Statistical Multiplexer Performance IEEE Journal On Selected Areas in Commun,sept 1991

16.RUSELL E., Building Simulation Models with Simscript Π.5 CACI 1983

17. Hong D., Suda T., Congestion Control and Prevention in ATM Networks .,July 1991 pp 24-30

18.Wernik M.,Aoul-Magd O.Gilbert H.,*Traffic Management for B-ISDN*

19.de-prycker M., Asynchronous Transfer Mode .,Solution for Broadband ISDN .,Ellis Horwood 1991

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

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

22.Hirmatsu A.,Integration Of ATM call Admission Control and Link Capacity Control by Distributed Neural Networks., IEEE - Journal On selected Areas in Communications,vol9,september 1991 pp-1131-1138