Queen Mary
University of London

Traffic control mechanisms with cell rate simulation for ATM networks.

Freire, Fonseca Paula, Christina,

Traffic Control Mechanisms with Cell-Rate Simulation for ATM Networks

by

Paula Cristina Freire Fonseca

Submitted for the Degree of Doctor of Philosophy

Department of Electronic Engineering

Queen Mary and Westfield College

University of London

April 1996

Abstract

This thesis considers the introduction of space priority mechanisms in Asynchronous Transfer Mode (ATM) networks, as a means of optimising network utilisation and evaluates the impact of controlling traffic through Usage Parameter Control (UPC) mechanisms.

The work focuses on the Partial Buffer Sharing (PBS) mechanism with two levels of priority, which only accepts low priority cells into an input buffer while its occupancy is below a given threshold and it shows that a network's throughput can be improved by utilising priority mechanisms. It uses a cell-rate simulator where an approximate version of PBS is implemented, which assumes that the queue size above the threshold is zero. This approximation simplifies the implementation and only affects results for high priority traffic.

An exact fluid-flow analysis for a single On/Off source feeding an ATM buffer has also been modified to take priorities into account. The exact analysis has been applied, both to the approximated version of PBS implemented in the cell-rate simulator and to the exact PBS mechanism. Results from simulation and analysis indicate good agreement between the two approaches.

UPC mechanisms aim at protecting network resources and to this end, they take actions on any misbehaving traffic by discarding or tagging violating cells. Tagging cells implies changing high priority cells into low priority cells, which will affect the Quality of Service (QoS) of traffic with priorities. This thesis investigates several possible scenarios for deciding when tagging or discarding of violating traffic should take place. The implementation of the different scenarios again makes use of the prioritised cell-rate simulator.

The thesis also includes an application of the work to a real ATM network. Traffic experiments are performed which aim at validating the prioritised cell-rate simulator, using the simulator to investigate priorities and policing.

Contents

Tables and Figures

Acknowledgements

To my parents, for their love and support.

First and foremost, I would like to thank my supervisor, Prof. Laurie Cuthbert, and Dr. Jonathan Pitts, for their encouragement and help during the course of this research work.

I would also like to express my thanks to all the people (now good friends) that have worked at the Telecommunications Research Group in the Department of Electronic Engineering (QMW) since 1993, for their friendship and understanding.

A big THANK YOU is also due to Andy Martin, for his help with the software debugging and for being my best friend ever.

Finally, I would like to thank Centro de Estudos de Telecomunicações (CET) - Portugal Telecom for giving me the opportunity of coming to QMW and take a PhD.

The author gratefully acknowledges support during the course of this work from the institution Junta Nacional de Investigação Científica e Tecnológica (Portugal) under Grant no. CIÊNCIA/BD/2188/92-IA from December 1992 to November 1995, and under Grant no. PRAXIS XXI/BD/5464/95 from December 1995 to April 1996.

Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
BER	Bit Error Ratio
B-ISDN	Broadband Integrated Services Digital Network
CAC	Connection Admission Control
CAD	Computer Aided Design
CBR	Constant Bit Rate
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CDV	Cell Delay Variation
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CSLB	Continuous-State Leaky Bucket
DTMC	Discrete-Time Markov Chain
DMAP	Discrete-Time Markovian Arrival Process
DOS	Disk Operating System
EWMA	Exponentially Weighed Moving Average
FEC	Forward Error Correction
FIFO	First-In First-Out
GCRA	Generic Cell Rate Algorithm
GFC	Generic Flow Control
GMDP	General Modulated Deterministic Process
GoS	Grade of Service
GPSS	General Purpose Simulation System
HDTV	High Definition Television
HEC	Header Error Check
HOL	Head-Of-the-Line
HOL-PJ	HOL with Priority Jumps
IDP	Interrupted Deterministic Process
IPP	Interrupted Poisson Process
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
ITU-TS	International Telecommunication Union - Telecommunications Standards
JW	Jumping Window

LAN	Local Area Network
LHS	Left Hand Side
MAP	Markov Arrival Process
MMBP	Markov Modulated Bernoulli Process
MMDP	Markov Modulated Deterministic Process
MMPP	Markov Modulated Poisson Process
MPEG	Moving Picture Coding Experts Group
NE	Network Element
NNI	Network Node Interface
NT	Network Termination
OAM	Operation and Maintenance
OSI	Open Systems Interconnection
PBS	Partial Buffer Sharing
PT	Payload Type
QMW	Queen Mary and Westfield College (University of London)
QoS	Quality of Service
RACE	Research into Advanced Communications in Europe
RBLB	Rate Based Leaky Bucket
RESTART	REpetitive Simulation Trials After Reaching Thresholds
RHS	Right Hand Side
RNG	Random Number Generator
SDH	Synchronous Digital Hierarchy
STM	Synchronous Transfer Mode
TCP	Transmission Control Protocol
TJMW	Triggered Jumping Moving Window
TV	Television
UBR	Unspecified Bit Rate
UDP	User Data Protocol
UNI	User Network Interface
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC/VP	Virtual Channel/Path
VCI/VPI	Virtual Channel/Path Identifier
VSA	Virtual Scheduling Algorithm

1. Introduction

This thesis describes work on space priority mechanisms for traffic in Asynchronous Transfer Mode (ATM) networks and their relation with the ATM control function, Usage Parameter Control (UPC). In particular, the space priority mechanism Partial Buffer Sharing (PBS) has been implemented in a cell-rate simulator, confirming the improvement of a network's throughput when using priorities. An analytical model is also extended to cater for traffic with priorities and results from both simulation and analysis show good agreement. This is the new work in the area. UPC is then applied to several environments to identify the best action (either to tag or discard cells) to be taken by the policing function when a network encounters traffic that does not respect its own declared parameters. Finally, the thesis presents an application of the work to a real ATM testbed. Traffic experiments are carried out to validate the cell-rate simulator and thus determine its utility to help predict the behaviour of a real ATM network when traffic with priorities is taken into account.

ATM has been agreed to be the transfer mode for Broadband Integrated Service Digital Networks (B-ISDN) (see [ITU90c]). It is a technique that enables information to be transferred across a network asynchronously with its arrival at the network; this is done by mapping the information sent by users onto *cells* (see [Cuth93, pp.2-5]). This transfer mode is said to be *asynchronous* because the cells that contain the information from a user will not necessarily appear in the network at time intervals directly related to the user data.

ATM can be used in a wide range of environments and it has been designed to support a great variety of services, with different Quality of Service (QoS) requirements. However, this flexibility implies addressing the areas of resource management and traffic control in order to achieve a specific network performance. In particular, the resource management mechanisms *cell-level quality control* (priority control) and *congestion control* must be considered to optimise the utilisation of network resources. *Priority control* refers to the introduction of priorities in ATM cells, with the objective of determining which cells are considered less important (and therefore, most probably lost in a situation of congestion in the network). The assignment of priorities can be done either,

implicitly ..... using separate identifier pairs, called Virtual Path/Channel Identifiers (VPI/VCI), for each priority level; in this case, all the cells in a connection will have the same priority, *or*

explicitly ..... using the Cell Loss Priority (CLP) bit in the cell header, which may be set by the user (through the terminal equipment), with CLP=0 referring to a cell with high priority, while a low priority cell has its CLP bit set to 1; in this case, it will be possible to have different cells in the same connection with different priorities.

It has been verified (see [Krön90]) that the use of implicit priorities (although the simplest to implement) usually leads to a poorer performance than explicit priorities. On the other hand, *congestion control* consists of a set of actions related to the selective discarding of cells by taking into account their priority level. In a situation of buffer overflow, low priority cells will be discarded first. Nevertheless, the QoS requirements are still met for low and high priority traffic.

Before a source can start sending data through an ATM network, it must (at the call set-up phase) provide information to the network to characterise its traffic; this information is given in the form of traffic parameters and is called the *traffic contract*. The function that acknowledges the traffic characteristics of a source and thereafter decides whether to accept or reject its connection, while still guaranteeing the QoS of other connections already established in the network, is called Connection Admission Control (CAC). The traffic characteristics given by the source at call set-up are then used by the UPC function to monitor the traffic being generated and protect the network resources from misbehaving traffic. This thesis also considers the effect of controlling traffic with priorities by using UPC mechanisms. The reason for this study comes from the fact that the actions (like tagging) taken on misbehaving traffic may affect the QoS of traffic with priorities that is respecting the traffic contract.

When a real system (e.g., a link in an ATM network) needs to be studied, three techniques are usually available: analysis, simulation and experiments. The first solution makes use of mathematical methods (such as probability and queuing theory) and gives exact information about the system, provided the system is simple. However, as the system's complexity begins to increase, known analytical techniques become too difficult to evaluate. This is when resort must be made to simulation. With this technique, a model of the system is evaluated by using numerical methods. Simulation is thus a valuable method in that it helps to reproduce the behaviour of complex real systems and it can also help predicting the behaviour of a modified real system without being necessary to previously change the system itself. Since there are no real ATM networks fully operating yet, experiments are not necessarily realistic and simulation techniques have therefore, been extensively used in ATM studies. This thesis makes use of cell-rate simulation, which operates at burst level and has been shown in [Pitt93] to provide a good comprise between processing speed and accuracy, making it more suitable than cell-level simulation in many applications.

This chapter has introduced the focus of the thesis and the main areas of study that needed to be addressed. Chapter 2 reviews the origin and evolution of Broadband Integrated Services Digital Networks (B-ISDN), the types of services that this kind of network can support and the protocol reference model agreed by the Comité Consultatif International Télégraphique et Téléphonique (CCITT) for B-ISDN. CCITT has recently changed its name to International Telecommunication Union - Telecommunications Standards (ITU-TS); the shorter acronym ITU will be used throughout the text to refer to CCITT, even in the reference to documents that were issued when CCITT had not yet changed its name. The same chapter also refers to some issues that remain to be standardised in B-ISDN networks in order to make them a reality.

Chapter 3 gives an overview of ATM related concepts and traffic issues, as well as its context in B-ISDN networks. A particular focus is on the various priority mechanisms for ATM networks that have been devised so far and how they perform. This part of the thesis also gives a detailed description of the concept of UPC and the mechanisms suggested to police traffic in ATM networks. The chapter ends with a brief summary of the views on ATM and its standards from the ATM Forum, another organisation that has been working towards the definition of standards.

Modelling is used in ATM networks to describe the traffic behaviour of sources. This subject is dealt with in Chapter 4, where several traffic models are presented and examples are given of real sources and associated models used to characterise them. Among these source models, two in particular are of special interest, as they are used throughout the thesis to carry out traffic experiments and validate the developed software tool referred to before: the *On/Off traffic model* and the *General Modulated Deterministic Process* (GMDP) model. Traffic source modelling is closely related to simulation, which has been the platform used to develop the present research work, since it is still not easily possible to collect results from a real ATM network, especially if it is required that they take traffic with priorities into account. Chapter 4 also addresses simulation methods; the main focus is on the so called *accelerated simulation techniques*, of which the cell-rate simulation is an example.

The model for PBS and respective implementation in the cell-rate simulator (see [Fons94a]) are described in Chapter 5. This includes a brief description of the original cell-rate simulator and is followed by the presentation of the validation process (see [Fons94b]), which compares previously published results (from other researchers) with the ones obtained from the prioritised cell-rate simulator. In this Chapter, the new work developed by the author comprises the implementation and validation of the PBS priority mechanism in the cell-rate simulator.

The prioritised simulator was also validated against a fluid flow analytical model (see [Fons95a]), developed in [Scho94]; this is presented in Chapter 6. The analysis considers a traffic source modelled as an On/Off source with *On* and *Off* periods Geometrically distributed. This model has been extended in the thesis in order to study traffic with priorities (see [Fons95b]). The chapter also explains the concept of fluid flow approximation. The extension of the fluid flow model to the case of priorities and subsequent comparison against the prioritised cell-rate simulator represents new material developed by the author.

Chapter 7 is dedicated to UPC and its impact on traffic with priorities and it starts by describing in detail the implementation process of UPC in the prioritised cell-rate simulator (used especially when tagging is the chosen UPC action on violating traffic). Next, the scenarios studied are described and some results comparing the effect (in terms of cell loss probability) of taking particular actions on misbehaving traffic are used to derive some conclusions on how UPC should be applied. The implementation of several policing scenarios in a cell-rate simulator represents new work in the field, since (so far) policing has only been implemented at cell level.

An application of the developed work is given in Chapter 8, where an ATM testbed is used to carry out traffic experiments which aim at validating the cell-rate simulator without priorities. The priority mechanism PBS has also been emulated in the testbed, thus enabling to obtain results concerning experiments using traffic sources of low and high priority. Results from the simulator and the testbed are compared.

Finally, Chapters 9 and 10 conclude the thesis by assessing the work developed and presenting possible extensions of the research carried out.

2. Broadband Integrated Services Digital Networks

The main characteristic of ISDN is its capability of supporting both voice and non-voice applications in the same network; with ISDN networks, existing services and new services can be provided in an integrated manner.

ISDN networks have two interfaces: *basic access*, which consists of two 64 kbit/s channels and a 16 kbit/s channel for signalling, and a *primary rate access*, with a 1.544 Mbit/s channel (or a 2.048 Mbit/s channel) plus a 64 kbit/s signalling channel. This type of network is sufficient for services like telephony. However, services like video and interconnection of Local Area Networks (LANs) require greater bandwidths, which has led to an effort in trying to establish standards for an integrated broadband communications network (the B-ISDN network) that consists conceptually of a number of customer premises and a network of nodes, links and switches that connect those premises. The developments on B-ISDN that are still going on aim at achieving a world-wide network topology that will facilitate the exchange of information between subscribers.

The next sections will present some of the features of B-ISDN networks, its protocol reference model and issues that still remain to be standardised.

2.1. Types of Services

The concept of *broadband* is defined in ITU's Recommendation I.113 [ITU91] as being “... a system that requires transmission channels capable of supporting greater rates than the ones provided by the primary access rate ...”; at present, the B-ISDN interfaces support up to 622 Mbit/s. It should be noted that there are already broadband communication systems in operation: LANs, which work at 10 Mbit/s. However, there is no concept of integration in this type of networks, i.e., customers must use a different network if for example they want to use a telephony service.

What is a B-ISDN network? This question can be briefly answered in one sentence, by saying that it is a connection-oriented cell-switching network, which has adopted ATM as its transfer mode. A summary of B-ISDN concepts can be found in

ITU Recommendation I.121 (see [ITU90c]), from which a brief summary has been taken to give an overview of the possibilities offered by such a network:

* support of switched, as well as permanent, point-to-point and point-to-multipoint connections;
* support of on-demand, reserved and permanent services;
* support of connections from services in circuit and packet mode, of mono and multimedia type, of connectionless or connection-oriented nature and in bidirectional and unidirectional configurations;
* support of services with constant and variable bit rates.

Before giving examples of applications and services for B-ISDN, it is worthwhile defining the meaning of each concept. A *service* is what the customer pays for and the way in which he makes use of a service is an *application*. Broadband applications have been divided by ITU (see [ITU90b]) as follows:

- interactive services
    - *conversational*
    - *retrieval*
    - *messaging*
- distribution services
    - *without user-individual presentation control*
    - *with user-individual presentation control.*

Conversational services include services for which a real time end-to-end information transfer is needed and the information flow can be bidirectional or unidirectional; this category of services is the one that more closely corresponds to the services in current telecommunications networks. Examples belonging to this category are telephony, videotelephony and high-speed data transmission.

Video and document retrieval services (and generally, any multimedia archive) are characterised by allowing their users to retrieve, at request, information that has been stored elsewhere in the network. These services are called *retrieval services*.

When services do not demand a real-time operation and the communication between users is done via storage units with store-and-forward or mailbox functions, they are referred to as *messaging services*. The video mail service (integrating the current electronic mail and phone messages) is a typical example.

Distribution services without user-individual presentation control provide a continuous flow of information distributed from a central location to a number of authorised users, which have no control over the starting time or the order of presentation. One common example is broadcast television (TV). On the other hand, *distribution services with user-individual presentation control* provide broadcast services in which the users can select individual information and control the start and the order of the information. The information distributed in this fashion includes text, graphics, audio and still images, as well as mixed documents that integrate more than one of these services; teletext and video on demand are such examples.

More examples of B-ISDN services and their features can be found in [Cuth93, pp.37-47] and [Onvu94, pp.2-6].

2.2. The B-ISDN Protocol Reference Model

The ambitious aim of B-ISDN to support a great variety of services and applications can only be met through a complete standardisation. The first step towards this objective was taken with the first published recommendation on the subject by ITU (see [ITU90c]), followed by recommendations that addressed the fundamental principles and initial specifications for B-ISDN. The protocol reference model shown in Fig.2.1 was part of those recommendations; its structure is similar to the Open Systems Interconnection (OSI) model developed by the International Organisation for Standardisation (ISO) for use in computer networks with a view to establish standards allowing computers to communicate.

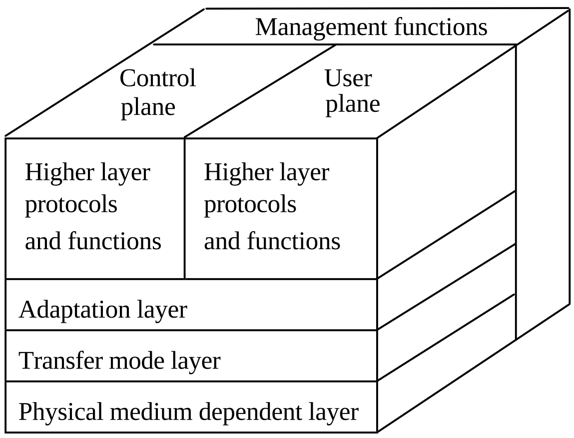


Fig.2.1 - The protocol reference model for B-ISDN.

Since then, all B-ISDN standards have been developed by taking that protocol into account. It is therefore worthwhile explaining briefly what is the function of each part of the protocol.

Bit timing, network clock and maximum bit-error rate are some of the services provided by the *physical layer*; it represents the underlying transport of the network. The *adaptation layer*, as well as the *transfer mode layer*, is used by the control plane and user plane to convert data units into cells. The *transfer mode layer* defines the way in which the physical layer is to receive the information coming from the higher layers; the *adaptation layer* provides adaptation functions like connectionless and packet-mode services. The set-up and release of connections are controlled by the *control plane*, while the *user plane* is used to transmit data coming from the user, when a connection is established. The way in which the *management functions* (which provide network supervision functions) relate with the two higher planes is also defined by the protocol.

2.3. B-ISDN Open Issues

Section 2.1 presented the wide range of services and applications that B-ISDN networks propose to support. One or more of the following attributes characterise that multitude of services,

* high bandwidth;
* bandwidth on demand;
* varying parameters for QoS;
* point-to-point/multipoint or multipoint-to-multipoint connections;
* constant or variable bit-rate services;
* connection oriented (e.g., telephony) or connectionless services.

This implies that B-ISDN networks must be able to perform tasks such as,

* assign usable capacity dynamically on demand;
* switch all types of services;
* take the bursty nature of certain services into account when allocating the available bandwidth.

The problem of providing high bandwidth to B-ISDN networks has been solved by introducing reliable fibre systems into the access network. Nevertheless, other problems remain to be dealt with; for example, it is still not clear if existing transport protocol such as the Transmission Control Protocol (TCP), which was designed to provide reliable data delivery between two end nodes, are adequate for B-ISDN services or if another protocol will have to be designed. Other areas that need to be further studied are traffic control and congestion control. In fact, B-ISDN networks will support the existence of several connections (with different QoS requirements) simultaneously; thus, the simple call admission mechanism that exists in the current telephone network cannot be used anymore and will be extended to a more complex Connection Admission Control.

2.4. Conclusion

This Chapter provides an introduction to the need for *broadband*. It also introduces the protocol reference model used to visualise the layered functions in a B-ISDN network. Finally, it defines (in a general way) the classes of services that this type of network aims to provide and it emphasises the fact that there are still some technical/conceptual issues to be solved.

The (brief) description of the protocol reference model helps to put in context the layer in which this thesis concentrates: the *ATM layer*.

On the matter of the issues still to be solved, reference is made to the necessity of further effort being put into the study of both traffic and congestion control methods. This is the main focus of the present thesis, which addresses a congestion control mechanism, *priority control*, and a traffic control mechanism, *policing*.

3. Asynchronous Transfer Mode

It was noted in the last chapter that ATM is the transfer mode chosen for B-ISDN networks; this means that ATM will define the way in which information from users is matched onto the physical network. With ATM, information from users is transmitted using packets of fixed size, called *ATM cells*. These are 53 octets long, with an *information field* (the cell payload) of 48 octets and a *header* (containing network information, such as routeing) of 5 octets (see Fig.3.1). Unlike the conventional Synchronous Transfer Mode (STM) that switches data according to a position in the recurrent structure (i.e., a frame), the switching of ATM cells is executed by using labels in the header of cells that contain routeing information.

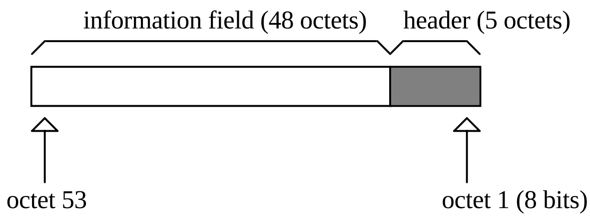


Fig.3.1 - Format of an ATM cell.

A closer look at the cell header (which is used to route cells between switches) shows its different functional parts (see Fig.3.2). In brief, the *Generic Flow Control* (GFC) *field* is reserved to indicate congestion (although it has not been standardised yet), while the *VPI field* gives a high level of routeing for ATM cells, representing a set of VC connections; the *VCI field* gives a low level of routeing, representing one single VC connection. The type of data contained in ATM cells (e.g., maintenance cells) is identified by the *Payload Type* (PT) *field*, while one of the functions of the *Header Error Check* (HEC) *field* is to execute a checksum over the cell header for bit error detection/correction in the header. Finally, the *Cell Loss Priority* (CLP) *bit* can be used to tell the network that a cell is or is not considered less important, in which case, when there is loss of cells, the "less important" ones are lost first. This field can be used either by the terminal or when the cell is found by the UPC function not to be respecting its traffic contract. One last remark should be made about the fact that the cell header's structure shown in the figure below is used only at the User Network Interface (UNI), the point at which users have access to the network; at the Network Node Interface (NNI) - i.e., between network nodes -, a similar structure is used, but no GFC field exists and instead, the size of the VPI field is increased.

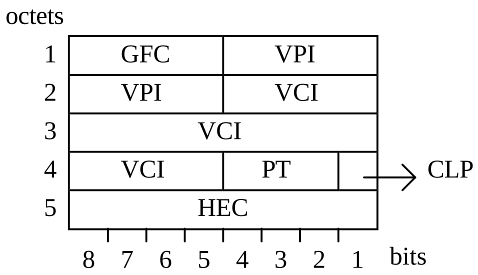


Fig.3.2 - Structure of an ATM cell's header.

In an ATM based network, the information is transferred in an asynchronous way with its arrival at the system input. When information arrives, it is put into buffers until there is enough data to fill an ATM cell, after which it is transported through the network. When there is no information to be transmitted, an unassigned cell will be transmitted (see [Cuth93, pp.2-5]).

The rest of this chapter will give an overview of ATM, as all the work described in this thesis is directed to be applied to ATM based networks; it will start however with a reference to the transfer modes currently in use and their relation with ATM and it will then look at subjects like priority mechanisms, traffic control and traffic measures.

Recently, another organisation involved in the definition of standards for ATM based networks (the ATM Forum, formed by a consortium of vendors), has defined concepts such as a new service type, the Available Bit Rate (ABR) service. This chapter also reports on some of the work carried out by the ATM Forum and its possible consequences in the global process of standardisation.

Additional information on ATM and ATM related subjects can be found for example in [Gilb91], [Hong91], [Izma93], [Okad91], [Pryc91] and [Robe92].

3.1. Current Transfer Modes and ATM

A *transfer mode* is a technique used for transmission, multiplexing and switching aspects of communication networks. Currently, the following types of networks exist, if considered according to the transfer mode they use (see [Port94]):

* *circuit-switched networks* (e.g., telephone networks)

  In this case, a circuit is established for the whole duration of a connection between two entities wanting to exchange information. Each channel has a fixed bandwidth and it is possible to multiplex channels onto a link; once the circuit has been established, traffic will flow continuously for the complete duration of the connection. For this type of networks, the transfer mode minimises the connections' end-to-end delay; the total delay can be expressed by,

$$td \approx pd + \sum_{i=1}^{n} (st)_i , \quad n \in N \qquad (3.1)$$

n = number of intermediate nodes
td = total delay
pd = propagation delay
st = switching time

* *message-switched networks* (e.g., data networks)

  The message switching mode considers each information unit as a message transmitted in the network, independently from other messages; this is possible by the addition of a header to each message that indicates the destination node. After receiving a message, each intermediate node stores it until it is processed and then transmits it. The processing of the message consists of finding out which is the next node that the message should be sent to, by looking at the header information field.

* *packet-switched networks* (e.g., X.25 networks)

  The packet switching mode tries to combine the qualities of the last two transfer modes, by operating as a message mode but with the size of the message not being greater than a few thousand bytes; thus, some messages may have to be partitioned into packets before transmission, which means that it is possible to have overlapping of reception and transmission. Similarly to the message-switching mode, each packet is treated independently; the difference is, all the packets belonging to a message have to be reassembled at the receiver end, so that the original information unit can be formed and then passed to the user. Packet streams are handled in one of two ways:

- *datagram packet switching*

  In this case, each packet is again treated independently but it can take different paths to arrive at its destination; therefore, it is possible for packets from the same message to arrive out of the original sequence.

- *virtual-circuit packet switching*

  For this type of switching, logical end-to-end connections can be established in a similar way to the circuit switching mode, before the transmission can start. All the packets from a message have the same size and may follow the same path, which guarantees sequencing, but a call set-up phase is still necessary.

All the transfer modes that have just been described could be candidates to be the chosen one for B-ISDN networks. However, the B-ISDN transfer mode must have the properties:

* to support all known services and to support future services whose characteristics are still unknown;
* to use network resources efficiently;
* to minimise the switching complexity;
* to minimise the processing time and the number of buffers at intermediate nodes;
* to guarantee performance requirements of existing and expected applications.

Therefore, *circuit switching* is not adequate for B-ISDN networks, where both constant and variable bit rate services can coexist, due to its required fixed bandwidth per connection. On the other hand, for *packet switching*, the problem lies in the fact that the transmission of a message requires extra information, given by the packet header, when compared with *message switching*.

Although the *packet switching mode* could be used in B-ISDN networks, ATM has been chosen to be the transfer mode for B-ISDN, as it combines the best from the circuit and packet switching modes; this can be seen by looking at some basic features of ATM (see [Sait94, pp.3-5]):

- blocks of fixed length, called *ATM cells* (as it was seen in the beginning of this chapter), carry the information;
- the header of ATM cells contains a Virtual Path/Channel Identifier (VPI/VCI) pair that includes information about the routeing address, in order to allow network components to distinguish between diverse traffic flows; these labels are used in multiplexing and they represent, respectively, a unique identifier value for a *Virtual Channel* (VC) describing unidirectional transport of ATM cells and a unique identifier value for a *Virtual Path* (VP) consisting of several VCs that can be switched as a whole (see Fig.3.3);
- in an ATM based network, flow control and error recovery are performed on an end-to-end basis;
- cells are transported at regular intervals and are generated only when there is information to be transferred; idle periods carry unassigned cells;
- cell sequence is preserved;
- short cell lengths keep delay to a manageable value.

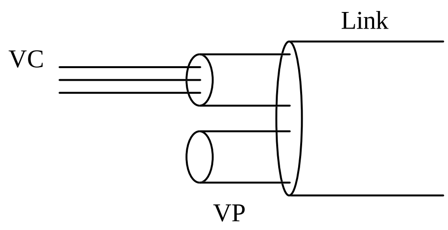


Fig.3.3 - Links, VPs and VCs.

This transfer mode can be described as a connection-oriented packet switching mode, which means that virtual-circuit packet switching is the most like ATM, which can be considered as a virtual circuit switching technique; in fact, before information from a user can be transmitted in the form of ATM cells, a path (comprising a sequence of VCs and VPs) to the destination has to be established first.

More detail on the transfer modes described in this section can be found in [Onvu94, pp.13-17] and [Pryc91, pp.26-96].

3.2. The ATM Protocol Reference Model

When the protocol reference model for B-ISDN (see Fig.2.1) is combined with the ATM concept, another protocol is obtained that emphasises the ATM role in broadband networks (see Fig.3.4).

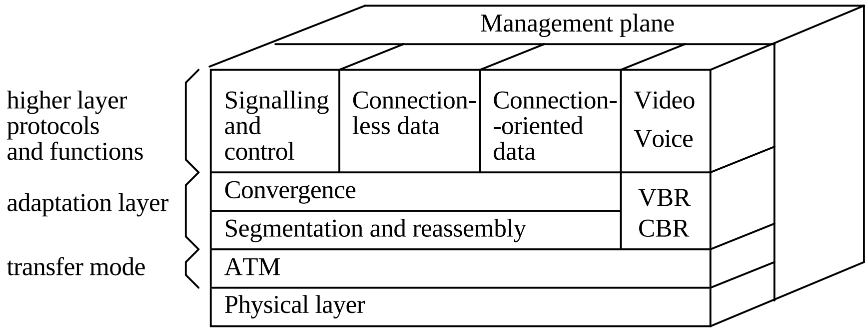


Fig.3.4 - The protocol reference model for ATM.

The transport of ATM cells between two ATM entities is a function of the *physical layer*, while cell multiplexing, cell demultiplexing and routeing functions using the VCI and VPI fields of the cell header are performed by the *ATM layer*, which is common to all types of services.

The *ATM Adaptation Layer* (AAL) provides the functionalities, through specific protocols (AAL1, AAL2, AAL3/4 and AAL5), required for each service class to reach its desired QoS. The information that the AAL receives from higher layers is segmented or put into ATM cells; a reverse process occurs when the information comes from lower layers. This layer also ensures that cell sequence is maintained for each source. Table 3.1 shows which protocol is used for each type of service.

Some of the functions of the Constant Bit Rate (CBR) part of the AAL are variable compensation of delay and cell assembly/disassembly. The functions of the AAL bursty data services include actions on lost cells (e.g., recovering of lost cells by transmission in most data applications) and segmentation of information units into cells, as well as the reverse process.

Similarly to the case of the B-ISDN protocol, the ATM protocol reference model is the reference point that helps the standardisation process. For more details on this protocol, see [Cuth93, pp.7-36] and [Onvu94, pp.19-33].

Service class	***Characteristics***	***Protocol used***
circuit emulation, CBR video (e.g., telephony)	CBR; connection-oriented; required timing relationship between source and destination	AAL1
VBR video and audio (e.g., compressed video)	VBR; connection-oriented; required timing relationship between source and destination	AAL2
connection-oriented data transfer (e.g., X.25* and frame relay)	VBR; connectionless; timing relationship between source and destination not required	AAL3/4
connectionless data transfer (e.g., Ethernet LANs)	VBR; connectionless; timing relationship between source and destination not required	AAL5

Table 3.1 - Adaptation layer protocols and service classes (*X.25 is the ITU packet level protocol for data terminal equipment).

3.3. ATM Traffic Classification

Until recently, sources generating traffic in ATM based networks were generally divided into two categories, according to their bit stream pattern: *Constant Bit Rate* (CBR) and *Variable Bit Rate* (VBR) traffic sources. The first type of sources produces a periodic bit stream at a constant rate (where the inter-cell interval is determined by the rate of the source) and they can be fully characterised by just their maximum bit rate (i.e., the peak bit rate). CBR services are usually sensitive to delays and delay variations; as a consequence, they require peak bandwidth allocation (see Section 3.4). On the other hand, it is much more difficult to characterise VBR sources, as they generate a continuous bit stream at varying rates; parameters usually used for this purpose include,

* *peak cell rate* (i.e., the maximum cell rate generated by a traffic source);
* *average number of cells in an active period* (i.e., a period in which cells are generated);
* *average time cycle* (i.e., the average time from the start of an active period to the start of the next active period);

∗ *average cell rate*;

∗ *burstiness* (i.e., the ratio between the peak and mean cell arrival rates).

Note however, that other definitions have been proposed to measure burstiness, such as considering the ratio of the standard deviation to mean of the interarrival intervals of the source's arrival process (see [Fros94]). The former definition of burstiness (although widely used) has the disadvantage of its accuracy depending on the interval length used for the measure.

The next table (Table 3.2) shows some examples of these two categories of traffic sources (CBR and VBR traffic sources) and Table 3.3 gives typical values for the peak and average bit rates of some applications (see [Mase91]).

Traffic Class	CBR	VBR
Examples	telephony, fax, data retrieval	video conference, TV, CAD, HDTV

Table 3.2 - Traffic sources classified into CBR and VBR.

Note that the peak bit rate values indicated in Table 3.3 do not consider the effect of using the standard for compression known as Moving Picture Coding Experts Group (MPEG); for example, when using MPEG, the peak bit rate of broadcast TV can become as low as 2 Mbit/s.

Application	***Average Bit Rate***	***Peak Bit Rate***
voice	64 kbit/s	64 kbit/s
video	20 Mbit/s	34 Mbit/s
HDTV	100 Mbit/s	136 Mbit/s
video conference	400 kbit/s	2 Mbit/s
interactive image	64 kbit/s	1 Mbit/s

Table 3.3 - Typical values for some traffic parameters.

Two other classes, *Available Bit Rate* (ABR) and *Unspecified Bit Rate* (UBR) have been added since (see [Bono95] and [Newm94]) to CBR and VBR source classes (which are now part of just one class, named *guaranteed traffic* class), mainly due to the influence of the ATM Forum. A traffic source is classified as ABR (e.g., file transfer and LAN interconnection) when it has no possibility of describing its traffic characteristics in a very precise manner, giving however some minimum traffic requirements such as the minimum cell rate, which means that it will be sensitive to loss. On the other hand, an UBR service (e.g., a talk session in a computer that uses the User Data Protocol (UDP)) does not provide any descriptive traffic characteristics; consequently, this type of service does not get any QoS guarantee from the network.

For the *guaranteed traffic*, the network must assure very stringent QoS requirements. The objective of introducing the ABR and UBR traffic classes (which are not sensitive to delays) is to utilise dynamically the bandwidth unused by the guaranteed traffic, therefore maximising the usage of the bandwidth available.

The need for a new classification of traffic classes is the result of the increasing importance of data traffic in communications networks. With these traffic classes, it is possible to accommodate in one network the so called *socialist traffic* (predominant in private networks such as data communications, where the bandwidth is shared), as well as the *capitalist traffic* (typical of public networks such as telecommunications, where the bandwidth is negotiated) (see [Bono95]). Note however that this thesis only uses guaranteed traffic; this suggests that future work should be undertaken in order to test the research work developed against the presence of ABR and UBR traffic types in ATM networks.

Introducing new traffic classes implied the development of traffic control mechanisms that can handle them. In particular, the following methods were developed to be applied on ABR traffic:

- *Credit Based Scheme*

  This method regulates the input traffic on a switch on the basis of the buffer occupancy of the receiving switch, which decides on the number of cells that the source can send. This number is called a *credit number* and it represents an authorisation for the source to transmit cells. The calculation of the credit number aims at avoiding loss in the receiving buffer.

- *Rate Based Scheme* (or Backpressure mechanism)

  This is the recently adopted traffic control method for ABR traffic and it provides an end-to-end congestion avoidance mechanism. With this mechanism, *stop* and *go* signals are sent from the downstream nodes to the upstream nodes. When the buffer occupancy of the downstream node reaches a fixed threshold, a *stop* signal is sent to the upstream nodes; from this time instant until the buffer occupancy of the downstream node drops again below another fixed threshold, the source cannot send any traffic. When that happens, a *go* signal is sent to the upstream nodes and the source can send traffic again.

A significant amount of research is being carried out with that objective (see for example [Anti95], [Balt95] and [Mitr95]).

3.4. Statistical Multiplexing *versus* Deterministic Multiplexing

Communications' networks need to have some process that can control the way in which the resources are allocated, as this prevents congestion and loss. Thus, the way in which *bandwidth* (i.e., the bit rate or transfer capacity necessary for a given service or link) is allocated inside a network is crucial for it to obtain a good throughput and to operate at acceptable levels.

Determining the bandwidth required by a connection, so that the network can provide it with a certain QoS, is the function of *bandwidth allocation* (see for example [Fisa91]). This can be done in two ways:

* using *deterministic multiplexing*

  With this approach, the bandwidth given to each connection is its peak bit rate (i.e., the maximum speed at which the source can generate its traffic). Applying this to bursty connections will imply a great waste of bandwidth, especially if the connection is characterised by a large value of

$$\frac{\text{peak bit rate}}{\text{average bit rate}}. \tag{3.2}$$

  However, this approach (see [Gilb91]) can minimise *cell level congestion* (which happens when there are several cell arrivals occurring more or less at the same time); nevertheless, there is still a very small probability that buffers will overflow and cells will be lost. Deterministic multiplexing is not adequate for ATM, as it restricts the utilisation of network resources and does not take advantage of the multiplexing capacity that ATM offers.

∗ using *statistical multiplexing* (see [Gall89])

In this case, the bandwidth that is allocated to each source is less than its peak bit rate but greater than its average bit rate; this amount of bandwidth is commonly known as *statistical bandwidth* (also sometimes called *equivalent bandwidth* and *effective bandwidth*) of a source, whose value should guarantee the source its required QoS. So, with this approach several sources can share a link with a capacity that is less than the sum of the sources' peak bit rates; it is therefore assumed that the sources will operate at their peak bit rate for very short periods of time (see [Gilb91]). Obviously, the efficiency of this multiplexing technique will be better when the sources' statistical bandwidths approach the respective value of average bit rate. Statistical multiplexing gives better utilisation than deterministic multiplexing because, given a certain amount of bandwidth, it allows more connections to be multiplexed. However, its adequate use (i.e., when resulting in some multiplexing gain) produces congestion at cell level and potential cell loss.

The calculation of the *equivalent bandwidth* of a traffic source has been studied by several authors (see for example [Gall89], [Habi91] and [Hong91]) and it seems to be dependent on the following factors, by decreasing order of importance:

– *ratio of the source's peak bit rate to the link rate*, which should be smaller than 0.1 in order to obtain some statistical multiplexing gain (see [Hong91]);
– *burstiness*, which becomes important when the previous factor has a very low value (i.e., there is multiplexing gain);
– *burst length*, since cell loss and delay increase when there is an increase in this factor.

For example, [Pitt91c] and [Pitt91d] apply burst-level simulation to various traffic mixes in an ATM link, in order to find what is the relation between the bandwidth allocated to a traffic mix and the proportion of VBR services. The authors conclude that, when there is a decrease in the proportion of VBR sources in a traffic mix, the bandwidth (per source) allocated to those sources must increase, so as to maintain a constant cell loss probability.

Although statistical multiplexing seems to be the best way to allocate bandwidth in an ATM network, its significant gain (or not) depends not only on the burstiness of the sources to which it is applied, but also on the sources' chosen traffic parameters and the traffic models used to describe the cell arrival process.

Finally, it is worth noting that even using statistical bandwidth, an ATM network may not allow the loading of a link bandwidth at 100% and the bandwidth available to users may not exceed 80% of the link's peak bandwidth.

3.5. QoS in ATM

The expression *quality of service* has been used before in this document without any definition being given; this is because of the intuitive meaning of QoS. However, its real meaning and role in ATM networks is far more complicated than it would seem to be.

This section will define, according to ITU, the concept of QoS and will present some of the performance measures of ATM multiplexers that can help in the difficult task of quantifying quality of service in ATM networks. Some of the measures described in this section have been used to produce results from simulations relevant to the research work described in this thesis.

Apart from being intuitive, the definition of QoS is also subjective, as it depends on the user's view of a service; to complicate things even more, in B-ISDN networks, there are also different types of users and services. The definition that ITU gives for QoS in its Recommendation I.350 (see [ITU88a]) says that, "... it is the collective effect of service performances that determine the degree of satisfaction of a user of the specific service ...". Table 3.4 (taken from [Fisc94]) shows the QoS requirements of some applications.

Some of the parameters that are used to quantify the connections' QoS are characteristics of end-to-end delay, cell loss probability and bit error rate. The *traffic related measures for QoS* in ATM networks can be divided into two classes:

* *connection level parameters* that are associated with connection-oriented networks,
    - connection set-up delay;
    - connection release delay;

- connection acceptance probability (called *blocking probability* in telephony networks);

* *cell level parameters* defined for packet networks,
  - Bit Error Ratio (BER);
  - Transfer Delay;
  - Cell Delay Variation (CDV).

Application	*Traffic Characteristics*	*QoS Requirements*
real-time video (using MPEG)	4 Mbit/s to 6 Mbit/s, CBR traffic	low average delay and delay variation; very low cell loss ($< 10^{-10}$)
electronic mail	each message size ranging from less than 1000 bytes up to several Mbytes, with very long message interarrival times	maximum delay should be in the order of a few minutes
software download	file sizes up to several Mbytes	maximum delay in the order of a few seconds

Table 3.4 - ATM applications and their QoS requirements.

Connection set-up delay represents the time that it takes for a call set-up message transfer to be acknowledged, excluding the user's response time. This parameter should usually have a mean value of less than 4.5ms and 95% of the delay values should be less than 8.35ms (see [ITU88b]). On the other hand, *connection release delay* is the time that it takes for a call release message transfer to be acknowledged. In this case, the mean value should be less than 300ms and 95% of the values for delay should be less than 850ms.

Connection acceptance probability is defined as the quotient

$$\frac{\text{accepted calls}}{\text{total number of calls}}, \tag{3.3}$$

calculated over a long period of time, while the *BER* is given by,

$$\frac{\text{number of bit errors}}{\text{total number of bits transmitted}}, \tag{3.4}$$

with respect to the information field; for example, videoconference should have a bit error ratio of 10^{-11} (not counting the error handling in the AAL). The quotient,

$$\frac{\text{number of lost cells}}{\text{total number of cells sent}} \tag{3.5}$$

gives the *cell loss ratio* (CLR), which has a great impact on the QoS provided to users. The videoconference service (with a mean bit rate of 5 Mbit/s) should have, according to [ITU92], CLR = $4{\cdot}10^{-9}$ (without error handling in AAL), and an ordinary voice communication tolerates an end-to-end cell loss ratio of 10^{-3}.

Different types of services react differently to cell losses; for example, data services need to have low CLR values, but voice traffic can tolerate moderate cell losses. The simulation results shown in this thesis concentrate on this measure.

Transfer delay between two points in the network is the time necessary for all the bits of a cell to go from the first point (which can be a switch) to the second point. Some of the factors that determine transfer delay in ATM networks are,

* *packetisation delay*

  It is the time needed to accumulate the necessary amount of bits in order to form an ATM cell; this time depends on the type of AAL used, as well as on the source bit rate (it will be longer for low bit rate sources).

* *propagation delay*

  This type of delay depends on the distance between the source and the destination and it represents the amount of time that is necessary for a signal (e.g., a bit or a cell) to travel along the transmission medium. For coaxial cable, the propagation delay is $4{\cdot}10^{-9}$ s/m.

* *transmission delay*

  It corresponds to the time that is necessary to wait until all the bits of a cell arrive from the transmission link, before they can be processed. The transmission delay becomes negligible when the transmission speed is very high.

Finally, there is another parameter, *Cell Delay Variation* (CDV), that has been defined in a number of ways according to the utilisation purpose. One of them says that, if W_i is the random delay component of the *ith* cell, then CDV is the variance of the transmission delay of a connection, i.e.,

$$E\left(\left[W_i - E(W_i)\right]^2\right). \tag{3.6}$$

This is the so called *2-point CDV* (see [ITU93]), used to measure the end-to-end delay of a connection. Another definition of CDV represents it by the complementary distribution function of the difference in transfer delay between consecutive cells (see [Kühn95]); this is also called *1-point CDV* (see [ITU93]) and it is used by the UPC function.

CDV becomes more noticeable with services like speech transmission, for which the difference in delay alters the user's view of QoS; services with this characteristic are called *delay-sensitive services*. The next table (Table 3.5) gives some values for CDV taken from [Onvu94, pp.81].

Application	***Delay (ms)***	***CDV (ms)***
video conference at 64 Kbit/s	300	130
compressed voice at 16 Kbit/s	30	130
HDTV video at 20 Mbit/s	0.8	1

Table 3.5 - CDV values for some services.

Of all the QoS measures described, *propagation delay* and *cell loss* due to errors in the header field during transmission contribute to degrade QoS; however, they are independent of the traffic type considered. According to [Sait94, pp.135], *cell loss* and *delay* in ATM nodes are the parameters that more significantly contribute, at cell level, to degrade ATM network performance. It is worth noting that the control of each of these parameters has a special impact in the other parameter; for example, real time traffic such as voice can tolerate some cell loss but is very sensitive to cell delay, while non real time traffic such as file transfer can allow for some cell delay but not cell loss.

One last remark should be made about a measure for the network traffic performance and to which not much attention is usually paid: the users' Grade of Service (GoS), which is closely related with the QoS. The parameters associated with that measure are the *end-to-end cell delay* and *end-to-end cell loss*. These are determined by combining the information obtained for the cell delay and cell loss, respectively, at each of the intermediate switches of an ATM connection (see [Cuth93, pp.141-142]).

3.6. Traffic Control

The set-up of a connection in an ATM network implies the specification of a cell stream's traffic characteristics. After this declaration, the connection will only be accepted if the network has enough available resources to accommodate both the new connection and the already established connections, without compromising the QoS of those connections. During the life time of the connection, monitoring is performed to ensure that the traffic characteristics of the cell stream agree with the ones declared at the connection set-up phase; if there are any discrepancies, penalties are imposed to the user. This global process is called *traffic control*, while the decision of accepting a connection and allocating network resources to it is called *Connection Admission control* (CAC). The monitoring mechanism is called *Usage Parameter Control* (UPC); these are the two most important traffic control mechanisms.

The procedures that should be followed to decide how to provide resources and monitor the sources are still partially unsolved problems. One of the reasons for this is that ATM networks allow a great diversity of traffic characteristics and QoS requirements, which makes it difficult to take decisions on a network scale. Another problem has to do with the CDV that occurs in a subscriber line; in fact, even if the users specify the traffic characteristics at the terminal, then the network is unable, because of CDV, to determine the true characteristics of the traffic offered to it.

The rest of this section presents possible parameters that may help to determine the admission/refusal of a connection and once admitted, the subsequent monitoring. The description of UPC in this thesis is to give an introduction to the subject, which will be analysed (in relation to traffic with priorities) in a greater detail in Chapter 7. CAC is defined because of its close relation with UPC. For a general introduction to traffic control mechanisms, see [Eckb92].

3.6.1. Connection Admission Control

CAC consists of a set of actions taken by the network at the set-up of a VC or VP connection that determine if the connection can be accepted or not. The acceptance decision is based on,

- the estimated traffic characteristics of the VC;
- the QoS requirements of the VC;
- the current network load,

and it is made for each VP and VC along the connection route. The estimation of the traffic characteristics takes into account the values from the traffic descriptor (defined in Section 3.6.2) and the CDV values, while the estimation of the current network load is usually based on the traffic descriptor values of the existing connections; if possible, a measurement of traffic is also taken into account.

CAC will only admit the new connection if there are enough resources available for the network to maintain the existing connections' QoS and to provide the requested QoS to the new connection. The VC connection will only be established if it is accepted at each VP along the route. If CAC decides not to admit the connection, then the routeing algorithm will have to try to find another route; the connection attempt is lost and cleared if there is no other route.

Several approaches have been suggested for CAC algorithms (see for example [Cast91], [Mase91] and [Lee92]) which differ mainly in the method used to characterise the traffic and predict future traffic. However, so far, the methods for characterising traffic (that are needed as an input to CAC) have not yet been standardised but some guidelines can be found in [ITU95a] and [ITU95b]. These documents propose as mandatory traffic parameters the *peak cell rate* and the *cell delay variation tolerance* for that peak cell rate. The aim of the approaches developed is to produce an algorithm that is *simple* (in terms of processing and storage requirements), *robust* (to guarantee network performance) and *efficient* (in order to obtain statistical multiplexing gain).

Most CAC algorithms make use of the concept of *effective bandwidth* of a source. This represents the bandwidth required by a source in order for the network to be able of providing the requested QoS. Several methods have been devised to calculate the effective bandwidth of a source (see for example [Guér91]). In [Miya93], a CAC algorithm is developed that, unlike others, takes into account the variations of QoS being received by individual connection types over time (due to the burstiness of traffic). [Mura91] reports on a CAC algorithm that considers a bufferless fluid model for both homogeneous and heterogeneous scenarios of On/Off traffic sources characterised by their peak and mean rates. The CAC scheme is based on estimating the cell loss quality of individual bursty sources.

The simplest approach to execute the CAC function is *peak rate allocation* (see Section 3.4); however, this does not take advantage of statistical multiplexing inherent in ATM networks. Other possible approaches include,

* *linear*

This algorithm is simple but it can produce results that are close to peak rate allocation. Two quantities are compared, assuming that there are *m* types of traffic sources in the system:

– the sum of the calculated effective bandwidths required by the total number of sources in the system plus the new source,

$$f(N_1, N_2, ..., N_m) = \sum_{i=1}^{m} B_i \cdot N_i \,, \tag{3.7}$$

where B_i is the effective bandwidth for each source of type *i* and N_i is the number of sources of type *i*;

– the link capacity *C* (which is a previously calculated percentage of the maximum capacity).

The new connection will only be accepted by the system if the following condition applies,

$$f(N_1, N_2, ..., N_m) \le C \,. \tag{3.8}$$

* *two-moment allocation (or Gaussian approximation)*

With this scheme, two parameters (the mean bit rate and the standard deviation) are used to describe each connection present in the system. The arrival of a new connection to the network implies the evaluation of the bandwidth necessary to accommodate all the connections, given the existing link capacity. To this end, it is assumed that the aggregate bit rate follows a Gaussian distribution. The drawback of this type of algorithm is that, usually, it only leads to good results if the peak bit rate of each connection is a small percentage of the link capacity.

* *convolution*

In this case, it is assumed that there is a knowledge of the bandwidth distribution for each service type in the system. With this information, the distribution of the aggregate traffic is calculated by convolution and a new connection will be accepted by the network if the obtained values for parameters like the maximum allowed load and congestion probability are within previously fixed bounds. Although this algorithm is relatively accurate, it is computationally heavy.

A description of these approaches can be found in [RACE92]. CAC can also be combined with priorities; this is the so called *hybrid CAC* (see [Sait94, pp.112-120]) that can be applied to connections requiring a different QoS by providing two cell loss ratio classes (by making use of the CLP bit in the header of an ATM cell) and introducing a priority mechanism at the output buffer. On the other hand, [Esak90] considers CAC and priorities simultaneously to develop a CAC mechanism that works in real-time; the traffic sources used are characterised only by their peak and mean bit rate.

3.6.2. Usage Parameter Control

According to ITU Rec. I.371 (see [ITU94]), UPC "... is the set of actions taken by the network to monitor and control traffic, in terms of traffic offered and validity of the ATM connection at the user access network ...". In other words, after a connection is accepted by the network using a CAC procedure, it must be ensured that network resources are protected from malicious or unintentional misbehaviour of the source producing the call, which can affect the QoS of other already established calls in the network. This is the job of the UPC function, sometimes also called *traffic policing*, situated at the access point to the network, i.e., at the end of the first *link*. Typical locations for the UPC function include the entrances to a local switching node and a cross-connect.

The monitoring of traffic sources in a network is performed by both detecting violations of traffic parameters negotiated at call set-up and taking actions, if violations occur. To this end, UPC is performed for each user VP/VC pair being used in the network and the following tasks are executed:

* to check the validity of each VP/VC identifier;
* to count the number of cells that are arriving and that belong to a certain connection;
* to check the agreement between the counted value and the declared value at the connection set-up phase; if they do not agree, then a penalty should be imposed on the connection.

UPC is performed for each traffic parameter of a source traffic descriptor, where a *source traffic descriptor* is a set of traffic parameters that specify the traffic characteristics of a connection at the time it is set up (see [ITU94]). Every *traffic parameter* that is part of a source traffic descriptor should be understandable by the user and have significant use in resource allocation; the peak cell rate is one such traffic parameter that is mandatory in a source traffic descriptor.

When a violation is detected by the UPC, this traffic control can either cause cells to be discarded or it can mark them, so that they can be lost first when the network is congested (see [ITU94]). The latter action implies a method to mark cells; this is still under study, but a method has been suggested that uses the CLP bit in the cell header. This means that, when a situation of congestion occurs in the network, marked cells will be lost first. An alternative way to tag cells could be to use the payload type field present in the cell header (see [Sait94, pp.73-76]). Other actions that a policing function can take in case of contract violation are,

* delaying violating cells in a queue so that the departure from the queue agrees with what was declared in the contract;
* controlling the traffic in an adaptive way, by informing the source of when it starts violating the contract.

Each request by a user to establish a call in the network requires a *traffic contract*. This comprises the declaration by the user of certain traffic parameters that describe the characteristics of the call request. So far, the only traffic parameter standardised by ITU is the *peak cell rate*, although the last released version of ITU Rec. I.371 (see [ITU95b]) also indicates as mandatory the associated *cell delay variation tolerance* and proposes the use of the *sustainable cell rate*. Other parameters have however been suggested to characterise traffic sources (and have been used in research work, e.g., [Butt91] and [Cast91]), such as:

- average cell rate;
- burstiness;
- burst duration.

The problem with some of these parameters is that, although they contain more information about the traffic source that they represent, it is felt that normal users may not be able to be very precise about the traffic characteristics of their sources. As a response to this problem, a new class of traffic sources has been defined by the ATM Forum: the Available Bit Rate (ABR) traffic, which was described in Section 3.3. Importance is also given to ABR traffic by ITU in its last released version of Rec. I.371 (see [ITU95b]).

3.6.2.1. Policing Algorithms

A policing algorithm should always be able to detect and quickly respond to any traffic violation in the network. It should also not take any action on traffic that is compliant, i.e., traffic that does not violate the traffic contract negotiated at call set-up. Several policing algorithms have been proposed (see for example [Butt91], [Habi91], [Okad91], [Guil92], [Cuth93, pp.113-118] and [Onvu94, pp.134-145]) to regulate traffic flows in a network, by controlling the sources' traffic parameters. In most cases, the controlled parameters are *peak* and *mean cell rate*. The algorithms can be divided into two main groups:

- window based mechanisms;
- Leaky Bucket.

The first group considers algorithms in which fixed or variable time windows limit the number of cell arrivals, while the Leaky Bucket mechanism (and its variants) is based on a counter that is incremented whenever there is a cell arrival and decremented in the opposite case.

Examples of window algorithms (see [Butt91] and [RACE92]) are schemes like Jumping Window (JW), Triggered Jumping Moving Window (TJMW) and the Exponentially Weighed Moving Average (EWMA). In the *JW* mechanism, fixed and consecutive intervals with X time slots are observed; they allow a maximum of Y cell arrivals. If more than Y cell arrivals are observed in a time window, the first Y cells are allowed to pass and the others are lost (or marked). When each time window ends, the counters used to evaluate the number of cell arrivals and time slots are reset and a new time window begins. *TJMW* is a variation of this mechanism; it considers time windows that are started by a new cell arrival; thus, the time windows are not necessarily consecutive. Although similar to the JW, the *EWMA* scheme does not consider a fixed maximum number of cell arrivals in each time window. The number of cell arrivals that can be accepted in each time window is an exponentially weighed sum of the number of accepted cells in the previous time window and the mean number of cells. In other words, if we take

T = window size

m_i = number of accepted cells in the *ith* window; i =1,2, …

m = average number of cells

then

$$m_i = \frac{m - \delta \cdot S_{i-1}}{1-\delta}, \; 0 \le \delta \le 1, \; \text{where } S_i = (1-\delta) \cdot m_i + S_{i-1}; \; i = 1,2, \ldots \tag{3.9}$$

The variable δ controls the flexibility of the mechanism on what concerns the traffic's burstiness and S_0 is the initial value of the mechanism's measurement. In the particular case of $\delta = 0$, this policing mechanism reduces itself to the JW mechanism.

Leaky Bucket (see Fig.3.5) is probably the most well known algorithm that helps UPC judging a violation in the connection set-up phase and a number of variants can be found in the literature (see [Ahma90], [Butt91], [Cast91] and [Chao91]).

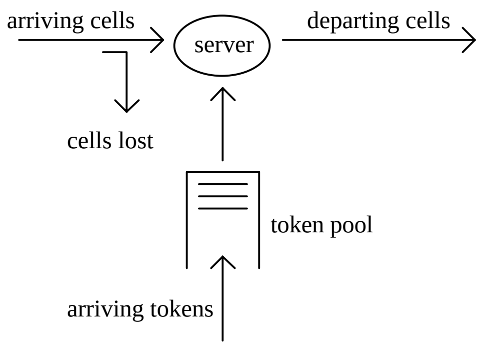


Fig.3.5 - The principle of the Leaky Bucket mechanism.

The idea behind Leaky Bucket is that, when a cell arrives to the network, a token is introduced in a token pool (also called *bucket*, hence the name of Leaky Bucket for this policing mechanism). If the token pool is not full, then a token is added to the token pool and the token pool size is increased by one. The token pool is emptied at a constant rate and it works as a virtual finite queue. Cell arrivals are discarded (or tagged) when the token pool is full. The algorithm can be schematically described in its simplest form by the following:

1. decrement a counter *C* by 1 every *T* seconds down to *0* if there are no cell arrivals; $(0 \leq C \leq M)$
2. increment the counter *C* by 1 for each transmitted cell;
3. drop (or tag) cells when C=M.

M represents the maximum burst size, while *1/T* is the peak cell rate of the traffic source being observed. The rate at which the counter is decreased is called the *leak rate* and the counter *C* represents a *bucket* being filled with tokens, as referred to before. This mechanism can also be used to police the parameter *mean rate*; in this situation, the size of the bucket is taken to be a large value, while the leak rate is set to the mean rate of the source being policed (see [Mitr94]).

When Leaky Bucket is used to police the parameter *peak rate,* then the bucket size is fixed to have to have a unitary value and the leak rate is taken to be equal to the peak rate of the policed traffic source. This changes when CDV is catered for (see Section 3.6.2.2), as in that situation, the policing rate will be slightly higher than the declared peak bit rate of the source.

A quick look at the mean and peak rate policing methods shows that when policing the mean rate, the time taken by the UPC mechanism to detect traffic violations will be much longer than for peak rate policing; therefore, the use of mean rate policing leads to an ineffective protection of the network against misbehaving traffic (see [Butt91]).

It should be noted that the variants of the Leaky Bucket mechanism usually consider buffering before the traffic is analysed by the UPC. Although this helps shaping the traffic (this subject will be dealt with in Section 3.7), it also introduces extra delay. Examples of this approach can be found in [Kim92] and [Wu93], where bursty input traffic is modelled as an Markov Modulated Poisson Process (MMPP) and buffered Leaky Bucket is considered.

The Leaky Bucket scheme can monitor traffic at VP or VC level. When VP policing is performed, the only measure of interest is the aggregate declared bit rate offered to the VPs, which can usually be easily policed; therefore, the contents of each VC multiplexed in a VP and their source characteristics are ignored for policing purposes. However, when policing is executed on a VC basis, it becomes more complicated to control the sources' traffic parameters, as in this case, the sources have to be characterised by at least their peak and mean cell rates. In [Butt91], the fluid flow approach was used to derive a formula for the cell loss probability of On/Off sources having their activity and silence periods exponentially distributed. This paper shows that, while peak cell rate can be easily controlled by the Leaky Bucket mechanism, the same does not happen with the control of mean cell rate; this is due to the long time needed to collect statistics leading to an accurate estimation of the mean cell rate of a traffic source, which prevents the policing function from giving a quick response to possible traffic contract violations.

Several authors have addressed the problem of comparing the performance of the different policing mechanisms proposed (see for example [Rath91], [Butt91] and [Kim92]). Of the window mechanisms, the JW scheme is considered to be the simplest but also the one that gives the worst performance. The Leaky Bucket algorithm is usually agreed to give better results than window based schemes and it is also the policing mechanism recommended by ITU (see [ITU95b]).

It has been demonstrated (see [Grav91b]) that sources' traffic streams are better controlled when violating cells are discarded, as opposed to tagging them. However, this solution implies stringent requirements for the violating probabilities of well-behaving traffic sources. For example, [Yin91] considers the analysis of Leaky Bucket for On/Off data sources modelled as two-state Markov modulated rate processes. Both unbuffered and buffered Leaky Bucket algorithms are studied with either discarding or marking of non compliant data. The authors derive explicit expressions to calculate the bucket size (queue buffer size) as a function of the loss (tagging) probability and source characteristics (according to whether non compliant cells are discarded or marked, respectively). It is shown that there is a logarithmic increase in the necessary bucket size as the loss or tagging probability decreases.

3.6.2.2. Effect of CDV on UPC

In an ATM network, the cell delay experienced by cells of the same connection can vary. This is called *Cell Delay Variation* (CDV), which was defined earlier in Section 3.5 and is caused mainly by buffering at ATM switching nodes. Since CDV can change the traffic characteristics of sources in the network, the bandwidth required by those sources can also change. Therefore, UPC should take CDV into account when deciding whether cell arrivals are or are not violating the respective traffic contract negotiated at call set-up. In this case, the maximum allowed CDV should be an extra parameter to be included in the traffic contract (see [Skli93]).

When UPC takes CDV into account, it introduces tolerances in the amount of traffic from a source that is compliant to the respective traffic contract. It is thus possible that some cells that are really violating the traffic contract are not discarded. One way to compensate for this effect is for the network to introduce shaping (see Section 3.7) just before UPC is performed. This reduces the CDV generated in a network (see [Sait94, pp.125-126]). Shaping is used to smooth the clumping of cells from a traffic source by using buffers where cells are put before they enter the network. The buffers are read at a rate determined by the shaping process. Although this process can reduce the clumping, it does so at the expense of delaying the traffic generated by the source; thus, it is a process that can only be tolerated by traffic not sensitive to delay.

3.6.2.3. Policing and Standardisation

Recently, two organisations (the ITU and the ATM Forum) have been addressing the subject of policing. The policing algorithm used by ITU (see [ITU94]), known as Continuous-State Leaky Bucket Algorithm (CSLB) and which has an equivalent in the Virtual Scheduling Algorithm (VSA) also defined in the same document, takes two parameters: the peak emission interval T (which is the inverse of the peak cell rate of an ATM connection) and the CDV tolerance value τ. Let,

C = value of the Leaky Bucket counter

lct = last compliance time

t_arrival = time of a cell arrival.

The algorithm is then as shown in Fig.3.6.

```
record first cell arrival in t_arrival;
C := 0;
lct := t_arrival;
while there are cell arrivals do
  begin
     record a cell arrival in t_arrival;
     C := C - (t_arrival - lct);
     if (C < 0) then
       begin
          C := 0;
          C := C + T;
          lct := t_arrival;
          (* arrival in t_arrival is a compliant cell *)
       end
     else begin
            if (C > τ) then
              (* arrival in t_arrival is a non-compliant cell *)
            else begin
                   C := C + T;
                   lct = t_arrival;
                   (* arrival in t_arrival is a compliant cell *)
                 end;
          end;
  end;
```

Fig.3.6 - The Continuous-State Leaky Bucket algorithm.

The decision of considering a cell as non compliant implies one of two actions, as referred to before: tagging or discarding. In the first case, the tagging of a cell consists of setting the CLP bit to 1, thus turning it into a low priority cell. Two scenarios are possible (see [ITU94]) for deciding on whether to allow a cell (be it low or high priority) to pass, to mark a cell or discard it:

a) *no cell tagging*

With this strategy, UPC starts by checking if the CLP=0 stream (representing high priority traffic) is compliant (i.e., is respecting the agreed traffic contract). Any cells that are found to be non-compliant will be discarded. A second conformance test is then performed on the aggregate traffic stream (i.e., the CLP=0+1 stream). The aggregate traffic stream is now the sum of the compliant CLP=0 traffic (after the first conformance test) and the CLP=1 traffic (i.e., low priority traffic). The second conformance test will again discard any cells (irrespective of their

CLP bit value) that do not respect the agreed traffic contract.

b) *cell tagging*

Two conformance tests are performed in this case. The first will again check whether the CLP=0 stream is respecting the traffic contract of the corresponding traffic source. Cells that violate the traffic contract will be tagged (i.e., their CLP bit will be changed to 1). After this test, the CLP=1 traffic stream will consist of the original CLP=1 stream and the tagged CLP=0 cells. The second conformance test is once more applied to the aggregate traffic, which consists of the CLP=0 and CLP=1 traffic streams' sum. Any cells (be it low priority - CLP=1 - or high priority - CLP=0) that are found to violate the traffic contract will be discarded.

The conformance tests are executed by making use of the chosen algorithm, which in this case is CSLB.

The Generic Cell Rate Algorithm (GCRA) proposed by the ATM Forum is basically a VSA algorithm (see [ATMF94]). However, this organisation is also considering UPC algorithms with three components:

- a peak rate controller for the aggregate cell stream (CLP=0+1);
- maximum sustainable rate controllers for both low and high priority cell streams (CLP=1 and CLP=0 streams, respectively).

The GCRA algorithm is characterised by two parameters, representing:

- an *increment*, proportional to either the peak cell rate or sustainable cell rate:
- a *limit*, which determines the bucket size used (this limit reflects either the CDV or the burst tolerance, respectively).

The drain (or leak) rate has been fixed by the ATM Forum to be one, so both the increment and the limit are normalised to it.

Finally, another UPC algorithm (based on the Leaky Bucket algorithm) is being considered by the ATM Forum (see [ATMF92], [Rama94] and [Mark95]): the *dual Leaky Bucket algorithm*. With this scheme, two Leaky Bucket algorithms are executed in parallel and two new concepts are used: *sustainable cell rate* (defined as an upper limit for the maximum allowed average rate of a connection) and *burst tolerance* (the maximum time during which a source can send traffic at its peak cell rate).

The two Leaky Buckets are described by the parameters (T_1, M_1) and (T_2, M_2)

for Leaky Buckets 1 and 2, respectively, in an analogue way to the description of Leaky Bucket given in Section 3.6.2.1. This means that T_1 and T_2 are the leak rates and M_1 and M_2 are the bucket sizes. Each cell arrival will go through both Leaky Buckets and it will be considered to be compliant if the following condition holds:

$$C_1 \leq M_1 \quad \textit{and} \quad C_2 \leq M_2\,, \tag{3.10}$$

where C_1 and C_2 are the Leaky Buckets' counters. If $T_1 > T_2$ then the first Leaky Bucket is known as *peak rate Leaky Bucket* and the second is named *sustainable rate Leaky Bucket*, from which it follows that T_1 is called *peak rate*, T_2 is the *sustainable rate* and $M_1 + 1$ is the *burst tolerance*.

In [Rama94], a series of traffic classes are defined, each with a specific set of control parameters. The UPC mechanism used thus enables or disables the appropriate controls when in presence of traffic from a certain class.

It is felt that a double Leaky Bucket mechanism will be more efficient in terms of policing traffic sources than a simple Leaky Bucket algorithm. Note also that the policing schemes proposed by ITU (see [ITU94]) consider a process with two Leaky Buckets (one for the CLP=0 stream, i.e., the high priority traffic, and another for the CLP=0+1 stream, i.e., the aggregate traffic).

3.7. Traffic Shaping

Many Variable Bit Rate (VBR) sources (defined in Section 3.3) generate cells at their peak bit rate during the so called *active period*; during the *silent period*, no cells are generated. In order to reduce the peak rate of such type of sources, cells can be buffered before they enter the network so that they leave the queue at a smaller rate, when compared with the peak rate (but still greater than the average bit rate, to prevent the queue from becoming unstable). This process is called *traffic shaping* and it is used particularly for bursty sources. Although the process can reduce the peak rate of a source, it does so at the expense of delaying the traffic generated by the source; thus, it is a process that can only be tolerated by traffic not sensitive to delay (such as file transfer). Another disadvantage of using shaping is the potential necessity to have additional buffer space (see [Gilb91]).

The shaping function allows the users to control their traffic parameters, like the peak bit rate and the maximum source activity (fraction of time during which a source transmits) allowed in a given time period (see [Yazi92]); it also maintains the cell-sequence integrity of a connection. Fig.3.7 shows a general block diagram of traffic shaping where the traffic shaper represents the function that determines how the spacing of cells is executed.

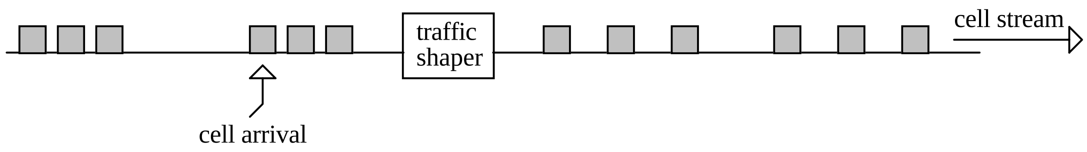


Fig.3.7 - Diagram of traffic shaping.

It has been shown (see [Yama92]) that the use of traffic shaping improves the link efficiency for VBR traffic, especially if the bursts are small. Other references to this mechanism can be found in [Cuth93, pp.118-119] and [Sait94, pp.125-126].

3.8. Priority Mechanisms

In order to optimise the network utilisation, while meeting the requirements of each type of traffic source, it is possible to use priority mechanisms. The user may generate different priority traffic flows by using the CLP bit capability and when buffer overflow occurs, cells from the low priority flow can be selectively discarded by network elements. The need to classify traffic/customers into priority classes can be found for example in computer systems and in the computer control part of digital switching exchanges; it becomes especially important to guarantee that information as vital as control and signalling messages is transmitted rapidly and securely across a network.

Priority mechanisms include time priorities and space priorities. *Time priority mechanisms*, such as the Head-Of-the-Line (HOL) scheme (see [Hong91]), take into account that some services may tolerate longer delays than others (e.g., specific data services *versus* voice). No explicit support for time priorities exists in ATM, but the subject has been studied. On the other hand, *space (or loss) priorities* propose to provide several grades of services through the selective discarding of low priority cells; this type of priority mechanisms exploits the fact that any traffic source may generate certain cells that are less important than others and may therefore be discarded in the case of congestion, without significantly compromising the sources' QoS requirements. The use of priority mechanisms affects mainly two traffic performance measures: the cell loss probability due to buffer overflow and the waiting time, translated into delay. In other words, the cell loss probability and waiting time of high priority traffic is decreased, when compared with the no-priorities case. On the other hand, the low priority traffic suffers in many cases, but not all (see [Schw87]), a small increase in the waiting time and cell loss. However, it should be noted that a conservation law is verified for the waiting times, provided an infinite waiting space is considered). In fact, if there are *n* traffic priority classes in a system, the weighted sum of the waiting times of all the classes is always conserved, as represented by the following equation:

$$\sum_{k=1}^{n} \rho_k \cdot E(W_k) = \rho \cdot E(W) \tag{3.11}$$

where ρ_k represents the traffic intensity of traffic class *k* (k = 1, ..., n), $E(W_k)$ is the mean waiting time of traffic class *k*, ρ is the traffic intensity of the system without priorities and $E(W)$ is the mean waiting time of the system without priorities.

In Chapter 1 it was noted that, due to the possibility of using the CLP bit in the header of ATM cells, it is possible to assign priority levels either at cell level (in which case sources can have both low and high priority cells) or at connection level (in this situation, all the cells generated by a traffic source will either be of low or high priority). In the second case, called *route separation*, different queues can accommodate the low and high priority traffic. Some authors have considered this situation (see [Alon89] and [Gall90]).

Many studies have addressed priority mechanisms; more information can be found in [Grav91a], [Roth90], [Ren94] and [Krön90]; [Robe91a] gives a summary of some priority mechanisms developed and points to specific papers in this area.

The sub-sections that follow will describe several types of priority mechanisms that have been studied so far.

3.8.1. Time Priorities

It was referred to in the last Section that time priorities are not explicitly identified in ITU Recommendation I.361 (see [ITU90a]) on the specification of the B-ISDN ATM Layer; however, they could be implicitly represented by using VPI/VCI combinations.

HOL is the most well known mechanism applicable to time priorities. This is a simple scheme to serve multiple classes with various delay requirements, by classifying the traffic into *k* fixed priorities. The input buffer is divided into *k* queues and an arriving cell is placed in its corresponding queue. As long as the *class 1* queue (representing the highest priority) is not empty, cells in this queue are served. When the *class 1* queue becomes empty, cells from the *class 2* queue can be served. When both the *class 1* and *2* queues become empty, cells from *class 3* can be served and so forth. This method is useful for continuous bit rate traffic, since it will always have service priority. However, performance is poor for lower priority classes. The delay for these classes can become too large if there is a large volume of high priority traffic.

HOL with Priority Jumps (HOL-PJ) is a variation of HOL that tries to solve the problem of giving too much priority to one class. The basic idea is that cells in lower priorities should also have some chance to transmit even if there are higher priority cells in the queue. This will put some bound on the maximum delay that lower priority cells will encounter. The method consists of allowing a cell to jump to the next higher priority queue when that cell has spent a time in a queue that is greater than the local delay limit for that queue. The implementation of this method has been shown to be simple (see [Eckb92]).

Time priorities have been studied for example in [Grav91a], [Scho91] and [Scho92]. In the first case, the study combined time and loss priority mechanisms (HOL and Pushout, respectively). [Scho92] presents algorithms for calculating steady state waiting time probabilities for cells of any priority level in ATM queuing models; the algorithms include an arbitrary number of non pre-emptive time priority levels (i.e., in which a cell of the highest priority goes to the head of the queue on arrival but cannot get into service until the cell that is being serviced finishes, even though this cell may be of a lower priority) and represent an extension of the GEO/D/1 queuing model. Another example of research that considers both time and loss priorities can be found in [Huan94]. In this paper, the objective is to develop a mechanism, using a Markov chain, that can guarantee both the delay and packet loss requirements for each traffic class present in the system (an *NxN* ATM switch with a buffer of size *B*); the arrival of packets (or cells) is a Bernoulli process with a certain traffic load. The algorithm proves to have a time consuming implementation, even for small size switches.

3.8.2. Space Priorities

The mechanisms investigated in the literature are fundamentally the Partial Buffer Sharing (PBS) and the Pushout mechanism. Both mechanisms provide more than one QoS. To this end, each source marks every cell with a priority level indicator. Thus, each source may have so called *low priority cells* and *high priority cells*. High priority cells (that can also be designated by *Class 1 cells* or *vital cells*), are cells that should have a very low loss rate and low priority cells (also called *Class 2 cells* or *ordinary cells*) are cells that may be lost in case of congestion. Thus, when the proportion of vital cells is small if compared with the amount of ordinary cells, the cell loss requirement will be lower, which means that it will be possible to increase the load. For the description (given in Sections 3.8.2.1 and 3.8.2.2) of the two space priority mechanisms, a queue is considered with a server attached to it, to which cells of both priorities may have access and that serves those cells according to the mechanism implemented.

Performance studies (see [RACE92]) have proved that a significant improvement of the admissible traffic load can be obtained when cell loss priorities are applied. This allows smaller buffer sizes to be chosen, therefore reducing the complexity of the implementation.

A description of several space priority mechanisms is given in [Krön91], along with a study of their performance characteristics; results are presented for sources with Poisson input and for On/Off sources with exponentially distributed *On* and *Off* periods. [Lin91a] makes use of the model $\mathrm{D}^{[\mathrm{A1},\,\mathrm{A2}]}/\mathrm{D}/\mathrm{c}/\mathrm{B}$ to study the performance of various space priority mechanisms, taking also into account the possible correlation between cells of different priority levels. In [Garc92], matrix analytic methods are applied to evaluate the Pushout and the PBS, modelling the input traffic by means of a Markov Arrival Process (MAP); this process is at the same time analytically simple and possesses properties that make it suitable for the approximation of complicated non-renewal processes (renewal processes are defined in Section 4.1.3). Numerical results are given for the case of an MMPP (also described in Section 4.1.3), which is a particular case of an MAP process. In [Garc91], a fluid flow approximation model is used to study the multiplexing gain that can be achieved for different classes of VBR sources as On/Off sources. The PBS and the Pushout mechanisms are applied by modelling the system with an M/G/1 queue; both fixed and adaptive thresholds are studied (see [Krön90]). The second case aims at taking the best threshold value when varying the load of the system, in order to maximise the admissible load.

A final remark should go to the work developed in [Dagi93]. In this paper, an algorithm is proposed that combines the philosophy of both the Pushout and PBS mechanisms. The mechanism is very similar to PBS up to the full occupancy of the queue (described in Section 3.8.2.2), i.e., it only accepts low priority traffic up to the threshold of the queue and it always accepts high priority traffic, provided the queue is not full. However, if a high priority cell arrives at the queue when the buffer is full, then that cell may take out of the queue the last low priority cell stored and be inserted at the end of the queue.

3.8.2.1. Pushout Schemes

In the *Pushout mechanism*, a high priority cell may enter the queue when in its full state, by taking the place of a low priority cell already in the queue (usually, it is the last low priority cell that entered the queue). If a low priority cell arrives at the queue when it is full, then it will be discarded. With this mechanism, vital cells will only be lost when the queue is full and there are no ordinary cells waiting for service in the queue. An analysis of this mechanism, using the model M/D/1, is performed in [Hébu90].

Fig.3.8 shows the way in which cells are accepted or rejected by the queue, according to their priority level. The main disadvantage of this mechanism is its complex implementation. While high priority cells are replacing low priority cells, it is still necessary to ensure that the sequence of cells is preserved. Hence, the buffer can no longer have a First In First Out (FIFO) discipline, which means that it will be necessary to keep track of where the low priority cells are stored, as well as the sequence of both high and low priority cells and overall sequence of cells. However, this mechanism can assure a very low cell loss to loss sensitive traffic (thus considered as high priority traffic), even in the case of a variable load of the low priority traffic (see [Robe91a]).

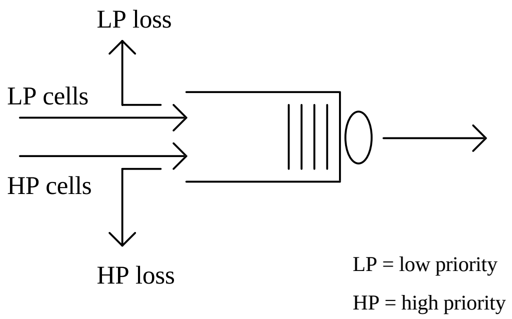


Fig.3.8 - The Pushout mechanism.

A recent work (see [Leme94]) has proposed another type of Pushout mechanism; it describes the so called *Multiple Pushout mechanism*, which is based on the utilisation of the features, both from the AAL and ATM layers and on a particular definition of the CLP bit. By making use of the two layers, the authors intend to avoid a problem that occurs with other loss priority mechanisms in case of overload: the discarding of cells without any semantic information about the type of cells, which causes, at the destination, the discarding also of all the fragments from corrupted messages, since the ATM layer cannot detect cell losses and does not provide selective cell retransmission mechanisms.

3.8.2.2. Partial Buffer Sharing Schemes

With the *Partial Buffer Sharing mechanism* (see Fig.3.9), both vital cells and ordinary cells are accepted by the queue until it reaches a threshold (usually taken to be about 70% of the total length of the queue). When this threshold has been filled, only high priority cells will be accepted, provided the queue is not full. The implementation of this mechanism is simpler than the implementation of Pushout, but its efficiency is not as good, since in this case, high priority cells do not have priority over low priority cells at all times (e.g., when the queue is full and there are ordinary cells waiting for service, arriving vital cells will be lost). However, it has been shown that the correct use of Partial Buffer Sharing may improve the maximum admissible load in a network (see [Boud91]).

The main problem with this approach lies in the determination of the threshold value. If it is set to a very low value, low priority cells at the buffer may be unnecessarily discarded (i.e., allowing more low priority cells would not affect the QoS given to high priority cells), thus restricting the effectiveness of marking cells. On the other hand, if the threshold is set to a large value, the performance of high priority cells may deteriorate, since there may not be enough space left to accommodate them. Besides, even though the threshold value depends on the characteristics (like load, burstiness and correlation) of both types of cell flows, more importance is given to the high priority cells' characteristics in order to guarantee their QoS requirements. Therefore, it may be necessary to adjust the value of the threshold when a change occurs in the characteristics of the traffic at the buffer. However, contrary to the Pushout mechanism, PBS can be implemented using the FIFO discipline without much complexity.

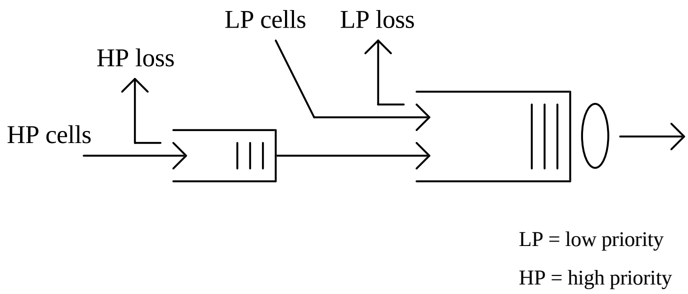


Fig.3.9 - The Partial Buffer Sharing mechanism.

The analysis of Partial Buffer Sharing mechanisms is reported in several papers. In [Boud91], algorithms are proposed to determine cell loss rates and the input sources utilised are modelled by a Markov Modulated Bernoulli Process.

[Liao94] has considered a discrete-time queuing model for PBS with two Markov modulated Poisson inputs, that can be used to analyse the effects of PBS on system performance for cases of bursty services. In [Bae92], the scenario studied comprises a queuing system with a finite buffer and heterogeneous sources, modelled as a Markov Modulated Poisson Process (MMPP) in continuous time and as a Markov Modulated

Bernoulli Process (MMBP) in discrete time. This study then calculates the loss probabilities for a priority packet discarding scheme (in this case, the PBS mechanism). A three-state Discrete Time Markov Chain (DTMC) is used in [Kang93] to model sources generating bursty traffic in a system with loss priorities. The three states of the model are: one *idle state*, where no cells are generated; one *low priority state*, where low priority cells are generated; and one *high priority state*, for the generation of high priority cells. The analysis is presented for the case of two types of services (or calls), but it becomes computationally intractable for a greater number of call types. [Izma93] considers a system (in particular, a multiplexer) with a PBS mechanism where two main classes of traffic coexist: VBR video traffic and CBR traffic. For the first class of traffic, a Markov modulated fluid model is applied. The work developed gives exact expressions for the loss probabilities and delays; it is also possible to determine the buffer size requirements of the multiplexer. In [Meye93], the model used to evaluate the loss and delay performance of an ATM switching element is a finite state Markov process; the priority mechanism applied is again the PBS scheme. This paper then uses the analytical results to dimension the buffer capacity and the threshold level of the queue, in order to obtain the cell loss probabilities required by each type of traffic; it is also shown that the loss performance and the utilisation of resources by the switch improve in presence of a priority mechanism like PBS.

Until now, all the references in this thesis to the PBS mechanism have been considering it as handling only two priority levels for cells belonging to a traffic source; this is because only one bit in the header of ATM cells is available as a priority indicator. However, systems with more than two levels of priority, the so called *nested threshold discarding systems*, have been studied and search techniques have been devised to determine the set of thresholds that can maximise the offered load, for example in [Petr91]. Also, the work described in [Elwa92] and [Elwa94] considers the analysis of a stochastic fluid model with loss priorities (using in this case, the PBS mechanism) for Markov Modulated fluid sources; the study comprises both the case where only two levels of priority are present and the more general situation, for an arbitrary number of priority levels. One of the characteristics of this analysis is that it provides a complete delay distribution for each priority class.

3.9. Conclusion

This Chapter describes ATM aspects with the objective of showing the relationship between diverse functions like CAC and UPC, as well as indicating the issues that this thesis addresses; at the same time, the Chapter tries to give an idea of the state-of-the-art in terms of standards for ATM.

The Chapter starts by comparing the characteristics of current transfer modes (such as STM with circuit-switched networks) with ATM including the basic *building block* in ATM (the cell).

The classification of traffic into categories gives an indication of the main characteristics present in each traffic type; this includes examples of the applications that fall in each of the traffic categories. This thesis only considers *guaranteed traffic*, particularly VBR traffic; this is explained by the fact that the idea of having non-guaranteed (i.e., ABR and UBR) traffic types in ATM networks only appeared half way through the development of the research work.

Another aspect that this Chapter describes is the role of QoS in ATM networks and the parameters used to describe that measure. Of the QoS parameters mentioned, the present research work has concentrated on the *cell loss ratio*.

Then, the two main traffic control mechanisms, CAC and UPC (intimately related to each other), are discussed. Attention is paid to the existing standards (from both the ITU and the ATM Forum) and the various algorithms available to execute those mechanisms. The UPC function depends on the CAC to gain a knowledge of the traffic that it monitors, so as to be able of taking actions when the traffic appears to misbehave. The research work described in this thesis assumes that the CAC function is performing well and then analyses *UPC* and its possible actions in the presence of misbehaving traffic; the UPC mechanism used is the *Leaky Bucket*.

Finally, the last important ATM aspect described in this Chapter concerns *priority mechanisms*, their advantages and their drawbacks. Several approaches, comprising time and space (or loss) priorities, are discussed. This thesis makes use of the loss priority mechanism known as *Partial Buffer Sharing*, considered to be "... the only likely candidate for implementation ..." (see [Grav91b]), due to its compromise between simplicity and accuracy.

4. Modelling and Simulation

One of the most important objectives of ATM is to provide a platform for broadband networks that allows the transport and transfer of information that may originate from traffic sources with different traffic characteristics (e.g., bandwidth requirements, delay and error rate). Therefore, it becomes vital to have some form of representation of the various traffic sources supported by the services that a network supplies, so that it is possible to guarantee that the network optimises the usage of the available bandwidth, avoids congestion and maintains the QoS of the users.

A model that can describe the behaviour of a real traffic source (with a reasonable degree of accuracy) is an invaluable input to the simulation of the behaviour and QoS requirements of that source; in ATM systems, a source model describes the arrival process of cells.

The simulation of traffic sources relies on the right choice of a model to obtain correct results; in fact, even if the simulation technique is very good, the wrong choice of a source model to describe a real traffic source will produce wrong results. Apart from choosing the right traffic model to describe the behaviour of a source, the choice and quantity of traffic parameters is also important (see Section 4.1.4), as it can determine the degree of complexity imposed on the analysis of a traffic source. The traffic models used throughout this thesis are the Interrupted Deterministic Process (or On/Off model) and the Generally Modulated Deterministic Process (GMDP), defined in Section 4.1.3.

Simulation is one of the performance evaluation methods for ATM networks; *measurement* and *mathematical analysis* are the other two methods available. Simulation has been used in this thesis, as measurement methods require real ATM networks to be available for experimentation and mathematical analysis uses sophisticated traffic models that can sometimes lead to a very complex mathematical analysis; besides, this last technique is very restrictive for most real-time systems. Also, simulation is fundamental for validating results obtained by using approximate analytically tractable models. Simulation aspects are treated in Section 4.2.

4.1. Traffic Source Modelling

Stochastic processes and queuing models have been widely used in telecommunications for the most diverse applications, such as characterisation of real traffic sources, evaluation of network performance and network link dimensioning. They can be used in two ways: 1) as part of an analytical model; 2) to carry out a discrete-event simulation (described in Section 4.2).

In [Rama91], three queuing models are used to investigate the performance of an ATM switch as a function of the traffic model applied to its input lines:

a) a discrete time process with an associated four-state Markov chain (comprising two silent states and two active states, whose durations are geometrically distributed) that explicitly incorporates periodicity in the packet streams;

b) a continuous time MMPP process with two states;

c) a continuous time MMPP process with four states.

The study concludes that for traffic sources with very small interarrival times, the resulting cell loss probabilities are quite high and therefore, for traffic sources that are very loss sensitive, the desired QoS can only be achieved in lightly loaded networks.

The analysis of a DMAP/G/1 queuing model is studied in [Brie91]; the exact cell loss probability formula is derived and a special case of this stochastic process (the MMDP model with three states) is applied to study the characteristics of the Leaky Bucket algorithm in ATM networks. Studying the performance of different ATM systems where the input traffic results from the superposition of diverse traffic sources with different traffic characteristics is the purpose of the work developed in [Syka91a] and [Syka91b]. Two source models are used: the On/Off model with exponentially distributed *On* and *Off* states and the MMPP process. It is found that the latter model performs very well, except for low traffic load conditions, while the On/Off model is more suitable for heavy traffic load conditions.

[Dron91] uses the nD/D/1 queue to model a multiplexer for CBR traffic and calculates the corresponding delay distribution; the queuing model M/D/1 is applied as an approximation of the former model.

Other research work in the area of modelling can be found for example in [Magl88], [Guer91], [Pan91], [Robe91b] and [Stav92].

The next Sections are dedicated to traffic modelling, starting with the identification of the several levels at which the traffic generated by a source (or set of sources) can be modelled and analysed. The general behaviour of the main source types is then briefly reviewed, as well as the criteria that usually determine the choice of a particular model. Some examples of the most common traffic models are also presented.

4.1.1. Hierarchical View of Traffic

Published work (see for example [Hui88] and [RACE91]) has suggested that the traffic carried by a network can be considered in one of five *resolution levels in time*:

* *calendar level*

  The way in which a traffic source varies in time (either daily, weekly or seasonally).

* *connection/call level*

  The behaviour of a traffic source on a VC basis (e.g., the typical duration of a connection can vary between 100s and 1000s).

* *dialogue level*

  It represents the interaction between voice or data agents at both ends of a connection; in this case, the duration is about 10s.

* *burst level*

  At this level of resolution, the analysis is performed over the *bursts of cells* (i.e., cell rates that last for a particular time period during which the time between successive arrivals of cells is constant). For telephony, the active and silent periods have durations between 100 ms and a few seconds.

* *cell level*

  This level represents the statistical behaviour of cell generation, e.g., the minimum interval between two successive cells. In this case, each event is a cell. For example, in a typical connection, cells are transmitted over links with a bandwidth of 155.52 Mbit/s, which means that the interarrival time between consecutive cells is of the order of approximately 3µs.

Fig.4.1 shows the time scale between call, burst and cell levels in ATM; the research work described in this thesis has focused on the burst level.

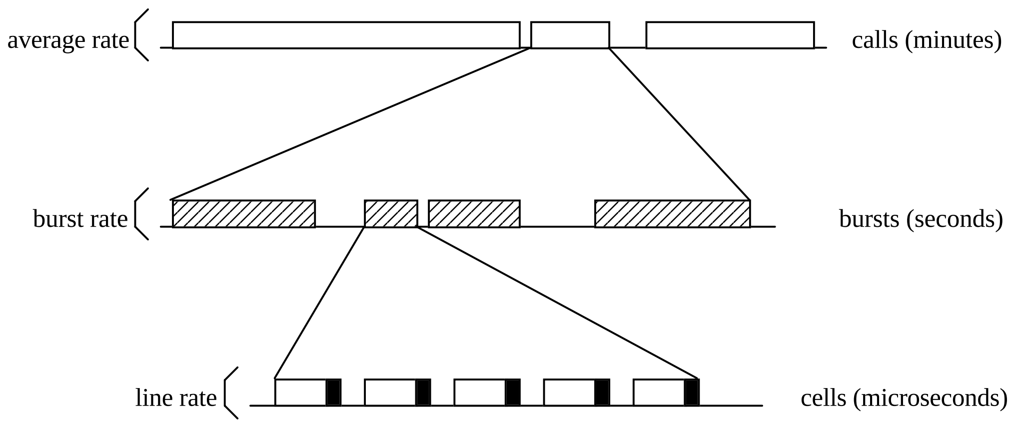


Fig.4.1 - Time scale for call, burst and cell levels.

The main levels at which a traffic source can be observed are the *call*, *burst* and *cell levels*. In fact, if a time period of several minutes is considered, then it is likely that a complete connection from say, a voice communication, may be observed. If the observation period is reduced to a few milliseconds, then a few bursts of activity and of inactivity will be detected. When the observation period lasts no more than a few microseconds, only a few cells from a burst will probably be observed.

At the higher level just considered (i.e., the *call level*), it is possible to determine the probability of being able to access the network (which is analogous to the probability of a call not being accepted by a telephony network because of not enough available resources). The traffic parameters used are the arrival process of the calls and a description of the call holding-time process. At the lower levels (i.e., *burst* and *cell levels*), the traffic parameters of interest are the burst length and corresponding distribution, as well as the statistical measures of cell arrivals in a burst (e.g., cell blocking and cell delay).

The analytical treatment of a source model does not usually take into account more than one or two hierarchical levels at the same time; this is because the analysis becomes quite complex (see [Kühn95]).

4.1.2. Behaviour of Traffic Sources

Traffic sources may be divided into three main classes, according to their behaviour within the network and type of information generated: voice, data and video sources.

The statistics of a single *voice source* are composed of two phases and they normally depend on the technique of voice coding that is being used. The two phases are the *active period* and the *silent period*. It is possible to model the arrival rate of the events (i.e., cell arrivals) approximating it by a Poisson arrival process; the resultant arrival process represents an Interrupted Poisson Process (IPP), which is described in Section 4.1.3. This is a Poisson process that is alternatively turned *On* for an exponentially distributed period of time (the so called *active period*) and turned *Off* for another independent exponentially distributed period of time (the so called *silent period*). During the active period, the interarrival times of packets (or cells, in the specific case of ATM) are exponentially distributed, while no packets are generated during the silent period; all processes are assumed to be mutually independent. When integer interarrival times are considered, the Geometric distribution can be used to approximate the duration of each period. On the other hand, the arrival of the packets may be approximated by a Bernoulli distribution. As a consequence, the process (which is bursty) will easily be modelled by a Markov chain with two states.

Data traffic tends to behave in bursts of constant bit rate with an exponentially distributed duration followed by an exponentially distributed silence period. The Poisson arrival process may be used to model the data generated by a single data source for the continuous time case; the Geometric interarrival process is used for the discrete time case. When information loss occurs, this type of service uses retransmission as a way of recovering information. The retransmission of the complete data frame is executed every time there is cell loss, no matter how severe the loss itself is. In the case of interactive data transmission, cells may be generated one at a time while for bulk data transmission, (e.g., file transfer) a large number of cells may be generated at a time (also called *batch arrivals*).

Video traffic can be divided into still pictures and moving pictures. This type of traffic usually has severe QoS requirements in terms of cell loss rate and cell delay jitter. Services based on VBR video are less bursty than data services and error recovery can be performed by using Forward Error Correction (FEC). However, video sources containing scenes with high motion will generate traffic that is both bursty and has high bit rate (see [Habi92]).

Video services may be classified into:

– *communication video services*

In this case, two or more parties use video links for an interactive conversation.

– *distribution video services*

For these services, several destinations receive a video sequence (they usually require a higher bit rate than the communication video services).

Video images can be statistically characterised by four components that are dependent on the type of codec:

* *line correlation* (or spatial correlation)

  It occurs when data at one part of the image is highly correlated with data on the same part of the next line.

* *frame correlation* (or temporal correlation)

  It means that data at one part of an image is highly correlated with data on the same part of the next image.

* *scene correlation*

  It occurs because sequences of scenes may, to a greater or less extent, be coincidentally correlated with each other.

* *white noise*

  It is a memoryless process and there is no correlation associated with it.

For example, non-frame buffered video codecs (i.e., codecs for which the frames are not buffered) have all four of the correlations, whilst frame buffered video codecs (i.e., codecs that always have their frames buffered before being sent) only have scene and white noise correlations, of which the former can be reduced by buffering multiple frames. The information rate provided by CBR and VBR video codecs may vary between 8 kbit/s and 140 Mbit/s. Their information rate and statistical characteristics are dependent on the spatial and temporal sampling rate as well as on the type of video codec that is used.

4.1.3. Models for Traffic Sources

Traffic source models are usually divided into two categories, according to the way they consider the variable *time*, i.e., either continuously or in a discrete manner. In the first case, the slotted nature of constant cell transfer events (i.e., cell arrivals and departures) is ignored; models with this characteristic are called *continuous time source models*. The models (such as the Bernoulli process, described later) belonging to the latter class are named *discrete time source models*.

In what follows, a description is given of some continuous and discrete time traffic source models. Let λ represent a source's arrival rate and T be the random time between arrivals. If the interarrival times are assumed independent (i.e., not correlated) and follow an Exponential distribution and t represents the variable time, then

$$\mathrm{A}(\mathrm{t}) = \mathrm{Prob.}\{\mathrm{T} \le \mathrm{t}\} = 1 - \mathrm{e}^{-\lambda \cdot \mathrm{t}}, \quad \mathrm{t} \ge 0 \tag{4.1}$$

is a *Poisson Process*. Therefore, the number of events that occur in the time interval [0,t] has a Poisson distribution. This process is said to be *memoryless* because the following property is verified for any exponentially distributed variable X (see [Çinl75]):

$$\mathrm{Prob.}\{\mathrm{X} > \mathrm{t} + \mathrm{s} \mid \mathrm{X} > \mathrm{t}\} = \mathrm{Prob.}\{\mathrm{X} > \mathrm{s}\}; \quad \mathrm{t},\ \mathrm{s} \ge 0\,. \tag{4.2}$$

In other words, knowing that an interarrival time has lasted already t units of time does not change the probability of that interarrival time lasting another s time units. Another analytical property of a Poisson process is the fact that the superposition of independent Poisson processes is still a Poisson process where the total arrival rate is the sum of the component rates.

Poisson processes can be used to describe the superposition of many traffic sources, when none of the sources dominates, but they are not adequate to model the cell level behaviour of individual sources with bursty characteristics. This type of process has been used mainly for buffer dimensioning in switch fabrics (see [Robe91b]).

A *Markov Modulated Poisson Process* (MMPP) is a double stochastic Poisson Process where the function $\lambda(t)$, representing the arrival rates over time, is controlled by another stochastic process with a finite number of states {1,2,...,m}; $\lambda(t)$ is called the *intensity function* and it represents the derivative of the so called *leading function* $\Lambda(t)$ that defines the average number of events over the interval [0,t]. The modulating process is a finite Markov chain with an infinite generator Q, such that:

$$\begin{cases} Q = (q_{ij}) \\ q_{ij} = \text{transition rate from state } i \text{ to state } j \ (i \neq j) \\ q_{ii} = -\sum_{i \neq j} (q_{ij}) \end{cases} \tag{4.3}$$

The diagonal matrix $\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$ represents the intensity functions of the state dependent Poisson Processes. The next figure (Fig.4.2) shows a schematic diagram of an MMPP with two states, where λ_i is the arrival rate for each state and $1/r_i$ corresponds to the mean sojourn time in the *ith* state (i =1,2).

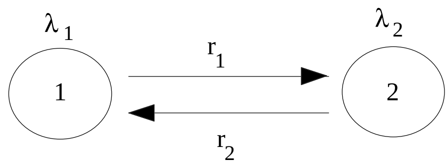


Fig.4.2 - An MMPP model with two states.

When an MMPP process is considered with *m=2* (i.e., the parameters are q_{12}, q_{21}, λ_1 and λ_2) and either λ_1 is null or not (with the other arrival rate λ_2 having a non null or null value, respectively), the resulting process is called an *Interrupted Poisson Process* (IPP) (see [Yama91b]). This type of process models the bursty nature of sources that are represented by two states (one *On* state, where the source is active and one *Off* state where the source is inactive). This is also the case of the *Fluid Flow Approximation Model* (first described in [Anic82]), which replaces the discrete cell stream by a continuous flow of information where the flow intensity is typically modulated with a Markov chain or an On/Off mechanism; this type of model removes the cell-level details and gives accurate results when the traffic considered varies in large time scales (e.g., for traffic sources like voice and video). A special case of the fluid flow model is the fluid flow version of the Talkspurt/Silence model, which is a particular case of the GMDP model (defined later in this Section). In this case, the source is defined by

* the information flow rate (or cell rate) in the active state, that corresponds to the inverse of the constant interarrival time in the original talkspurt/silence source model;

* the mean burst duration;
* the mean silent period.

Another continuous time model is the *Renewal Process* (also called a *General Independent Process*), which is characterised by statistically independent and identically distributed interarrival times. Since there is no correlation between the interarrival times, the process is completely defined by its probability distribution. The interarrival times can be determined in two ways:

* measuring the relative frequency of the interarrival times and using this as an approximation for the interarrival time distribution;
* deriving the corresponding interarrival time distribution, given the autocovariance function of the number of arrivals at successive time instants.

The Bernoulli (defined later in this Section) and Poisson processes are special cases of a General Independent Process with geometrically distributed interarrival times and exponentially distributed interarrival times, respectively.

In mathematical terms, the sequence $S = \{S_n;\ n = 1,\ 2,\ ...\}$ of successive occurrences of a given phenomenon is called a Renewal Process (see [Çinl75]) if the times $W_1, W_2, ...$ between the successive occurrences of S are independent and identically distributed non-negative random variables (where the term "identically distributed" means that the interarrival times follow the same probability distribution). Moreover, the following relation is verified between the renewal times S_n and the interarrival times W_n:

$$\begin{cases} S_0 = 0 \\ S_{n+1} = S_n + W_{n+1};\ \ n = 1,\ 2,\ ... \end{cases} \tag{4.4}$$

The source models described so far belong to the category of continuous time source models. The models referred to next reflect the slotted nature of cell arrivals; thus, they are called discrete time source models. The first example is that of a *Bernoulli Process*; at each slot interval, there is a cell arrival with probability *p* and no cell arrival with probability *q=1-p*. Since the occurrence of a cell arrival is statistically independent of any previous arrivals, the process is memoryless. The interarrival time of two successive cells of such a process has a Geometric distribution with a minimum of one time slot *Δt*. Mathematically, if $\{X_n;\ n = 1,\ 2,\ ...\}$ is a stochastic process with probability of success *p* and $\{X_n\}$ demonstrates the conditions,

a) the random variables $X_1, X_2, \ldots$ are independent and they only take the values *0* and *1*;

b) $\text{Prob.}\{X_n = 1\} = p$, $\text{Prob.}\{X_n = 0\} = q$, for all *n*; $p + q = 1$;

then, $\{X_n\}$ is a *Bernoulli Process* that exhibits the following properties (see [Fros94]):

– the time between arrivals is geometrically distributed (given that *T* represents the random interarrival time), i.e.,

$$\text{Prob.}\{T = i \cdot \Delta t\} = (1 - q)^{i-1} \cdot q;\quad i = 1, 2, \ldots \tag{4.5}$$

– the number of arrivals in each time slot is binomially distributed.

This model is completely characterised by its mean interarrival time; since it does not include any correlation, it will not represent individual ATM traffic sources with sufficient accuracy, because they usually generate cell streams that are modulated at burst level and are therefore highly correlated. Another characteristic of this model is that it is also a renewal process, since the interarrival times are statistically independent and identically distributed.

On the other hand, the *Generally Modulated Deterministic Process* (GMDP) is a double stochastic point process; it is based on a finite state process having *m* states (see [RACE92]). In each state, cells are generated with constant interarrival time, d_i, between successive cell arrivals in state *i*,

$$d_i \cdot \Delta t = \frac{1}{\lambda_i};\quad i = 1, 2, \ldots, m \tag{4.6}$$

with λ_i representing the arrival rate in the *ith* state. The number of cells X_i that are generated in state *i* may have a general discrete distribution

$$f_i(k) = \text{Prob.}\{X_i = k \cdot \Delta t\},\quad i = 1, 2, \ldots, m;\quad k = 1, 2, \ldots \tag{4.7}$$

Usually, the GMDP model includes also silence states where no cells are generated and the duration of these states may also have a general discrete distribution. The state changes of the underlying state process are governed by a *m×m* transition matrix $P = (p_{ij})$, where p_{ij} is the probability that the source moves to state *j* at the end of its sojourn time in state *i* (i≠j).

Fig.4.3 represents an example of a GMDP model with three states, where λ_i is the constant cell rate in the *ith* state (i = 1, 2, 3). In most situations, voice traffic sources can be characterised when using this model with two states, whilst video traffic sources need at least three states to be characterised (note however that with this model, it is not possible to obtain an unbiased estimation of the source traffic parameters - see [Cosm90]). In this thesis, the GMDP process has been used with geometrically distributed state durations (see Chapter 5).

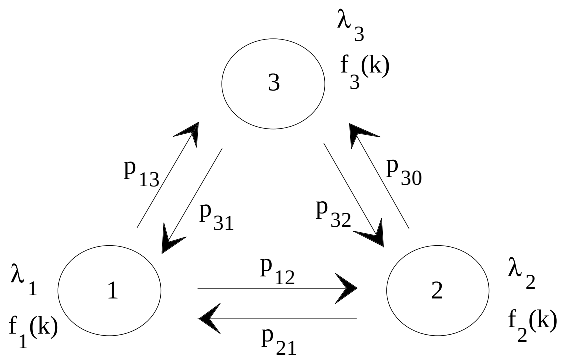


Fig.4.3 - GMDP model with 3 states.

If the modulating process of the GMDP model is a finite discrete time Markov chain, then the process is named *Markov Modulated Deterministic Process* (MMDP); thus, the sojourn times of the modulating process' states are geometrically distributed. This type of process can be used to model the changes of distributed video at burst level (see [RACE91]); at cell level, it is used to model the cell interarrival pattern of traffic sources. Since the MMDP is a non-periodic Markov model, there is independence between the probability of any future state and past events; this probability depends only on the present state of the process. Hence, it is not suitable for modelling structures that exhibit correlation, such as non-frame buffered video codecs.

Another type of discrete time model is the *Interrupted Deterministic Process* (also called in the literature by *On/Off Source Model*, *Burst/Silence Model* or *Talkspurt/Silence Model* due to its application in the modelling of packetised voice - see [Srir86] and [Heff86]), which consists of an MMDP model with two states, alternating between phases of activity (i.e., a *talkspurt* or *burst state*) and silence phases (see [Hong92]). Within a talkspurt, cells are emitted with constant interarrival time. The model has cell arrival rates $\lambda_1 \neq 0$ and $\lambda_2 = 0$. In this case, the modulating process can be described by the two parameters α and β of the Geometric distributions for the *On* and *Off* states, respectively. Thus, they verify the following relations:

$$f_1(k) = (1-\alpha)^{k-1} \cdot \alpha \quad \text{and} \quad f_2(k) = (1-\beta)^{k-1} \cdot \beta; \quad k = 1, 2, \ldots \tag{4.8}$$

Also, the average duration of the *On* and *Off* states is $\Delta t/\alpha$ and $\Delta t/\beta$, respectively; e.g., with the voice application, the *On* state has typically a duration between 0.4 seconds and 1.3 seconds, while the *Off* state usually lasts between 0.6 seconds and 2 seconds (see [Schw95]).

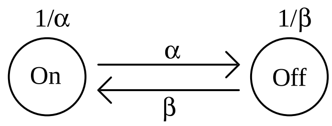


Fig.4.4 - The talkspurt/silence model.

Some of the research work described in this thesis makes use of the Interrupted Deterministic Process (IDP) process with both exponentially and geometrically distributed state durations. Finally, since both *On* and *Off* states are Geometrically distributed, the modulating process can be viewed as (see Fig.4.4):

– if the source is in the *On* state, then it will continue in that state (after each time slot) with a probability $1-\alpha$ and it will change to the *Off* state with probability α;

– if the source is in the *Off* state, then it will continue in the same state (after each time slot) with a probability $1-\beta$ and it will change to the *On* state with probability β.

On the other hand, the *Discrete Time Markovian Arrival Process* (DMAP) consists of a discrete time stochastic process that is based on a discrete time Markov chain with *m* states. In this model, the transition rates are governed by a transition matrix $P = (p_{ij})$; the probabilities p_{ij} are formed by the probabilities c_{ij} (relative to no cell arrivals) and the probabilities d_{ij} (relative to cell arrivals) (see [Kühn95]); the cell arrivals are only generated at the time instants of state transitions and the number of arrivals depends on the particular state transition.

Finally, the *Multi-Minisource Model* (see [Magl88]) represents the superposition of n identical and independent talkspurt/silence sources (see Fig.4.4) in a single fluid flow model and it represents a special case of the GMDP; the duration of the *On* and *Off* states is exponentially distributed with mean durations of $1/\alpha$ and $1/\beta$, respectively, and thus, α and β represent the transitional rates. The model (i.e., the multi-minisource model) is based on a continuous time Markov chain with $n+1$ states describing the number of sources that are currently active (i.e., if the system is in state i, then this means that i sources - of the total n sources - are active); also, the arrival rate in state i is given by ia, where a is the information flow rate. The next figure (Fig.4.5) shows a diagram of this model, with the transition rates between any two adjacent states. [Magl88] has shown that a video source can be modelled as a number of identical On/Off traffic sources.

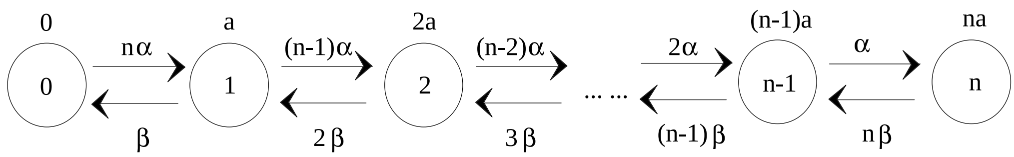


Fig.4.5 - The multi-minisource model.

A considerable amount of research in teletraffic engineering has been devoted to find adequate models to describe traffic sources in ATM networks. Table 4.1 shows some source models and their possible application in the characterisation of real traffic services.

Services	***Source Models***
CBR services	renewal process
interactive data (low speed); LAN interconnection	GMDP; interrupted deterministic process
videotelephony (frame buffered); videoconference (frame buffered)	renewal process + multi-minisource model
video (high quality); TV; HDTV	GMDP + multi-minisource model

Table 4.1 - Models suitable for each service.

The analysis of several published papers shows that it is possible to find a source model for every ATM source, except for non-frame buffered videotelephony and videoconference. Other published work about traffic source modelling can be found in [Çinl75], [Cosm90], [Syka91a], [Syka91b], [Lee92] and [Suth92].

Once a model (or stochastic process) has been chosen to describe a certain traffic source (according to the criteria defined in Section 4.1.4), in what way is it used in a telecommunication or computer system? The usual approach (see [Kühn95]) consists of four main steps:

1. use of the stochastic process to model the arrivals of the source(s) in the system;
2. model the behaviour of the system considered (e.g., a link or a network) by using a service system that consists of components like servers and queues;
3. model the operation of the system by using queuing disciplines (like FIFO, referred to before);
4. carry out a performance analysis of the system state process, queuing processes and obtained QoS characteristics.

The fourth step above is executed through the use and study of queuing models (see Table 4.2) based on the source models described earlier. More traffic source models are proposed in [Kouv94b], [Kouv95b] and [Lin91b].

Hierarchical Level	***Queuing Models***
cell level	M/D/1, GEO/D/1, ΣD_i/D/1, ΣM_i/D/1
burst level	Fluid Flow Approximation
combined burst and cell level	MMPP/D/1, IPP/D/1, MMDP/D/1, IDP/D/1, DMAP/D/1

Table 4.2 - Queuing models for ATM systems (from [Kühn95]).

The mathematical shorthand notation for queuing models - introduced by D. G. Kendall (see [Mitr87]) - is in general as follows: for a *X/Y/Z* queuing model, *X* represents the nature of the arrival process (e.g., *M* designates Poisson streams and *GEO* designates Geometrically distributed arrivals), *Y* describes the distribution of service times (e.g., *D* represents deterministic or constant service times) and *Z* denotes the number of servers; if the considered queue is of finite size, then a fourth parameter is added to the abbreviated notation to represent the capacity of the queue.

The outcome of carrying out procedures 1. to 4. can then be applied to the dimensioning of system resources (e.g., number of servers and buffer capacities) and in network planning (taking into account data relative to load and QoS).

4.1.4. Selection of a Source Model

The selection of a source model for the characterisation of real traffic sources must be performed according to some criteria; this section will consider some of them.

Important qualities of a source model are *accuracy*, *closeness to reality* and *physical meaning*. The second quality is fundamental as the results given by the model should describe as much as possible the reality. Also, source models should always generate positive random variables because there can never be a negative cell interarrival time or a negative number of arrivals in an interval.

Another important feature of source models is *generality*. This characteristic represents the capability of the model to cover a large class of sources in order to model a wide range of traffic streams with different characteristics.

It should also be possible, given a source model, to *use it in simulations*; this is possible when the model is stable from the point of view of statistics. The statistical stability is measured over a period of time that is proportional to the highest level of resolution in time specified by the source model and the number of different states of the model.

From the analytical point of view, a model should also be *tractable*, which means that the source model leads to solutions that can be used for numerical computation. In many cases, general methods such as iteration methods to solve large systems of linear equations, aggregation methods to reduce the dimensionality or matrix analytical methods (as described in [Rama91] and [Syka91a] for the MMPP/G/1/k queuing model) could be applied to solve structured Markov chains, but often, the exploitation of the special structure of the processes involved may make the model much more suitable for numerical solutions, without losing its probability interpretation.

All the statistical models are required to be parameterised; hence, the *method of model parameterisation* is another important criterion for selection. The preferred method is unbiased estimation of the parameters for a given model. In this method, one or more equations of the model's parameters (e.g., the mean, variance, autocovariance and probability distribution) are equated to the corresponding measured ones to solve for the model's parameters. However, sometimes it is not possible or not applicable to determine the model's parameters using unbiased estimation; in this case, they will have to be directly measured.

Finally, another criterion to take into account is the *number of parameters* of the model. In fact, as the number of parameters is directly related to the complexity of the model's description and hence to the dimensionality of the model, a limited number of parameters is required and advisable.

4.2. Simulation Methods

As mentioned in the beginning of this Chapter, simulation is a widely used approach to study complex systems, especially when measurement and analytical methods cannot be applied; simulation is also important to help in the validation process of the results obtained by using approximate analytically-tractable models. When using simulation, it is not only possible to study a system over a long time frame in compressed time but also to maintain a better control over the experimental conditions than when handling the real system. However, there are several disadvantages with simulation:

1) the (sometimes high) computation time required;
2) the (sometimes) small degree of accuracy when simulating rare events, such as cell losses;
3) the fact that each run of a simulation model produces only estimates of the true characteristics of a model, for a given set of input parameters.

To avoid this last disadvantage and in the case where a valid analytical model is available, recourse should be made to analysis instead of simulation. The second drawback can be minimised by making use of special simulation techniques, called *accelerated simulation techniques,* which are described in Section 4.2.2. The research work contained in this thesis has used one such technique, called *cell-rate* (or burst-level) *simulation technique*.

Simulation has been used by several researchers in ATM studies. [Mant91] uses a cell-level simulator to assess the impact of broadband traffic mixes (where the traffic sources are modelled as GMDPs) on buffer occupancy and compares the results with the case where a renewal input traffic is assumed; the authors conclude that in the case of high burstiness or a large value of the ratio burst-length to buffer-size, the tail of the buffer occupancy distribution is significant and cannot be modelled by assuming renewal input traffic. Burst-level simulation is used in [Sun92] to study delay, delay jitter and cell loss in ATM networks. In [Sun93], a burst level simulator is applied to the study of the multiplexing and demultiplexing performance of an ATM switch fabric. Cell-level and burst-level simulation methods for ATM networks are compared in [Pitt91b], where the traffic sources are again modelled as GMDPs. Burst-level simulation proves to be a good alternative (in terms of accuracy and speed in obtaining results) to cell-level simulation, especially when the network utilisation is high and when the buffer capacity and the number of traffic streams multiplexed in the system are not of the same order of magnitude. [Pitt90] and [Pitt91a] show how burst level (or cell-rate) simulation can be applied in modelling several telecommunication topics such as UPC, CAC and in emulation for network management studies. It also points out that the usage of burst-level simulation implies a trade-off between the level of detail of the traffic source characterisation and the speed of the simulation; however, this does not compromise the accuracy of the simulation measurements. A simulation study of congestion and flow control techniques is carried out in [Fisa91].

4.2.1. Basic Concepts

Simulation is an important tool for the performance evaluation of communication networks and systems. In the following sections, the focus will be on some simulation modelling aspects (such as the use of discrete event simulation, random number generators, confidence intervals and validation techniques) that help making the best use of the output results obtained from simulation experiments.

4.2.1.1. Discrete Event Simulation

In a telecommunications network, users generate demands for network resources and protocols control the allocation of resources in order to satisfy those demands. This can be described by making use of *discrete event simulation*, which models a system as it evolves over time, using a representation in terms of *state variables* to describe the system at particular points in time. These points are the time instants at which an event occurs, an *event* being an occurrence that may cause a change in the state of the system (e.g., the arrival of demands for network resources in a communications network simulation). Events are the fundamental elements in discrete event simulation; they may change the state of the system or cause an action concerned with measurement, monitoring or the progress of the simulation.

A discrete event simulator is formed by several functional components (variables or procedures) that aid the coding, debugging and future alterations in the computer program of a simulation model. The *system state* represents the set of state variables necessary to describe the system at a particular time. The current simulated time of an experiment is given by a variable called *simulation clock* whose value changes by random increments. A list, called *event list*, is used to store the next time when each type of event will occur and it contains events sorted in chronological order. Statistical information about system performance is stored in variables called *statistical counters*; they allow the estimation of performance measures that are computed by the *report generator* and are shown when the simulation ends. The *initialisation routine* is a subprogram used to initialise the simulation model at time zero. Another routine, the *timing routine*, calculates the next event from the event list and then advances the simulation clock to the time when that event is due to occur. The system state is updated by the *event routine* when a particular type of event occurs. *Library routines* generate random observations from probability distributions (determined before as part of the simulation model). Finally, the process of calling the timing routine to determine the next event and then transferring the control to the corresponding event routine to update the system state is executed by the *main program*.

Discrete event simulation has been applied in telecommunications networks as a basis for building simulation tools (see [Fros94]). In particular, this method has been used in ATM based networks, where the basic unit of traffic is a cell; it is then called *cell level simulation* (see also Section 4.1.1). Each cell arrival represents an event and cells are dealt with independently when they arrive to a network queue. This technique (which can be found in many network simulators) is described in detail in [Law91, pp.7-13].

4.2.1.2. Random Number Generators

The simulation of any system in which there are random components (such as the duration of calls in a telecommunications network) requires a method of producing numbers that are random. In other words, the method should be able to generate sequences of numbers *x*, such that $x \in [0,1]$, taken from the Uniform distribution and which appear to be statistically independent.

Random Number Generators (RNGs) should be fast and produce sequences of uncorrelated random numbers with long periods. RNGs should also allow the exact reproduction of a given stream of random numbers, as this can sometimes aid in the debugging or verification of computer programs using random numbers. The *period* of a RNG represents the length of a generated sequence of numbers before it repeats. Having a RNG with a long period is of utmost importance because when a sequence starts repeating itself, correlations will begin to appear in the results.

The simulation tool used in this research work makes use of a Wichmann-Hill algorithm (see [Pitt93, pp.65-66]), which has a period of about $7 \cdot 10^{12}$; this is sufficient for an application to ATM studies that use a cell-rate simulation method, as is the case of the present research work. In fact, because random numbers are applied to events and these represent many cells in a burst when used in cell-rate simulation (so that a small number of events can simulate many cells: the reason why it speeds up simulation), a period of $7 \cdot 10^{12}$ is enough to simulate cell losses as small as of the order of 10^{-10}.

4.2.1.3. Confidence Intervals

A *confidence interval* gives a measure of the reliability from the results of a simulation. For example, when a simulation is performed with a 95% confidence interval, this means that for 95 out of 100 simulations runs where an interval is calculated, the exact value of the measure of interest (e.g., the mean value of a variable) belongs to the calculated interval.

There are several methods to calculate confidence intervals from measurements obtained with simulation experiments. One of them is the *method of independent replications*, which manipulates *n* estimates obtained from *n* independent simulation runs. The *batch means method* divides one single simulation run into n_b batches (where each batch consists of a fixed number of observations, hereby denoted n_{obs}) and then it determines n_b estimates from those batches. The choice of the fixed number n_{obs} is important, as it is closely related with the correlation between batches. This was the method applied in the cell-rate simulator used for the present research work.

On the other hand, the *regenerative method* uses a single simulation run, but it depends on the definition of a state (that is usually difficult to obtain) after which the process repeats itself in a probabilistic way.

From statistical theory, it is known that, given a set of Normal distributed random variables X_i (i = 1, 2, ...) with mean value μ, the random variable t_n represented by

$$t_n = \frac{\overline{X}(n) - \mu}{\sqrt{\frac{S^2(n)}{n}}} \tag{4.9}$$

is t-Student distributed (see [Law91, pp.287-292]), with *n-1* degrees of freedom (where $\overline{X}(n)$ is the sample mean, $S^2(n)$ is the sample variance and *n* is the number of observations). In this case, a $100 \cdot (1-\alpha)$ percent confidence interval for the real value of μ is given by (with $0 < \alpha < 1$),

$$\mu \in \left]\overline{X}(n) - t_{n-1,1-\alpha/2} \cdot \sqrt{\frac{S^2(n)}{n}}, \overline{X}(n) + t_{n-1,1-\alpha/2} \cdot \sqrt{\frac{S^2(n)}{n}}\right[\tag{4.10}$$

where $t_{n-1,1-\alpha/2}$ is the upper *1-α/2* critical point for the t-Student distribution with *n-1* degrees of freedom; in other words, $t_{n-1,1-\alpha/2}$ verifies the following condition:

$$1 - \frac{\alpha}{2} = \text{Prob.}\left\{T_{n-1} \le t_{n-1,1-\alpha/2}\right\}, \tag{4.11}$$

T_{n-1} being a t-Student distributed random variable with *n-1* degrees of freedom. In this thesis, the confidence intervals obtained from the cell-rate simulator's batch means method (see Chapter 5) consider

$$n=40 \quad \text{and} \quad \alpha=0.01,$$

which means that results are obtained with 99% of confidence.

4.2.1.4. Simulation Validation Steps

After a simulation model is designed and implemented, either with the aid of a general-purpose language (e.g., Pascal and C) or with a simulation language (e.g., GPSS or SIMSCRIPT), it must be both verified and validated. In other words, it is necessary to debug the simulation program so as to ensure that it performs as intended (this is the *verification phase*) and to investigate whether the simulation model accurately represents the system being studied (this is the *validation phase*), as this shows that results obtained with the implemented simulation model should be representative of the real system.

[Law91, pp.298-300] describes several *verification methods* for debugging a simulation program and which can be used at each design stage; only four will be referred to here. One of the methods consists of using the *modular approach*, i.e., to write the simulation program as a set of modules for which errors can be more easily and gradually debugged. The second method considers a *variety of settings for the input parameters* and runs the simulation program under those assumptions; this allows the programmer to check whether the output is reasonable. *Tracing* is another technique often used; it consists of printing information about the simulated system (e.g., the contents of the event list and the state variables), after the occurrence of each event. The information thus obtained is then compared with hand calculations of what the simulation program should produce. Finally, one other method that can be applied consists of *running the simulation model under simplifying assumptions* for which the true characteristics are known (e.g., using an M/M/1 queue). The combination of these methods contributes to ensure that the simulation model is implemented in the correct way, i.e., it captures the true characteristics of the simulation model.

As mentioned before, the validation phase determines whether the conclusions taken for a real system (such as an ATM network) by using a simulation model are or are not reliable. To execute this task, a three-step approach has been proposed in [Nayl67]:

1. *development of a high face validity model*: this step consists of making sure that the model developed seems reasonable to people who are knowledgeable about the system represented by the simulation model, which implies making use of existing theory, experience, observations of the real system under study

and using results from similar simulation studies for comparison;

2. *empirical testing of the model assumptions*: sensitivity analysis is one of the techniques used, which consists of observing how much the simulation output changes when an alteration is introduced to an input parameter or in the level of detail of a subsystem; also, it is possible to check the adequacy of fit of a theoretical probability distribution which has been fitted to some observed data and used in the simulation model as input data;
3. *measurement of how representative the simulation output data are*: this is achieved by determining if the simulation results are close to the results expected from the real system being analysed; the comparison of simulator and real system can be performed by using a confidence interval approach, such as the paired-t or Welch confidence interval (see [Law91, pp.588-589]), where the difference between results from the two systems is considered to be statistically significant when the resulting confidence interval does not include the null value.

Note however that the difference between the real and simulated systems will only be significant in practice when that difference invalidates any conclusions about the system deduced from the simulation model.

4.2.2. Accelerated Simulation

Simulation allows complex systems to be studied in detail. However, it often requires large amounts of computer time to obtain results that have a small confidence interval. This is very important in ATM networks, as the values of interest for parameters like cell loss probability are very small (as low as 10^{-9}). This has lead to the development of techniques (see [Law91], [Robe92] and [Pitt93, pp.73-82]) that aim at accelerating the simulation of rare events, using different approaches. These techniques include three main classes of methods:

* *hybrid models* combine analytical models and simulation to increase the efficiency of a simulation;
* *variance reduction techniques* use statistical methods to obtain more accurate performance measures (e.g., the RESTART method, which has been specifically designed for ATM networks studies) and thus improve

computational efficiency;

* *extrapolative methods* use statistical methods to estimate the tail probability distribution outside the sample range.

Hybrid simulation techniques comprise methods such as *decomposition* (where the system is separated into several subsystems and where the solution involves the application of either theoretical models or simulation) and *reverse time model* (where the simulation results obtained are used as input to analytical models). The use of a hybrid simulation technique generally gives better results when a network must be simulated many times and the computational savings compensate for the effort put into the analytical technique.

Variance reduction techniques include *antithetic sampling*, *common random numbers*, *importance sampling* and the *RESTART method* (see [Alta91]). The first method takes two simulations that are run in parallel with the same input parameters but with complementary random number streams; this produces means for both simulations with smaller confidence intervals when the results are negatively correlated. The method of *common random numbers* is similar to antithetic sampling, except that the random number streams used are now the same for two or more simulations, in an attempt to have similar stochastic conditions. Little benefit will be gained when using either the antithetic sampling or common random numbers methods in the case of complex networks of queues, as it becomes difficult to obtain correlation between input and outputs. With importance sampling, some bias is introduced in an input distribution so as to increase the frequency of evaluation for rare events; the choice of the bias depends on the problem being analysed. Finally, the RESTART method (which is general enough to be applied to most simulation models) considers a rare event *A* (the event that is of interest in the simulation) and defines another event *C* which verifies both conditions

$$\mathrm{C} \supset \mathrm{A} \qquad \textit{and} \qquad 1 >> \mathrm{P}\{\mathrm{C}\} >> \mathrm{P}\{\mathrm{A}\}. \tag{4.12}$$

The probability of occurrence of event *A* is then

$$\mathrm{P}\{\mathrm{A}\} = \mathrm{P}\{\mathrm{C}\} \cdot \mathrm{P}\{\mathrm{A} \mid \mathrm{C}\}. \tag{4.13}$$

To estimate $P\{A\}$, it is necessary to estimate both $P\{C\}$ and $P\{A \mid C\}$; although it is possible to accurately estimate $P\{C\}$, it is not normally possible to estimate $P\{A \mid C\}$. The estimation of $P\{A \mid C\}$ is improved in RESTART with repetitive simulation trials of the time intervals in which the event *C* occurs.

When using extrapolative techniques, it is not necessary to make any modifications to the simulators. *Tail extrapolation* (used to dimension buffers for very low queue length probabilities) and *extreme value theory extrapolative estimation* are the methods included in this last category; the latter takes the obtained simulation results to estimate values beyond the simulated sample range.

Due to their characteristics, hybrid techniques can be considered as modelling techniques; the same can be said of the accelerated simulation technique used to develop the work described in this thesis (the *cell-rate simulation method*) as both techniques depend on the decomposition of a problem into parts. This last technique is an alternative model for the handling of traffic and the mechanism of queuing; it considers a hierarchical decomposition of the traffic (see Section 4.1.1) and only models the rate of flow of cells, instead of modelling the individual cells. Cell-rate simulation is able to simulate the same number of cells as a cell-level simulator, using less computing time; it does so by simulating many cells in each event. This number is determined by the characteristics of the traffic source. The method is fully described in [Pitt93, pp.83-121] and referred to briefly in Chapter 5.

4.3. Conclusion

The aim of this Chapter was to introduce some of the concepts related to simulation and modelling that are used by the work described later in this thesis.

On modelling traffic sources, a distinction was drawn between the various approaches to how traffic can be observed. Of the two most common approaches, burst level and cell level, the work described in this thesis focuses on the *burst level*. Then, the behaviour of voice, data and video type sources was described, followed by the most common traffic-source models (and some of their properties) used to characterise the behaviour of real traffic sources. Source models can be used both by simulators and by mathematical analysis. Of the source models indicated, two are used in this thesis: the *On/Off model* and the *GMDP model*. The On/Off model is used in a cell-rate simulator and in a fluid flow model (considering either exponentially or geometrically distributed *On* and *Off* state durations) and the GMDP model is used in a cell-rate simulator (considering geometrically distributed state durations). Finally, a set of main rules was given that helps in deciding on a particular traffic model.

Simulation aspects were also covered in this Chapter. First, the concept of discrete event simulation was described; the (adapted) technique is applied in the cell-rate simulator, taking into account the differences between working at cell and burst levels. The need for using random number generators with simulation was then explained and the *Wichmann-Hill random number generator* was indicated as the chosen algorithm for the cell-rate simulator described in Chapter 5. Confidence intervals and methods used to obtain them were also discussed; the cell-rate simulator used in this thesis uses the *batch means method* to obtain (and group) samples of information for each simulation run. With this information, the width of the confidence intervals for each measure in a simulation is then calculated with the aid of the *t-Student distribution*. The Chapter ended with a quick reference to the steps usually taken in the process of validating a simulator, followed by a classification of the several *accelerated simulation techniques,* of which the cell-rate simulation used in later Chapters of this thesis is an example.

5. Application of Partial Buffer Sharing (PBS) to an ATM Network Simulator

The ATM technique, agreed to be the switching and multiplexing platform for B-ISDN networks, is characterised by its flexibility. Indeed, it has been designed to support a great variety of services, with different QoS requirements. However, this flexibility implies solving problems such as management and congestion control, in order to optimise the utilisation of ATM networks.

One specific problem that arises with ATM is the necessity of minimising the cell loss from traffic sources in a network. If the most restrictive services are taken into account, the value suggested for an acceptable cell loss ratio is 10^{-8} (see [ITU89]) and several mechanisms have been developed so far in order to achieve that value, including peak rate allocation (see Sections 3.4 and 3.6.1) and priority schemes (see Section 3.8). The latter have the advantage of taking into account the different cell loss-rate requirements of the services by determining priority levels for given traffic sources and dealing with prioritised cells in an appropriate manner. This is possible through the cell loss priority bit in the header of ATM cells (see [ITU90a]).

This Chapter can be divided into five main parts, most of the material being new; the first part gives a brief overview of the cell-rate simulator used to implement a particular space priority mechanism, the Partial Buffer Sharing mechanism. The second part explains how the mechanism was added to the original cell-rate simulator. The next part considers a set of experiments used to validate the prioritised cell-rate simulator against published simulation results in [Krön91]; it also contains a comparison of the cell-rate simulated results with an approximate burst level analysis that provides an upper bound for cell loss. Then, a verification of some general properties for priority mechanisms is performed with the prioritised simulator; in this process, some interesting observations are made that contribute to a better understanding of the PBS scheme at burst level.

Finally, the Chapter concludes with an assessment of the impact of including the PBS mechanism in the cell-rate simulator.

5.1. The Simulation Tool

The simulation tool used to implement the PBS mechanism, LINKSIM, was developed at Queen Mary and Westfield College (QMW), initially by Pitts (see [Pitt93, pp.102-121]). It is based on the cell-rate simulation method (referred to in Section 4.2.2) and it allows multiple traffic sources to be multiplexed into only one queue. This means that the incoming information is treated as groups of cells and the events themselves are the changes between successive bursts. The implementation of the PBS mechanism is new work by the author.

5.1.1. The Model

The *cell-rate simulation method* is a burst level method where the basic unit of traffic is a burst of cells; this means that during a certain time period, the time between any two consecutive cell arrivals is treated as being constant. Moreover, cell-rate simulation does not account for cell synchronisation; in order words, the inherently discrete nature of ATM cells is approximated and so, any burst of cells can contain a non-integer number of cells (in the context of the simulator). The *events* (see Section 4.2.1.1) represent the time instants between bursts of cells with different cell rates (see Fig.5.1). Thus, with this method, it is possible to simulate the same number of cell arrivals as a cell level simulator but with less amount of simulation time being required; as each event comprises several cell arrivals, the processing of traffic is no longer performed on an individual basis giving the increase in speed relative to cell level simulation.

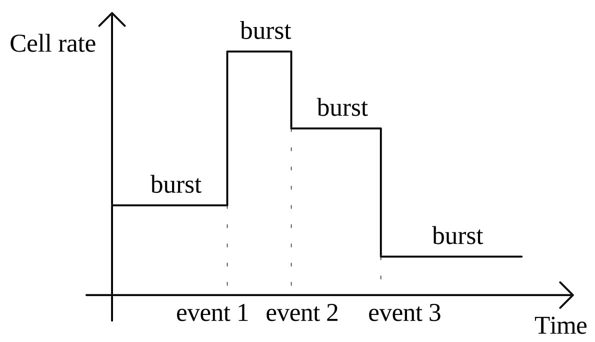


Fig.5.1 - Bursts of cells and events.

The traffic sources modelled in LINKSIM are described by a group of states, each state having a fixed cell-rate with a duration given by a probability distribution (in the present case, either the Exponential or the Geometric distribution). In particular, it is possible to simulate GMDP sources, according to the model described in Section 4.1.3, for which the duration of each state is geometrically distributed.

The network topology used in LINKSIM (see Fig.5.2) is composed of a Network Termination (NT), which can act as a source or a destination, depending on the place in the model, and a unidirectional Network Element (NE), which receives the multiplexed traffic from the incoming NT. The NE then sends the cells that it has received to a destination NT; the NE represents in fact a delay propagation time and it is also the place where traffic calculations are carried out. The network element is characterised by its cell-rate capacity, delay and queuing capacity.

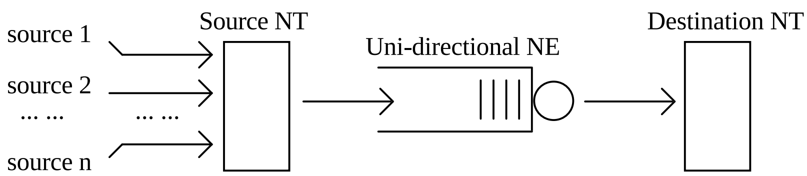


Fig.5.2 - The network topology of LINKSIM.

The execution of a simulation requires an input data file, which contains :

* *details about the call types*
  - number of call types,
  - number of states for each call type,
  - cell rate for each call type,
  - distribution of the duration for each state in each call type,
  - mean sojourn time of each state in each call type,
  - transition probabilities of changing from a state to another for each call type;
* *information about the network element*
  - propagation delay,
  - queue capacity,
  - bandwidth capacity;

* *simulation run details*
  - seeds for the random number generator,
  - number of sources (i.e., number of VCs) of each call type,
  - details for the batch means method (see Section 4.2.1.3 and [Law91, pp.553-564]) used in the measurement of cell loss probability due to buffer overflow.

The details relative to the batch means procedure, which is the used method to take measurements, consist of data from the t-distribution (sometimes called *Student's t distribution*) for the chosen confidence interval width and of the number of measurements. To obtain the t-distribution data, a value is chosen from the t-distribution (which has *m-1* degrees of freedom, for a sample of observations of size *m*) corresponding to the percentage confidence interval for 39 degrees of freedom. This value is explained because LINKSIM maintains 400 batches of measurements that, as the simulation progresses, represent averages of increasing numbers of individual measurements; the number of measurements for a given simulation run (at each intermediate stage of the simulation) comes to be $400 \cdot 2^{n}$ for increasing values of *n*, where *n* represents a positive integer (see [Pitt93]). When calculating the correlation measure and the confidence interval, these 400 batches are combined as 40 groups of 10 batches each, hence the figure of 40-1= 39 degrees of freedom (see also Section 4.2.1.3), where m=40.

LINKSIM's simulation runs produce information about the *total cell loss probability* of the system, as well as the value of the *correlation indicator* and the *width of the confidence interval* relative to the cell loss probability obtained. Other results include the *total number of cells sent* (i.e., the number of cells generated by the simulator), the *total number of cells lost* and the *total processing time* used by the simulator.

5.1.2. How does it Work?

The understanding of the queue's operation mode in LINKSIM is important, as it represents the basis of the simulation tool. Two parameters are used to describe the queue: its *buffer capacity* (i.e., the maximum length of the queue in cells) and its *service rate* (translated as the amount of cells that may be serviced in each unit of time). With these parameters, the states of the queue are calculated by taking into account,

* the *size of the queue* in a given time instant;
* the *input rates of all the VCs* that are accessing the queue;
* the *queue's parameters*.

The way in which traffic is processed in a queue is represented in Fig.5.3; it shows that every cell entering the queue will always either be served, queued or lost. The rates that describe this are the *input*, *queuing*, *loss* and *output rates*; they have to satisfy the following relation at all times for each individual VC and all VCs in the system (mathematically represented by equation (5.1)):

Input rate = Queuing rate + Loss rate + Output rate.

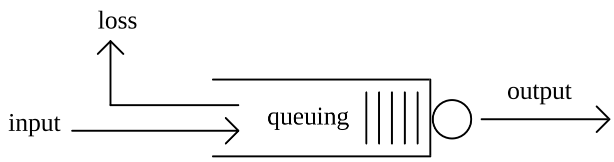


Fig.5.3 - The model for the queue.

The previous Section defined an event, in the context of the cell-rate simulator LINKSIM, as representing input rate changes. When a VC suffers a change in its input rate, the impact is not seen immediately at the output of the queue, unless there are no cells waiting to be served in the queue; if there are cells being queued, then only the queuing or loss rates are updated at the input of the queue. After all the cells queued have been served, the previous change in the input rate of a VC is finally visible at the output of the queue; at this point, only the queuing and output (or acceptance) rates are updated.

In what follows, mathematical equations will be introduced in order to clarify what happens during the operation of the queue. They will be referred to in the text using numbered brackets. Bearing in mind the queue's model, it is easy to see that it will be important to analyse what happens to arriving bursts of cells when the queue is either *empty*, *full*, is *being filled* or is *being emptied*. In the first case, if the sum of input rates from all VCs is less than the service rate of the queue, the VCs output rates will be equal to the corresponding input rates (5.2); this means that all incoming traffic will be accepted by the queue. Otherwise, the queue will start being filled and the bandwidth of the queue will be shared between the VCs, proportionally to their input rates (5.3). The queuing rate represents the excess of input over output rates (5.4).

During the period in which the queue is not empty, the total output rate equals the server capacity. When the queue is full, no more cells will be queued (the queuing rate is therefore null) and the loss rate will be the excess of the total input rate over the total output rate (5.5). It should be noted however that this may not be necessarily true for individual VCs (see [Pitt90]). The size of the queue will decrease when the total input rate comes to be less than the service rate of the queue. In this case, the total output rate will equal the server capacity until the queue becomes empty.

Analytically, it is possible to describe the various states of the queue and values of the input, output, queuing and loss rates, using the notation given next. Let,

C_{max} = cell buffer capacity

O_{max} = cell service rate

$I(i,e)$ = input cell rate for VC *i*

$I_{tot}(e)$ = total input cell rate

$O(i,e)$ = output cell rate for VC *i*

$Q(i,e)$ = cell queuing rate for VC *i*

$C_{tot}(e)$ = total number of cells queued

$L(i,e)$ = cell loss rate for VC *i*

$L_{tot}(e)$ = total cell loss rate,

where $i \in \{1,\ldots,n\}$ indicates the *ith* VC and $e \in \{0,1,2,\ldots\}$ indicates the *eth* event in the queue. Then, the most important relations (explained in the previous paragraphs) for each of the queue's states are:

$$I(i,e) = O(i,e) + Q(i,e) + L(i,e) \tag{5.1}$$

$$O(i,e) = I(i,e), \text{ when } I_{tot}(e) < O_{max} \tag{5.2}$$

$$O(i,e) = I(i,e) \cdot \left(\frac{O_{max}}{I_{tot}(e)} \right), \text{ when } C_{tot}(e) = 0 \;\; \textit{and} \;\; I_{tot}(e) > O_{max} \tag{5.3}$$

$$Q(i,e) = I(i,e) \cdot \left(1 - \frac{O_{max}}{I_{tot}(e)} \right), \text{ when } C_{tot}(e) = 0 \;\; \textit{and} \;\; I_{tot}(e) > O_{max} \tag{5.4}$$

$$L_{tot}(e) = I_{tot}(e) - O_{max} \tag{5.5}$$

After processing the traffic in the queue, the cell loss ratio is determined by using the next expression; it considers the ratio of the total number of cells lost to the total number of cells generated by the simulator and input to the queue:

$$\text{cell loss ratio} = \frac{\sum\limits_{e \in \{0,1,2,\ldots\}} \left[L_{tot}(e-1) \cdot (t(e) - t(e-1))\right]}{\sum\limits_{e \in \{0,1,2,\ldots\}} \left[I_{tot}(e-1) \cdot (t(e) - t(e-1))\right]}, \tag{5.6}$$

where *t(e)* represents the time instant at event *e*. A similar formula can be devised to evaluate the cell loss probability for each VC in the system (see [Pitt93, pp.97]).

5.2. Implementing the PBS Mechanism in the Simulator

LINKSIM, the simulation tool used to implement the space priority scheme PBS, has been written in the Pascal programming language. The code is divided into modules and it is possible to use the simulator both in the UNIX (e.g., SUN SPARC machines) and the DOS environments. The modifications by the author to the LINKSIM code, necessary for it to take priorities into account, were also executed in Pascal. The source code with this prioritised version is now approximately 5500 lines (representing an increase of about 28% on the original version of LINKSIM without priorities) and the executable code size is of the order of 260 Kbytes (which represents an increase of about 18%). The idea of implementing a space priority mechanism in a cell-rate simulator was first suggested by the author and described in [Fons94a] and in [Fons94b] the implementation of PBS in the cell-rate simulator was then validated against published simulation results from [Krön91].

Sections 5.2.1 and 5.2.2 will explain the main modifications necessary for the implementation of PBS, as well as the approximations considered.

5.2.1. Approximation for PBS

It has been noted (see Section 3.8.2.2) that the *Partial Buffer Sharing mechanism* divides a queue into two parts, the part above the threshold being accessed only by high priority cells. Sources can have both low and high priority cells in the new version of LINKSIM with priorities (e.g., like video sources); this has been specified by indicating the cell rate corresponding to each priority and, in each state, the sum of the cell rates from low and high priority cells must therefore equal the fixed cell rate of the state.

The way in which the cell rate simulation method works suggested the consideration of the existing queue size to be the threshold, in the process of implementing PBS in the cell-rate simulator. The reason for this approximation is that burst scale queuing above the threshold only occurs if the total input rate of the high priority cells exceeds the service rate of the queue. This event is very unlikely (especially since the so called *high priority traffic* usually represents a small proportion of the total traffic) and therefore it is reasonable to assume a zero queue size above the threshold. When this happens, the bursty nature of the traffic (more evident when the traffic sources are modelled as GMDP sources, that have the On/Off model as a special case) will almost certainly become buffer overflow and hence cell loss, regardless of the actual capacity above the threshold. The benefit that this approximation brings is a much simpler implementation. Besides, the actual space in practice above the threshold deals with the *cell scale* queuing effects and cell-rate simulation only models the burst-scale queuing behaviour.

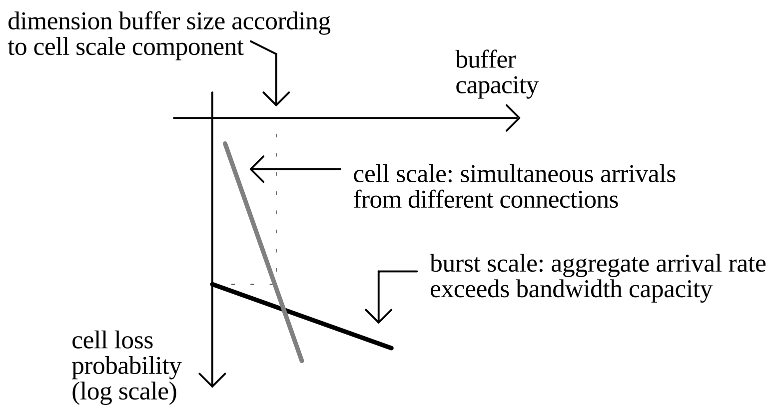


Fig.5.4 - Burst and cell scale components of queuing behaviour.

Fig.5.4 shows the burst and cell scale components of queuing behaviour; for small buffer sizes, losses occur due to simultaneous cell arrivals from the bursts, while for long buffer sizes, the losses are mainly due to burst scale fluctuations in the generation and termination of bursts. In the latter case, buffering tends to smooth the effect of cell scale fluctuations that normally happen with small buffers.

Normally, when the queue reaches the threshold (which in this case will correspond to the state full of the queue), the proportion of cells lost will have to be based on the amount of low priority traffic - i.e., the low priority traffic is the first to be lost. If the total high priority input rate exceeds the service rate of the queue, then the excess high priority traffic should be lost in addition to all the low priority traffic first (rather than model the queuing above the threshold). Modelling the queuing above the threshold would imply the analysis of more states in the queue (e.g., "below threshold", "equal to threshold" and "above threshold"), which would be significantly more complicated (see the next Section).

5.2.2. Necessary Modifications

How was the PBS mechanism added to LINKSIM? This Section describes new work by the author. In this tool, the analysis of the queue is executed according to its current state. The state of a queue is determined by the buffer capacity and the service rate, as referred to before (see Section 5.1.1). Thus, there are nine possible states in a queue, as depicted by the Table 5.1.

	Size of the queue		
	empty	mid	full
input > service rate	1	4	7
input = service rate	2	5	8
input < service rate	3	6	9

Table 5.1 - Possible states of the queue.

However, knowing the current state of the queue is not enough; in fact, because LINKSIM models the propagation of cell-rate changes through the queue, it is necessary to take into account the state of the queue in the previous event, which would lead to the need of analysing $9 \cdot 9 = 81$ possible transitions in the queue. This task is simplified by the fact that some transitions are illegal (see [Pitt93, pp.110-118]), thus taking us to the final 39 possible transitions in a queue, shown by Table 5.2; the grey areas represent illegal transitions. If modelling of the queuing above the threshold was to be implemented, then it would be necessary to analyse 3·5 = 15 queue states, which would take us to 15·15 = 225 possible transitions.

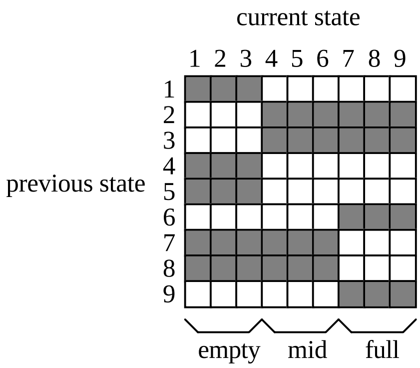


Table 5.2 - State transitions in the queue.

In the last section, it was seen that the modifications necessary to add the PBS scheme to LINKSIM take place only when the queue is being analysed in its so-called *full state*; this means that when the queue is in any of the other states, then equations (5.1) to (5.4) apply, both to low and high priority cells, since below the threshold (i.e., when the queue is not full), cells are served as if there were no distinctions between them in terms of priority levels. In the *full state*, low priority cells are accepted into the queue, provided the bandwidth available in the queue is enough for them, after accepting all incoming high priority cells. Therefore, it is necessary to investigate whether the input cell rate of high priority traffic combines to lesser or greater than the service rate of the queue.

In the first case, the output rate will equal the input rate (5.7) and what is accepted from the low priority traffic is the proportion of bandwidth still available in the queue (5.8). The loss rate for high priority traffic will therefore be null and for low priority traffic, the corresponding loss rate will be the excess of the input rate of this traffic over the output rate (i.e., what is accepted by the queue) (5.9), if the input rate of low priority traffic is greater than the bandwidth available.

On the other hand, if the input cell rate of high priority traffic combines to be more than the service rate of the queue, then the queue will accept only a proportion of the input cell rate of high priority traffic (5.10) and any incoming low priority traffic will be lost (5.11). The equations (5.7) to (5.11) below explain what happens both to low and high priority traffic in this circumstance. For these equations, a similar mathematical notation to that of Section 5.1.2 has been used, making the distinction between low priority and high priority traffic simply by indexing the different cell rates. Then,

for $I_{tothp} < O_{max}$

$$O_{hp}(i,j) = I_{hp}(i,j) \tag{5.7}$$

$$O_{lp}(i,j) = I_{lp}(i,j) \cdot \left(\frac{O_{max} - I_{tothp}(j)}{I_{totlp}(j)} \right) \tag{5.8}$$

$$L_{lp}(i,j) = I_{lp}(i,j) - O_{lp}(i,j) \tag{5.9}$$

for $I_{tothp} > O_{max}$

$$O_{hp}(i,j) = I_{hp}(i,j) \cdot \left(\frac{O_{max}}{I_{tothp}(j)} \right) \tag{5.10}$$

$$L_{lp}(i,j) = I_{lp}(i,j) \tag{5.11}$$

5.3. The Validation Process

The last sections have described how it is possible to implement a good approximation to the *Partial Buffer Sharing mechanism* by making use of the cell-rate simulation method; the reason for assuming a zero queue size above the threshold when modelling queuing was also explained. However, implementing the PBS in the cell-rate simulator is not enough; it is necessary to validate it by comparing the results obtained with analytical studies and/or other simulation techniques.

The process used for the validation of the implementation was carried out by the author in two main phases:

a) a *first phase* in which the results from the developed prioritised cell-rate simulator are compared against previously published simulation results in [Krön91] for On/Off sources;

b) a *second phase*, described in detail in Chapter 6, which meant the author having to extend an analytical fluid flow model in order to cope with priorities and again, comparing results from both analytical and simulation approaches.

In the first validation phase, another step was taken that consisted of comparing an approximate burst level analysis, also introduced in [Krön91], against the prioritised simulation results from the cell-rate simulator for priorities. This is described in Section 5.3.2.

5.3.1. Using Published Results to Validate LINKSIM with Priorities

A first validation was obtained by comparing the results from LINKSIM with those obtained by Kröner (see [Krön91]). There is a slight complication with the comparison in that Kröner's simulation was not intended to calculate a cell-loss rate, but to compare the number of sources of the two types that could be carried for a *given* cell-loss ratio; Kröner does not state the actual values of cell loss ratio. The values on Table 5.4 are upper bounds for his experiments.

Two types of sources were used for the validation in the first phase: *video* and *data*, both modelled as On/Off sources with *On* and *Off* periods exponentially distributed. Each type of source is assumed to have cells of only one type of priority. Since video communications are usually very sensitive to losses, video traffic sources have been assumed to represent the class of high priority traffic and data traffic sources are classified as low priority traffic, because losses can be recovered in higher layers. Table 5.3 gives the parameter values assumed for each type of source.

	Type of Source	
Traffic Parameters	video	data
On state mean duration	14.815 ms	100 ms
Off state mean duration	25.185 ms	400 ms
peak bit rate	10 Mbit/s	10 Mbit/s
mean bit rate	3.7 Mbit/s	2 Mbit/s
burstiness	2.7	5

Table 5.3 - Traffic parameters for each source type (from [Krön91]).

The transmission rate (taken from [Krön91]) used for the validation was 150 Mbit/s and simulations were carried out in order to confirm that a cell loss probability of no more than 10^{-2} and 10^{-4} would be achieved for low and high priority traffic, respectively, when combining video and data traffic sources according to the experiments from [Krön91]; the characteristics of the system and expected cell loss probability results are summarised in Table 5.4.

Input System Characteristics	
link transmission rate	150 Mbit/s
buffer length	64 cells
threshold size	48 cells
expected maximum low priority cell loss probability	10^{-2}
expected maximum high priority cell loss probability	10^{-4}
expected maximum global cell loss probability	10^{-4}

Table 5.4 - Kröner's values for the simulated queuing system and QoS requirements.

Results were also obtained for specific combinations of video and data sources when no priority mechanism is present; in this case, the cell loss probability that should be obtained must be better than 10^{-4}. This last set of results was necessary because it would not be reasonable to consider the output of the simulator for traffic with priorities if the original simulator's output did not match Kröner's results for the case without priorities. The number of sources of each type that are necessary to produce the required values of cell loss probability is shown in Fig.5.5 (taken from [Krön91]). The figure (Fig.5.5) shows that the introduction of a priority mechanism such as PBS allows a substantial increase in the number of admitted high priority traffic sources (i.e., video sources in this case) by the network, for the same number of low priority traffic sources (i.e., data sources in this case).

The cell-rate simulation results obtained are presented in Fig.5.6 and Fig.5.7 and can then be compared with the results obtained by [Krön91]; both graphs show the confidence intervals for the cell loss probability values obtained in each experiment. The first figure (Fig.5.6) gives the cell loss probability obtained for specific combinations of video and data sources (not considering priorities) with a buffer size of 64 cells; the number of sources used in each experiment is indicated in Table 5.5, which also gives the combination of low and high priority traffic sources used to compare Kröner's simulation results with the prioritised cell-rate simulator.

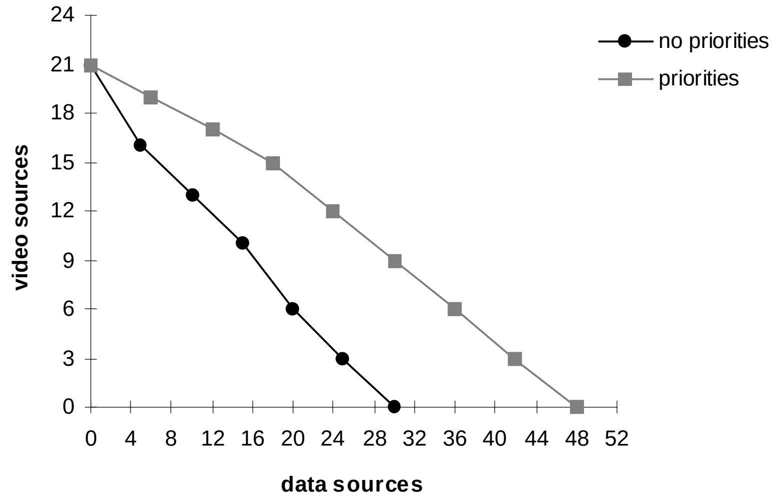


Fig.5.5 - Combinations of traffic sources (from [Krön91]).

In Fig.5.7, results are shown for the low and high priority cell loss probability obtained in each of the experiments carried out, considering a queue size of 48 cells (which corresponds also to the threshold value indicated in Table 5.4 and used in [Krön91] only for the situation of traffic with priorities).

Traffic without priorities	exp#	1	2	3	4	5	6	7	---	---
	video	0	3	6	10	13	16	21	---	---
	data	30	25	20	15	10	5	0	---	---
Traffic with priorities	exp#	1	2	3	4	5	6	7	8	9
	video (hp)	0	3	6	9	12	15	17	19	21
	data (lp)	48	42	36	30	24	18	12	6	0

Table 5.5 - Number of video and data sources used in each experiment (from [Krön91]).

With respect to Fig.5.6, it can be seen that for each experiment, the simulation result obtained for cell loss probability is always within one order of magnitude of the expected corresponding value obtained by Kröner (see [Krön91]), even when the confidence interval is taken into account for each cell loss probability simulation result. The cell-rate simulation results are higher than the expected cell loss probability value, 10^{-4}.

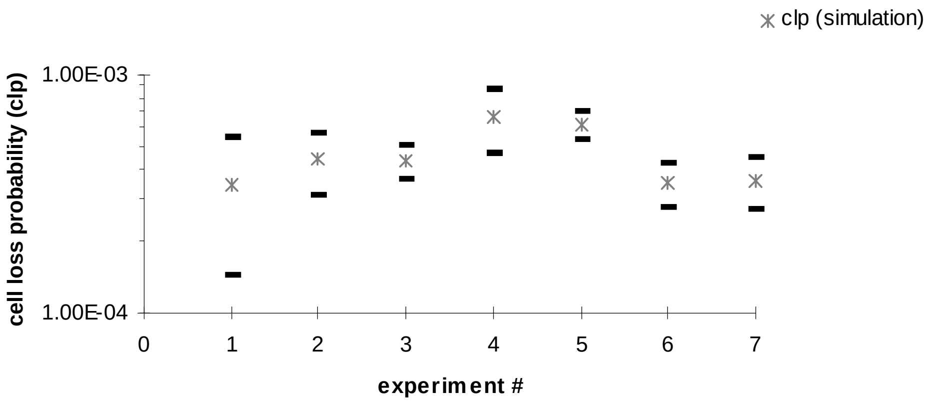


Fig.5.6 - Simulation results using LINKSIM for traffic without priorities.

The results in Fig.5.7 show that, for all experiments, there is a good agreement between the low priority cell loss results obtained with the cell-rate simulator and the expected cell loss values of 10^{-2} obtained in [Krön91]. The high priority cell loss results show a lower loss rate.

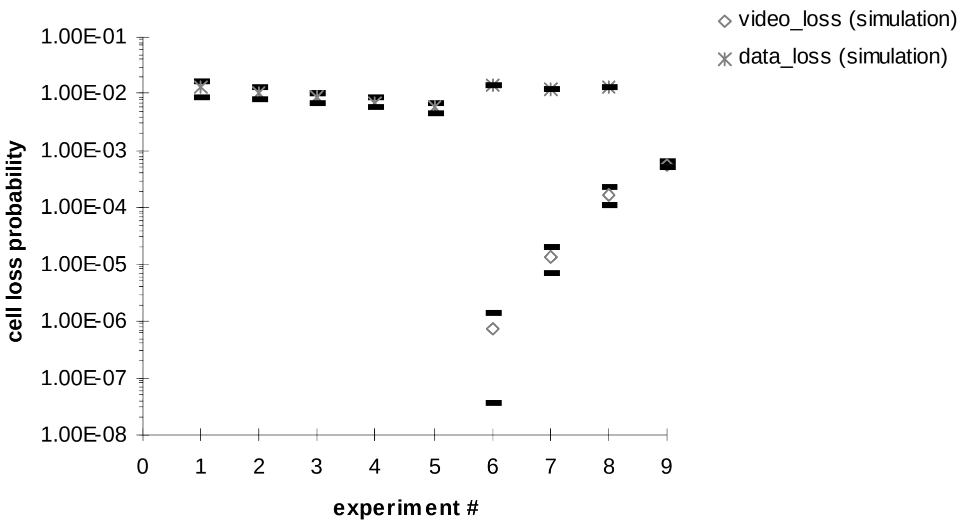


Fig.5.7 - Simulation results using LINKSIM with the PBS mechanism.

To explain this, it is necessary to take into account the following. As described in Section 5.2.1, the implementation of the Partial Buffer Sharing loss priority mechanism considered that the queue size above the threshold should be zero, without significantly compromising the accuracy of the results. Hence, in this case, the cell-rate simulation experiments took the threshold value of 48 cells used in [Krön91] as the queue size value. This means that high priority traffic will only be lost when *both* the queue is full and the sum of all high priority input rates combines to be greater than the service capacity of the queue. In the case of experiments 1 to 5, the total high priority input rate is never greater than the queue's service capacity, so no cell loss is seen by the prioritised cell-rate simulator. The remaining experiments consider a total high priority input rate that is increasingly greater than the service capacity of the queue; hence, the increasing high priority cell loss simulation results obtained. In particular, the maximum value of high priority cell loss is obtained with experiment number 9, while no low priority cell loss is registered. This is because that traffic experiment considers only high priority traffic sources.

It is also worth noting the apparent trend in some of the low priority cell loss results (see Fig.5.7). A look at Table 5.5 shows that the number of low priority traffic sources always decreases by 6 between any two consecutive experiments. On the other hand, the number of high priority traffic sources increases by 3 between most (but not all) experiments. Therefore, overall and only up to experiment 6, the total number of sources is decreasing by 3 but the total mean bit rate is decreasing by 0.9 Mbit/s, because 3 high priority sources do not correspond to 6 low priority sources (see Table 5.3 for the sources' mean bit rate values); this is why the low priority cell loss does not remain constant.

As a summary of this part of the first main phase in the validation of the prioritised cell-rate simulator, it can be said that the simulation results obtained agree generally with Kröner's simulation results, especially for the low priority traffic. On the high priority traffic, it is possible to conclude that in the buffer above the threshold, the major component of queuing is the cell-scale behaviour (not modelled by the present cell-rate simulator), particularly when the proportion of high priority traffic is low.

5.3.2. An Upper Bound for Cell Loss at Burst Level

Up to the moment, no exact queuing analysis for systems with the Partial Buffer Sharing mechanism seems to be mathematically tractable (see [Krön91]). Therefore, simulation (as used in the last Section) and approximate analysis are the alternative methods applied to evaluate the performance of such systems. In this Section, an approximate analysis, which has been introduced in [Krön91], is used by Kröner to derive an upper bound for the cell loss caused by congestion at burst level.

The occurrence of overload at burst level implies a loss of cells, since it cannot be buffered within the network. The analysis considers bursty traffic with priorities, but it ignores the buffering capabilities of ATM networks at burst level. Comparing the cell loss results simulated with the prioritised cell-rate simulator against the analytical results should give an indication of the prioritised cell-rate simulator's accuracy, thus providing additional information to the validation process described in the last Section. The description of the analysis given here merely indicates the formulas used to compare the cell-rate simulation results with the analysis.

In order to understand the approximate analysis that will be used here, a few definitions are introduced that characterise the bursty On/Off modelled traffic sources to which the analysis applies. Taking into account that a *class 1* traffic source represents high priority traffic and that *class 2* sources represent low priority traffic, consider,

N_i = total number of class *i* (i=1,2) sources in the system;

mbl_i = mean number of cells in a burst for a class *i* traffic source (i=1,2);

iat_i = interarrival time for a class *i* traffic source (i=1,2);

msd_i = mean silence duration for a class *i* traffic source (i=1,2);

pa_i = probability of a class *i* (i=1,2) connection being in a burst (or active) state;

$p(x_i)$ = probability of x_i traffic sources of class *i* (out of the total number of sources, N_i, for class *i*) being active;

h = queue's service time.

With these definitions and using the Binomial distribution to calculate the value of $p(x_i)$, the aggregate cell loss probability is then determined by the expression,

$$B = \frac{1}{\frac{N_1 \cdot pa_1}{iat_1} + \frac{N_2 \cdot pa_2}{iat_2}} \cdot \sum_{\substack{x_1=0,\dots,N_1 \\ x_2=0,\dots,N_2}} p(x_1) \cdot p(x_2) \cdot \left(\frac{x_1}{iat_1} + \frac{x_2}{iat_2} - \frac{1}{h} \right) \tag{5.12}$$

that is valid only when the total arrival rate to the system combines to more than the queue's service rate; in other words, the last term in the summation of equation (5.12) only contributes to the expression when it is positive. The formula takes also into account the mean aggregate cell arrival rate given by the inverse of the first factor in the equation.

How is the cell loss calculated for each traffic class? First, [Krön91] considers the case in which no priorities are taken into account; for this case, the cell loss suffered by each type of traffic on a certain system state (x_1, x_2) is influenced by the fraction of traffic offered by each traffic class. Taking this into consideration, the cell loss probability for traffic class *i* (i=1,2) is then given by,

$$B_i = \frac{1}{\frac{N_i \cdot pa_i}{iat_i}} \cdot \sum_{\substack{x_1=0,\dots,N_1 \\ x_2=0,\dots,N_2}} p(x_1) \cdot p(x_2) \cdot \left(\frac{x_1}{iat_1} + \frac{x_2}{iat_2} - \frac{1}{h} \right) \cdot \frac{\frac{x_i}{iat_i}}{\frac{x_1}{iat_1} + \frac{x_2}{iat_2}}. \qquad (5.13)$$

In this last formula, the aggregate loss rate must again be either positive or null and the last term in the summation represents the fraction of traffic offered by traffic class *i*, as referred to in the last paragraph. When priorities are considered, the loss probability of traffic *class 1* should be smaller than in the previous case. Moreover, there should be cell loss for high priority traffic only when this type of traffic enters an overload situation. Thus, the high priority cell loss is

$$B_1 = \frac{1}{\frac{N_1 \cdot pa_1}{iat_1}} \cdot \sum_{x_1=0,\dots,N_1} p(x_1) \cdot \left(\frac{x_1}{iat_1} - \frac{1}{h} \right), \qquad (5.14)$$

where again, the last term in the summation must be positive. On the other hand, the low priority cell loss is calculated by taking into account the conservation law for the aggregate loss probability *B* - as given by equation (5.12) -, that says: "... the product of the mean total arrival rate by the total cell loss of the system must equal the sum of the corresponding products, for low and high priority traffic cell losses ...". In other words, the following relation is verified,

$$\frac{N_1 \cdot pa_1}{iat_1} \cdot B_1 + \frac{N_2 \cdot pa_2}{iat_2} \cdot B_2 = \left(\frac{N_1 \cdot pa_1}{iat_1} + \frac{N_2 \cdot pa_2}{iat_2} \right) \cdot B. \qquad (5.15)$$

This equation is asymptotically exact for increasing burst and silence durations of the system's sources, provided the buffers are large enough to cope with cell level congestion.

Making use of equations (5.12) to (5.15), it is now possible to compare the results obtained with the prioritised simulator against those given by the approximate analysis. Table 5.6 translates the sources' traffic parameters previously given in Table 5.3 into the mean burst length, interarrival time and mean silence duration of each traffic class by taking into account the service rate of the queue.

	Type of Source	
Traffic Parameters	video (class 1)	data (class 2)
mean burst length (mbl)	385 cells	2604 cells
mean silence duration (msd)	25.185 ms	400 ms
interarrival time (iat)	38.4 μs	38.4 μs

Table 5.6 - Video and data sources' traffic parameters (from [Krön91]).

With the parameters defined in Table 5.6 and going through the experiment configurations given in Table 5.4 and Table 5.5 (in this table, only for the case of traffic with priorities), cell loss results were obtained for the cases of traffic without and with priorities (see Fig.5.8 and Fig.5.9, respectively). The graph in Fig.5.8 plots the cell loss obtained with the cell-rate simulator for the video and data sources without priorities, as well as the corresponding cell loss upper bounds obtained with the approximate analysis; the combination of sources used in each experiment is the same as for the case with priorities (see Fig.5.9), so that it is easier to see what happens to the cell loss of both video and data streams in each situation: priorities or no priorities (Kröner used different combinations for the two cases). In both situations, a buffer size of 48 cells was used and for each experiment, the cell loss results for simulation and analysis are shown side by side.

It can be seen (see Fig.5.8) that the cell-rate simulator produces, in all experiments, better cell loss values (for both traffic types) than the upper-bound burst level analysis. This is because the analysis considers a buffer of size zero to calculate the cell loss. If the cell-rate simulator were to consider a smaller buffer size, the cell loss simulation results obtained would be even closer to the analytical results.

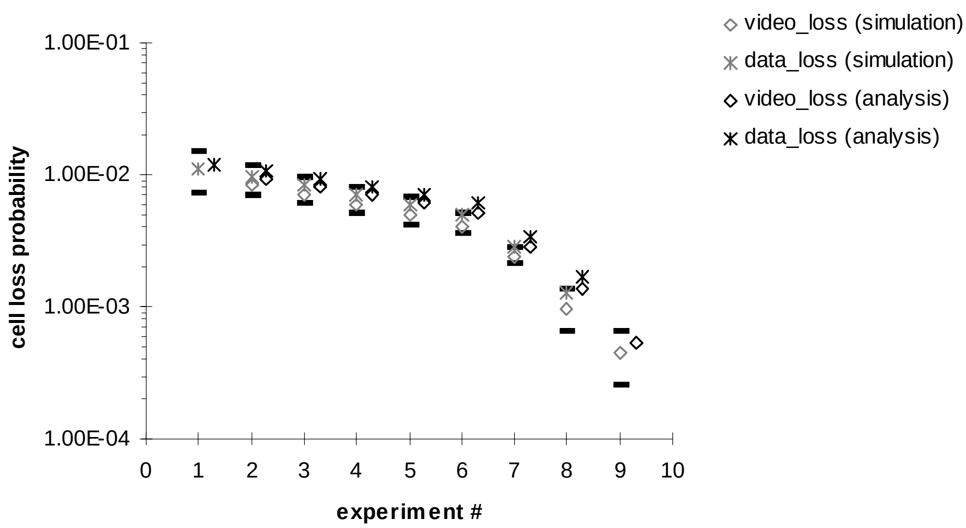


Fig.5.8 - Cell loss upper bound for traffic without priorities.

The graph in Fig.5.9 contains similar information to that of Fig.5.8 by using the same combination of video (now taken to be high priority traffic) and data (now taken to be low priority traffic) sources. However, in this case the cell loss upper bounds have been calculated by taking into account traffic with priorities.

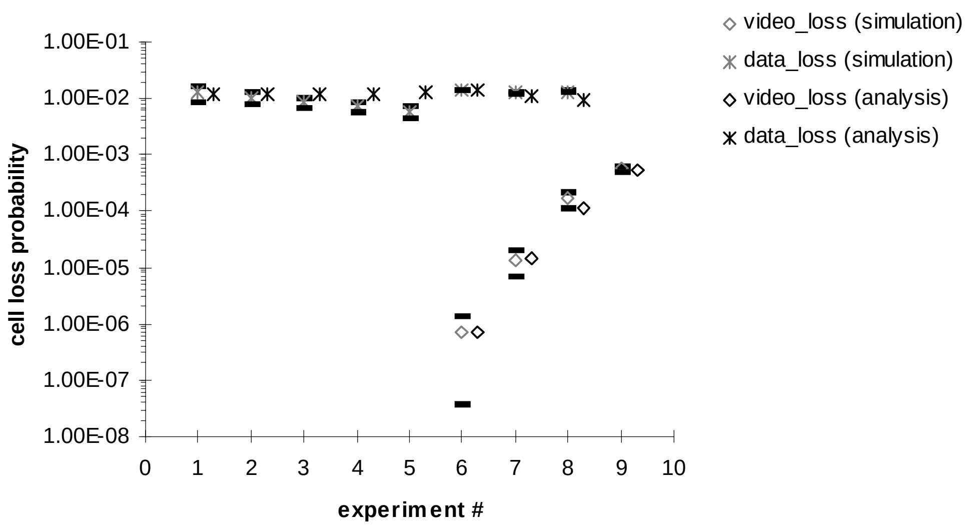


Fig.5.9 - Cell loss upper bound for traffic with priorities.

The results from both the prioritised simulator and the approximate analysis for traffic with priorities are very similar, due to the small buffer size used with the prioritised simulator. However, it can be seen once more that in most experiments, the analytical cell loss is slightly higher than the simulated cell loss as would be expected.

In conclusion, the simulation gives good results when compared with the upper-bound cell-loss results obtained with the approximate analysis described in the beginning of this Section. This increases the confidence in the implementation of the priority mechanism in the cell-rate simulator.

5.4. Typical Behaviour of Priority Mechanisms: Verification with the Prioritised LINKSIM

Priority mechanisms in general and loss priority mechanisms in particular (as is the case of the Partial Buffer Sharing scheme studied in this thesis) produce an impact in the behaviour of the system where they are implemented that can be translated into a set of properties for a typical priority mechanism (see for example [Roth90], [Krön91] and [Meye93]). These are:

1. the use of a priority mechanism can improve the admissible load (or equivalently, the number of traffic sources) in a network;
2. for a system with a priority mechanism, a decrease in the threshold size of the queue causes an increase in the cell loss probability of the low priority traffic;
3. the performance of priority mechanisms is better when the high priority traffic represents only a small proportion of the total traffic than in the case where the high priority traffic represents the bulk of the total traffic.

Note however, that the second property is valid only when the burst and cell scale queuing components are taken into account (see Fig.5.4); in other words, if the size of the queue is not of the same order of magnitude as the traffic sources' mean burst length in the system, then no significant change occurs in the cell loss. Overall, these characteristics give to a network that uses a priority mechanism the robustness necessary to cope with bursty traffic, which is not possible to achieve by just overdimensioning network queues.

In this Section, the author shows that the implemented approximate version of Partial Buffer Sharing verifies the characteristics referred to above. To this end, several experiments were carried out with different types of traffic sources and network features.

	Type of Source	
Traffic Parameters	A	B
On state mean duration	20 ms	40 ms
Off state mean duration	80 ms	60 ms
peak bit rate	10 Mbit/s	34 Mbit/s
mean bit rate	2 Mbit/s	13.6 Mbit/s
burstiness	5	2.5

Table 5.7 - Description of traffic sources (property 1).

In the verification of property 1, two types of sources were considered; their characteristics are given in Table 5.7. Two similar sets of experiments for different network loads were carried out in two steps: in the first step, a mixture of type *A* and type *B* sources was considered and simulated for the case without priorities. Then, three other experiments were carried out which maintained the number of type *B* sources (now representing high priority traffic) and successively increased the number of type *A* sources (now representing low priority traffic), therefore also increasing the network load. For these experiments, a threshold value corresponding to 75% and 80% of the buffer size was used in the first and second sets of experiments, respectively. Table 5.8 gives the configuration for each set of experiments.

Set 1	exp#1		exp#2		exp#3		exp#4	
threshold	sources	1A+5B	sources	3A+5B	sources	5A+5B	sources	7A+5B
30 cells	load	45%	load	47.6%	load	50.2%	load	52.7%
Set 2	exp#1		exp#2		exp#3		exp#4	
threshold	sources	9A+5B	sources	12A+5B	sources	15A+5B	sources	18A+5B
32 cells	load	55.3%	load	59.2%	load	63%	load	66.9%

Table 5.8 - Verification of property 1 for priority schemes: configuration of experiments.

As expected, the cell loss results obtained and shown in Fig.5.10 prove that it is possible to increase the load admitted to a network when priorities are taken into consideration. The graph (see Fig.5.10) shows that, for each initial load considered without priorities, the introduction of a priority mechanism and an increase of the number of sources in the system, produce only a slight increase in both the global cell loss and in the low priority cell loss while maintaining the performance of the high priority traffic. However, this degradation in the QoS does not exceed one order of magnitude for increases of up to about a 10% increase in the network load.

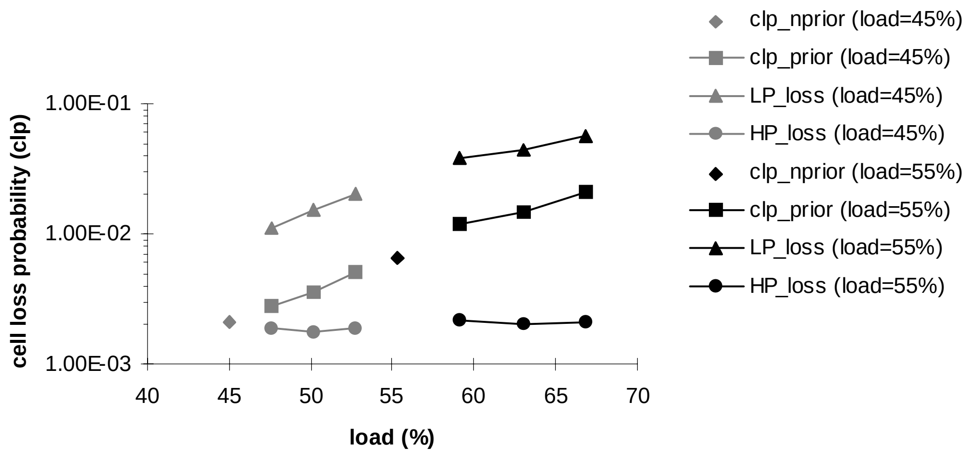


Fig.5.10 - Increase in the admissible network load when using priorities.

In the verification of the second property of a priority mechanism (as listed in the beginning of this Section), two types of On/Off modelled traffic sources, representing low and high priority traffic, were multiplexed into a buffer, under different loads. For each network load, the number of low and high priority sources was maintained, while the threshold value was scaled.

	Traffic Sources		***Network Parameters***	
	low priority	high priority	link bandwidth	155.52 Mbit/s
On state duration	50 ms	20 ms	buffer size	3150 cells
Off state duration	190 ms	80 ms	threshold	1080 cells to 3080 cells
peak bit rate	21 Mbit/s	34 Mbit/s	load	50% to 70%
mean bit rate	4.375 Mbit/s	6.8 Mbit/s	source mix	5HP, 10LP to 17LP
burstiness	4.8	5	---	---

Table 5.9 - Parameters for the network and traffic sources (property 2).

The table above (Table 5.9) gives the details of the traffic sources and network parameters used in the several experiments. Fig.5.11 gives the cell loss results obtained for two different network loads (including confidence intervals for all the experiments). It can be seen that for increasing threshold sizes, the cell loss suffered by both low and high priority traffic streams decreases and this is also true for increasing network loads. However, the decrease in the cell loss probability is more noticeable for a low load.

Finally, to verify the last characteristic of a typical priority mechanism, only one type of source was used; however, some of the sources are considered in each experiment as representing low priority traffic while the remaining refer to high priority traffic. The reason for doing this has to do with the fact that it is especially difficult to counter balance the effect of increasing the percentage of high priority traffic while maintaining the total traffic for the same network load, when using different traffic source types to represent low and high priority traffic.

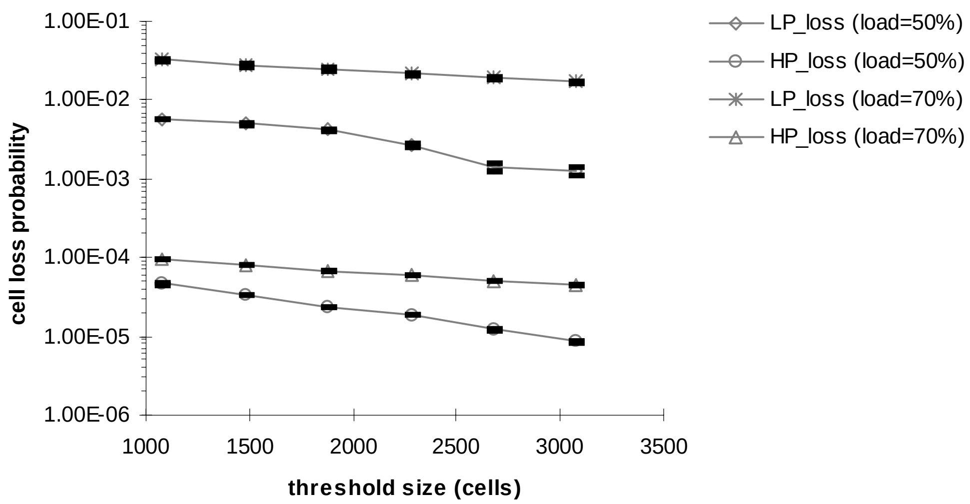


Fig.5.11 - Increase in cell loss probability for a decrease in the threshold.

In Table 5.10, both the traffic source parameter values and the characteristics of the system are given. The source type considered is again an On/Off source with exponentially distributed *On* and *Off* state durations. The simulated experiments considered an increasing network load (by altering the number of sources present in the system) and maintained the threshold size of the queue as a fixed value. Also, for each network load, the proportion of high priority traffic to the total traffic is increased from a minimum value of 30% to a maximum figure of 80%. The reason for considering several different loads, even though the number of sources is not maintained, is that this approach allows the observation of possible trends in the behaviour of the system.

Traffic Source		***System***	
On state mean duration	40 ms	link transmission rate	155.52 Mbit/s
Off state mean duration	70 ms	buffer length	50 cells
peak bit rate	20 Mbit/s	threshold size	35 cells
mean bit rate	7.3 Mbit/s	network load	50% to 80%
burstiness	2.75	high priority traffic %	30% to 80%
mean burst length	1887 cells	------	------

Table 5.10 - Source and system characteristics (property 3).

The results are shown in Fig.5.12; no confidence intervals are indicated as they tend to be very narrow, the ratio of the confidence interval's half amplitude over the mean cell loss being of the order 10^{-2}.

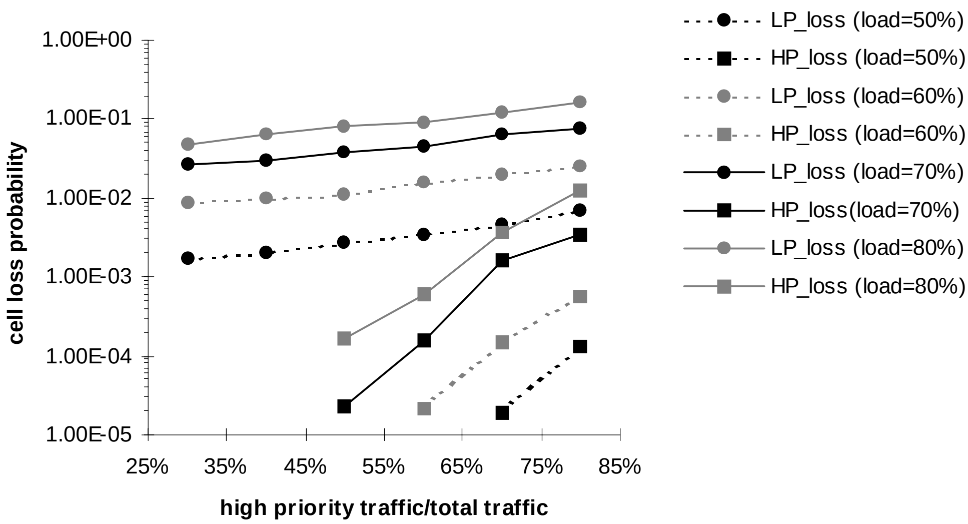


Fig.5.12 - Performance of PBS for an increasing proportion of high priority traffic.

The lines in the graph (Fig.5.12) represent the cell loss obtained for low and high priority traffic, under various network loads. It can be seen that, as the proportion of high priority traffic increases for a certain network load, the cell loss probability for high priority traffic tends to increase more rapidly than the low priority cell loss; a similar behaviour is also verified under different network loads. This confirms the property 3 described in the beginning of this Section. Also, as the network load increases, the degradation in the cell loss (of both low and high priority traffics) does not increase linearly, as it is shown more clearly in Fig.5.13. Here, the lines represent the low and high priority cell loss for given ratios of high priority traffic over the total traffic.

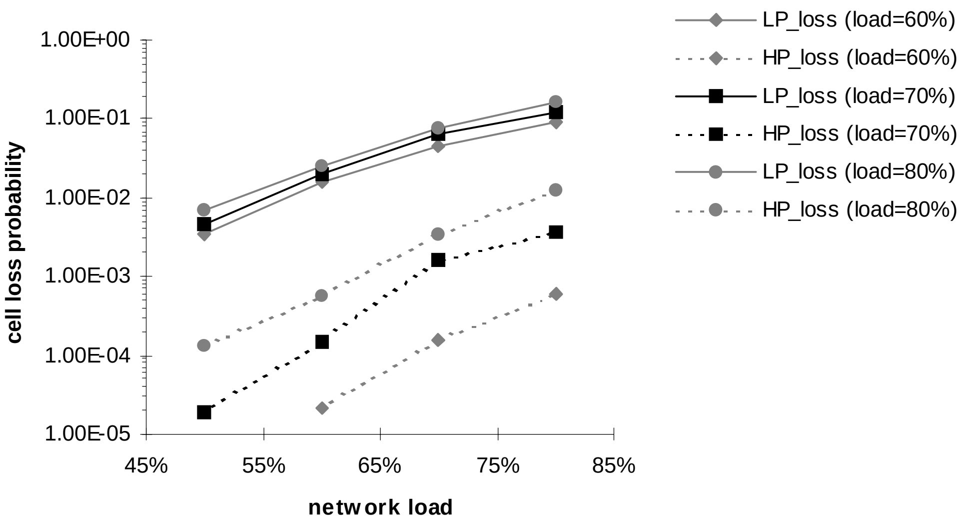


Fig.5.13 - Increasing the network load for a given high priority traffic percentage.

In conclusion, it can be said that the implemented approximate version of PBS verifies the main typical characteristics of a priority mechanism; this can also account for the validation of the implemented priority mechanism.

5.5. Processing Speed: Original LINKSIM *versus* LINKSIM with Priorities

In [Pitt93], it is demonstrated that "... in comparison with cell by cell simulation, cell-rate simulation shows speed increases of up to 4 orders of magnitude ...". The enhancement of the cell-rate simulator used in this thesis, LINKSIM, in order to study traffic with priorities, implied an increase in the complexity of the simulation program. It is therefore expected that a decrease in the processing speed of the prioritised LINKSIM will be seen when compared with the original version developed and described in [Pitt93, pp.102-121]. As a result, it becomes necessary to quantify the decrease in speed of the new extended version of the cell-rate simulator, in order to decide whether or not it is really worthwhile considering the prioritised LINKSIM as a useful tool in the study of priorities.

	Processing speed		***Speed***
experiment #	no priorities	priorities	***reduction***
1	6054	5111	16%
2	8108	6923	15%
3	6562	5781	12%
4	1645	1451	12%
5	2062	1786	13%
6	273105	204734	25%
7	144175	120239	17%
8	93914	85090	9%
9	98703	87026	12%
10	280539	264347	6%

Table 5.11 - Processing speed (in cell/s): no priorities *versus* priorities.

To achieve that objective, two main steps were considered. In the first one, several simulation experiments considering traffic with priorities were run and their processing times recorded. These simulation experiments were then repeated, using the original LINKSIM, for the same conditions (i.e., with both the same traffic source and system characteristics), but where all the traffic is taken to be of the same priority. With the obtained results, it was possible to compare the processing speed of both cell-rate simulator's versions. Table 5.11 shows the processing speed values obtained for the different experiments. The last 5 experiments in this Table used the traffic sources (a mix of two types of On/Off modelled traffic sources with *On* and *Off* exponentially distributed state durations) described in Section 5.3, while the other experiments made use of the experiment configuration (one On/Off modelled traffic source with *On* and *Off* geometrically distributed state durations) applied later in Section 6.4.

It can be seen from Table 5.11 and Fig.5.14 that the loss in processing speed with the new version of LINKSIM is of the order of 10%, which still makes it worthwhile to use the prioritised cell-rate simulator in the study of traffic with priorities. To complement these experiments and results, another step was considered which consisted of simulating experiments for two different network loads, while maintaining the buffer size.

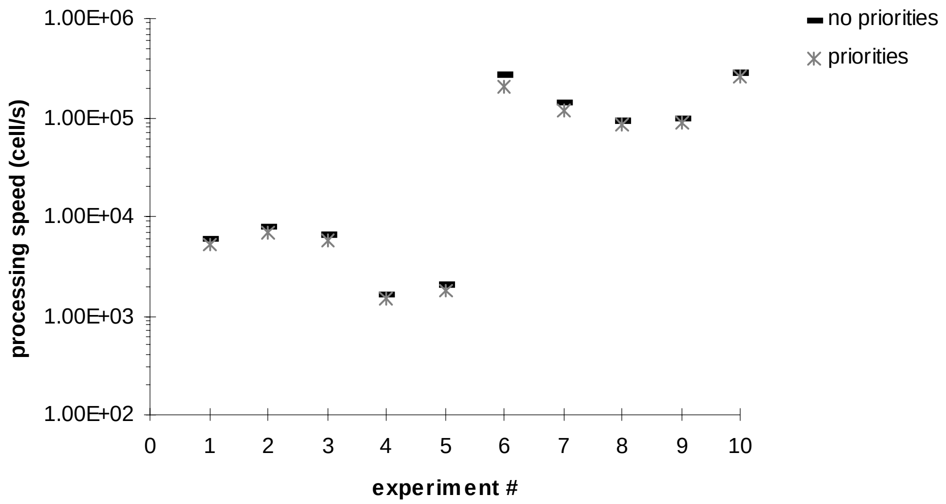


Fig.5.14 - Comparison of processing speed: no priorities *versus* priorities.

A similar set of experiments was carried out in [Pitt93] in order to compare the speed up of the cell-rate simulation method when compared with cell level simulation. For each network load, the *On* and *Off* state durations of the simulated traffic sources are scaled. Each experiment is executed for the case of traffic without priorities and is then repeated under the same experimental conditions for the case of traffic with priorities; in the latter case, the high priority traffic is always taken to represent 30% of the total traffic (see Table 5.12).

The aim of this second phase of experiments is two-fold. On one hand, it will be possible to compare again the processing speed of both the original version of LINKSIM and the prioritised version. On the other hand, by scaling the *On* and *Off* time durations while maintaining the load, any dependence on the priority aspects should become evident.

Traffic Sources & System	
On state mean duration	1.272 ms to 6360 ms
Off state mean duration	11.448 ms to 57240 ms
peak bit rate	10 Mbit/s
mean bit rate	1 Mbit/s
burstiness	10
mean burst length	30 cells to 15000 cells
buffer length	15 cells
threshold size	15 cells
queue capacity	135.85 Mbit/s
network load	40% to 80%

Table 5.12 - Processing speed of prioritised cell-rate simulator: sources and system characteristics.

The next figure (Fig.5.15) gives the obtained results in terms of speed increase when going from the prioritised cell-rate simulator to using the simulator without priorities.

It can be seen (see Fig.5.15) that up to reasonably high burst lengths, there is not much difference between the speed of the simulator without priorities and the prioritised one.

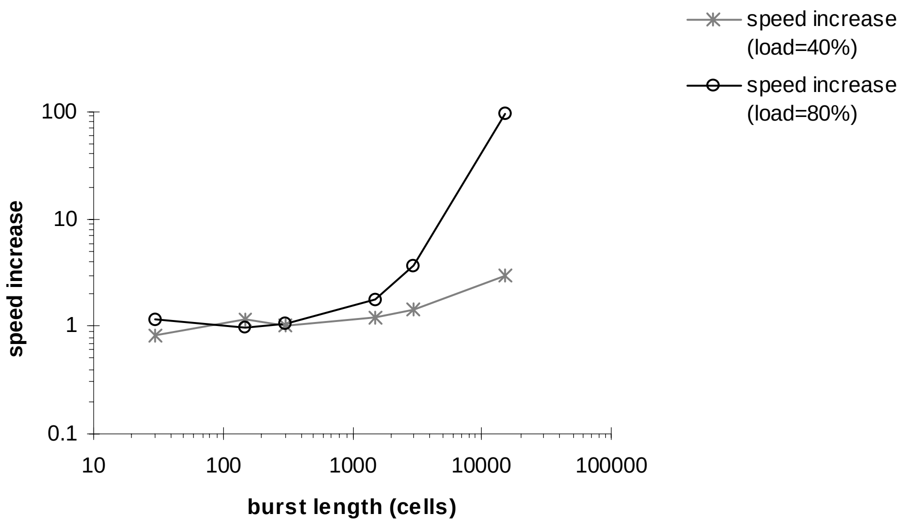


Fig.5.15 - Speed increase of original cell-rate simulator *versus* source's mean burst length.

However, for very high burst lengths, the use of the prioritised simulator becomes less advantageous. This is especially the case for very high network loads, due to the fact that in those situations (i.e., long burst lengths and high loads), more cell loss will be encountered, thus increasing the processing in the queue and the overall processing time of the simulator. Therefore, the prioritised simulator is of maximum value when the sources involved do not have very high mean burst lengths.

5.6. Conclusion

This Chapter described some of the new work developed by the author. In particular, it started by outlining the mode of operation of the *cell-rate simulator* (LINKSIM) used for the *implementation of the loss priority mechanism* Partial Buffer Sharing (PBS). This priority mechanism was taken in an *approximate form*, which does not consider the space in the queue above the threshold. Although this approximation implies some inaccuracy in the produced high priority cell loss results (as obtained by comparison with published simulation results from [Krön91]), the results generally compare well. Moreover, the approximation on the priority mechanism implied a considerable simplification of the implementation.

The process of implementation of the PBS mechanism was followed by its *validation*, which included a *comparison against simulation results* obtained by other authors (as mentioned in the beginning of this Section) and a *comparison with an approximate burst level analysis* that provides an upper bound for the cell loss of traffic with and without priorities. Both methods confirmed the prioritised cell-rate simulator as a useful tool to study traffic with priorities, especially when the proportion of high priority traffic to the total traffic is low. The process of validating the priority mechanism implemented in the cell-rate simulator is concluded in Chapter 6, where a fluid flow model has been extended to the case of traffic with priorities and afterwards compared to the prioritised simulator.

Apart from validating the prioritised cell-rate simulator against previously published results, it was demonstrated that the *typical properties of priority mechanisms* are also verified for the priority mechanism implemented in the cell-rate simulator.

The Chapter concluded with a comparison, in terms of processing speed, of the original cell-rate simulator against the new prioritised cell-rate simulator. Since the enhancement introduced in the original simulator implied a greater computational complexity, a *reduction in the processing speed of the new cell-rate simulator* was expected. A quantification of that reduction was obtained by considering several experiments that were repeated for the cases of traffic with and without priorities, while maintaining the experimental conditions. It was observed that the loss in the processing speed (when using the prioritised cell-rate simulator) is smaller for low network loads and for up to reasonably high burst lengths (in the order of a few thousand cells) of the traffic sources being analysed.

6. Fluid Flow Analysis Applied to PBS

The importance of source models has already been stressed in Chapter 4; they provide, under certain assumptions, a fairly accurate tool to characterise the behaviour of real traffic sources. Taking this into consideration, an exact analysis that provides a formula to calculate cell loss probability was extended by the author in order to take priorities into account and was then compared with the implementation of PBS in a cell-rate simulator, previously described in Section 5.2; this represents new work. The analysis was initially developed in [Scho94] for the case of a single traffic source, modelled as an On/Off source, having access to a finite buffer and where no priorities were considered. In the next Sections, a brief outline is given of the fluid flow analysis introduced in [Scho94], which is followed by the description of the prioritised fluid flow analysis. Also covered in this Chapter is the concept of fluid flow analysis and its usage environment.

6.1. What is Fluid Flow Analysis?

In Section 4.1.3, the Fluid Flow Approximation was briefly described as one possible traffic source model to be used in the characterisation of bursty sources. Also, Section 4.1.1 referred to the effects of studying traffic behaviour under different time scales; one of them was the burst scale. Fluid flow analysis (see [Anic82] and [Tuck88]) is especially appropriate to study ATM multiplexers at this time scale; it considers long time intervals, taking the mean burst period as the time unit. The model focuses on the beginning and termination of bursts and it does not take account of the cell population within each burst. It considers traffic sources converging to a buffer or queue as generating continuous streams of cells, that are characterised by their instantaneous arrival (or flow) rate. Using this type of analysis will imply studying long-term statistics of the system in continuous time.

A classic example of fluid flow analysis and also one of the first, can be found in [Anic82], where this type of analysis is applied to a switch that handles data in a computer system. This author describes the model used in the following way: first, let n information sources alternate between exponentially distributed *On* and *Off* states, and have access to a data-handling switch (see Fig.6.1).

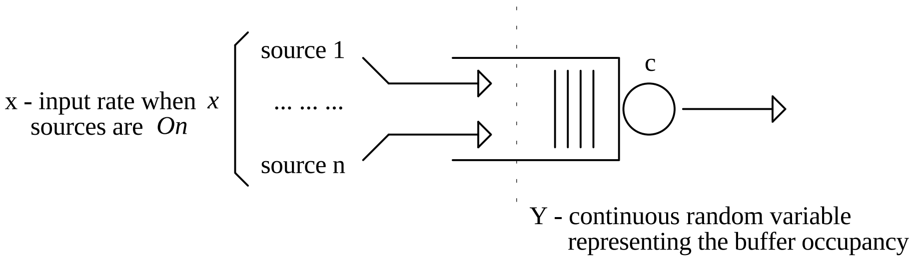


Fig.6.1 - Fluid flow model.

The average *On* period is taken to be the unit of time and the average *Off* period is denoted by $1/\lambda$. Also, the amount of information generated by a source in an average *On* period is taken to be the unit of information. This means that a source transmits at a rate of one unit of information per unit of time, when in the *On* state. Moreover, when x of the total n considered sources are *On*, the receiving rate at the switch will be x. If c represents the transmission rate of the switch then, provided the buffer is not empty, the instantaneous rate of change of the buffer occupancy is $x - c$. When the buffer is empty, it will remain empty while $x \leq c$. The author also considers an infinite buffer and in that situation, a stability condition must be verified:

$$\frac{n \cdot \lambda}{c \cdot (1+\lambda)} < 1, \tag{6.1}$$

that represents the traffic intensity of the system. With this model, it is then possible to evaluate the probability of overflow or the probability of the buffer occupancy exceeding a certain level y (see Fig.6.1); to this end, the author derives the equilibrium buffer distribution using differential equations (see also [Guér91]). This model is particularly useful in buffer dimensioning. The analysis described in [Scho94] is similar (but simpler) to the one of [Anic82], as described in the next Section.

6.2. Discretised Fluid Flow Model

This technique models the burst scale component of ATM queuing. It is a rate based model, in which the queue size varies in discrete steps; the technique deals with rates of flow of cells and the queue begins to fill only when the source rate exceeds the service rate of the queue. In a standard fluid flow analysis (such as the one described in

[Anic82]), the queue size variation would be approximated to a continuous variable. With this new fluid flow analysis, balance equations are used to derive the cell loss probability formula (see [Scho94]). This model assumes the sojourn times in each state to be memoryless and geometrically distributed. Fig.6.2 shows a diagram of the model.

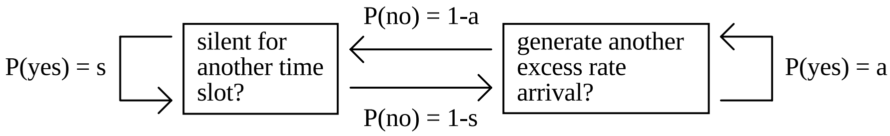


Fig.6.2 **-** The excess rate arrival model.

The closed formula obtained for *cell loss probability* (clp) is given by,

$$\mathrm{clp} = \frac{R - C}{R} \cdot p[N], \tag{6.2}$$

where

N = buffer capacity

p[N] = cell loss probability for excess-rate cells

R = cell arrival rate

C = cell transmission rate.

The exact cell loss probability formula for excess-rate cells is derived by solving the balance equations based on the level crossing approach (see [Scho94]) for the corresponding Markov chain of the process; it is given by the expression,

$$p[N] = \frac{1}{1 + \sum_{i=1}^{N} \left(\frac{s}{a}\right)^{i} \cdot \left(\frac{1-a}{s}\right)}, \tag{6.3}$$

where

$$a = 1 - \frac{1}{E[\mathit{On}\ \mathrm{time}] \cdot (R - C)} \tag{6.4}$$

$$s = 1 - \frac{1}{E[\mathit{Off}\ \mathrm{time}] \cdot C}. \tag{6.5}$$

The expected number of arrivals that increase the queue size (i.e., excess-rate arrivals) corresponds to the expected duration of the *On* period and it is represented by *E[On time]*, while *E[Off time]* is the expected duration of the *Off* period and it is given by the expected number of time slots in an *Off* period.

6.3. Fluid Flow Model with Priorities

It is possible to modify this analysis in order to study traffic with priorities. Two approaches have been considered by the author: the first one consists of inserting the approximate version of Partial Buffer Sharing in the analytical model, as described in Section 5.2; the other approach adds the exact version of the PBS scheme (as presented in Section 3.8.2.2) to the model. The approximate approach will allow a comparison of both cell-rate simulation and analytical techniques, which represents the second phase of validation of the priority mechanism in the cell-rate simulator (see Section 5.3); in this case, the priority mechanism operates at the threshold of the queue. On the other hand, by extending the fluid flow model to the exact version of PBS, it will be possible to compare the approximate and exact analyses, and judge the accuracy obtained with the approximate analysis.

The extension of this fluid flow model in order to study priorities represents new work by the author; it was first described in [Fons95b] and it was compared with the cell-rate simulation technique in [Fons95a].

6.3.1. Approximate Analysis

This approach models burst scale queuing up to the threshold of the queue, giving preference to the high priority traffic when the queue occupancy is at the threshold level. The approximate version of PBS (see Section 5.2) considers the threshold (which will be denoted here by k) of the queue to be equal to the queue length N, i.e., $N=k$. Therefore, with the first approach referred to in Section 6.3, $p[N]$ (now called $p[k]$, to be more coherent) will remain the same as in equation (6.3). However, as traffic with priorities is now being considered, the cell arrival rate, R, will consist of two parts:

$$\mathrm{R} = \mathrm{R}_{\mathrm{lp}} + \mathrm{R}_{\mathrm{hp}}, \tag{6.6}$$

which correspond to the arrival cell rates of the low and high priority traffics, respectively. Two cases must then be considered, knowing that it is $R>C$:

$R_{hp} \leq C$

In this case, no high priority traffic will be lost and the low priority traffic that is not lost is the part that can still be accepted after all the high priority traffic has been accepted by the queue. So,

$$aclp\_lp = p[k] \cdot \frac{R_{lp} - (C - R_{hp})}{R_{lp}} \tag{6.7}$$

$$aclp\_hp = 0 \tag{6.8}$$

$R_{hp} > C$

When the high priority traffic is greater than the transmission rate, all of the low priority traffic will be lost, as well as a proportion of the high priority traffic. Thus,

$$aclp\_lp = p[k] \cdot \frac{R_{lp}}{R_{lp}} = p[k] \tag{6.9}$$

$$aclp\_hp = p[k] \cdot \frac{R_{hp} - C}{R_{hp}} \tag{6.10}$$

6.3.2. Exact Analysis

With this analysis, burst scale queuing is modelled both up to and above the queue's threshold. For the second approach indicated in Section 6.3, let Fig.6.3 represent a queue with length *N* and threshold *k* (k<N). In this situation, the behaviour of the source will be different below and above the threshold *k* but there will only be queuing above the threshold when $R_{hp} > C$. In the case of $R_{hp} \leq C$, the equations used are again the ones indicated for the case of the approximate mechanism.

Let *eclp_lp(hp)* denote the cell loss probability for low (high) priority traffic with the exact mechanism and let $p_e[i]$ be the probability that an excess-rate arrival sees *i* cells in the queue.

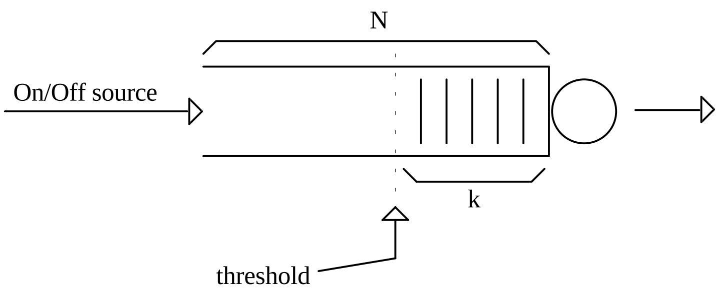


Fig.6.3 - Queue model for combined analysis and exact PBS.

The equation for the high priority cell loss is similar to (6.10), but now an excess-rate arrival of high priority is lost only when it sees a full queue, i.e., $p_e[N]$. On the other hand, arrivals of low priority are lost whenever they see a queue size at or above the threshold level. Thus,

$$\mathrm{eclp\_lp} = \sum_{i=k}^{N} p_e[i] \tag{6.11}$$

$$\mathrm{eclp\_hp} = p_e[N] \cdot \left(R_{hp} - C\right) / R_{hp} \tag{6.12}$$

To calculate $p_e[i]$ when $i \geq k$, we need to know whether an excess-rate arrival is of high priority or of low priority. The queue size changes only if the excess-rate arrival is of high priority. So, for $R_{hp} > C$, let

$$P_{hp} = \frac{R_{hp} - C}{R - C} = \mathrm{Prob}\{\ \textit{an excess-rate cell is of high priority}\ \} \tag{6.13}$$

$$P_{lp} = 1 - P_{hp} = \frac{R_{lp}}{R - C} = \mathrm{Prob}\{\ \textit{an excess-rate cell is of low priority}\ \} \tag{6.14}$$

For the level between states *N-1* and *N*, equating probabilities gives:

$$P_{hp} \cdot a \cdot p_e[N-1] = p_e[N] \cdot (1-a) \tag{6.15}$$

$$\Leftrightarrow p_e[N-1] = \frac{1-a}{a \cdot P_{hp}} \cdot p_e[N] \tag{6.16}$$

where the left hand side (LHS) of equation (6.15) is the probability of crossing up (i.e., arrival *i* is of high priority and sees *N-1* cells in the queue and arrival *i+1* is of either priority and sees *N* cells in the queue) and the right hand side (RHS) is the probability of crossing down (i.e., arrival *i* sees *N* cells in the queue and is lost, while arrival *i+1* sees *N-1* cells in the queue or fewer). For the level between states *N-2* and *N-1*,

$$P_{hp} \cdot a \cdot p_e[N-2] = p_e[N] \cdot (1-a) \cdot s +$$

$$+ p_e[N-1] \cdot P_{lp} \cdot (1-a) + p_e[N-1] \cdot P_{hp} \cdot (1-a) \cdot s \tag{6.17}$$

$$\Leftrightarrow P_{hp} \cdot a \cdot p_e[N-2] = \left\{(1-a) \cdot P_{lp} + s \cdot P_{hp}\right\} \cdot p_e[N-1] \tag{6.18}$$

$$\Leftrightarrow p_e[N-2] = \frac{1-a}{a \cdot P_{hp}} \cdot \frac{(1-a) \cdot P_{lp} + s \cdot P_{hp}}{a \cdot P_{hp}} \cdot p_e[N] \tag{6.19}$$

Here, for the LHS of equation (6.17), high priority arrival *i* sees *N-2* cells in the queue and arrival *i+1* (of either priority) sees *N-1* cells in the queue. The first term on the RHS corresponds to the case in which arrival *i* (of either priority) sees *N* cells in the queue and is lost, while arrival *i+1* sees *N-2* cells in the queue or fewer. The third term is when high priority arrival *i* sees *N-1* cells in the queue, thus maximising the queue occupancy, and arrival *i+1* sees *N-2* cells in the queue or fewer. The second term represents the case when low priority arrival *i* sees *N-1* cells in the queue and is lost, whilst arrival *i+1* sees *N-2* cells in the queue or fewer.

In a similar way, the comparison of levels *N-3* and *N-2* produces

$$\begin{aligned} P_{hp} \cdot a \cdot p_e[N-3] = {} & p_e[N] \cdot (1-a) \cdot s^2 + \\ & + p_e[N-1] \cdot P_{lp} \cdot (1-a) \cdot s + p_e[N-1] \cdot P_{hp} \cdot (1-a) \cdot s^2 + \\ & + p_e[N-2] \cdot P_{lp} \cdot (1-a) + p_e[N-2] \cdot P_{hp} \cdot (1-a) \cdot s \end{aligned} \tag{6.20}$$

… … … … …

$$\Leftrightarrow P_{hp} \cdot a \cdot p_e[N-3] = \left\{(1-a) \cdot P_{lp} + s \cdot P_{hp}\right\} \cdot p_e[N-2] \tag{6.21}$$

$$\Leftrightarrow p_e[N-3] = \frac{1-a}{a \cdot P_{hp}} \cdot \left(\frac{(1-a) \cdot P_{lp} + s \cdot P_{hp}}{a \cdot P_{hp}} \right)^2 \cdot p_e[N]. \tag{6.22}$$

Lower levels are compared in a similar way, except that below the threshold, an arrival of either priority increases the queue size by one. So, for levels *N-(N-k)* and *N-(N-k-1)*, i.e., the point when the queue occupancy reaches the threshold, the following relation is obtained:

$$\begin{aligned} P_{hp} \cdot a \cdot p_e[N-(N-k)] = {} & p_e[N] \cdot (1-a)^{N-k-1} + p_e[N-1] \cdot P_{lp} \cdot (1-a) \cdot s^{N-k-2} + \\ & + p_e[N-1] \cdot P_{hp} \cdot (1-a) \cdot s^{N-k-1} + \ldots + p_e[N-(N-k-2)] \cdot P_{lp} \cdot (1-a) \cdot s + \\ & + p_e[N-(N-k-2)] \cdot P_{hp} \cdot (1-a) \cdot s^2 + p_e[N-(N-k-1)] \cdot P_{lp} \cdot (1-a) + \\ & + p_e[N-(N-k-1)] \cdot P_{hp} \cdot (1-a) \cdot s \end{aligned} \tag{6.23}$$

… … … … …

$$\Leftrightarrow P_{hp} \cdot a \cdot p_e[N-(N-k)] = \left\{(1-a) \cdot P_{lp} + s \cdot P_{hp}\right\} \cdot p_e[N-(N-k-1)] \tag{6.24}$$

$$\Leftrightarrow p_e[N-(N-k)] = \frac{1-a}{a \cdot P_{hp}} \cdot \left(\frac{(1-a) \cdot P_{lp} + s \cdot P_{hp}}{a \cdot P_{hp}} \right)^{N-k-1} \cdot p_e[N]. \tag{6.25}$$

At levels *N-(N-k+1)* and *N-(N-k)*, just below the threshold, because both low and high priority cells are handled in the same way, the balance equations will give,

$$a \cdot p_e[N-(N-k+1)] = p_e[N] \cdot (1-a)^{N-k} + p_e[N-1] \cdot P_{lp} \cdot (1-a) \cdot s^{N-k-1} +$$

$$+p_e[N-1] \cdot P_{hp} \cdot (1-a) \cdot s^{N-k} + \ldots + p_e[N-(N-k-1)] \cdot P_{lp} \cdot (1-a) \cdot s +$$

$$+p_e[N-(N-k-1)] \cdot P_{hp} \cdot (1-a) \cdot s^2 + p_e[N-(N-k)] \cdot P_{lp} \cdot (1-a) +$$

$$+p_e[N-(N-k)] \cdot P_{hp} \cdot (1-a) \cdot s \tag{6.26}$$

… … … … …

$$\Leftrightarrow a \cdot p_e[N-(N-k+1)] = \left\{(1-a) \cdot P_{lp} + s \cdot P_{hp}\right\} \cdot p_e[N-(N-k)] \tag{6.27}$$

$$\Leftrightarrow p_e[N-(N-k+1)] = \frac{1-a}{a} \cdot \left(\frac{(1-a) \cdot P_{lp} + s \cdot P_{hp}}{a \cdot P_{hp}}\right)^{N-k} \cdot p_e[N]. \tag{6.28}$$

After some algebraical manipulation, we obtain,

$$p_e[N-i] = \begin{cases} p_e[N] \cdot \dfrac{1-a}{a.P_{hp}} \cdot (\text{ratio})^{i-1} & ; \; i = 1,\ldots,N-k \\ p_e[N] \cdot \dfrac{1-a}{a} \cdot \left(\dfrac{s}{a}\right)^{i-(N-k+1)} \cdot (\text{ratio})^{N-k} & ; \; i = N-k+1,\ldots,N \end{cases} \tag{6.29}$$

where

$$\text{ratio} = \frac{(1-a) \cdot P_{lp} + s \cdot P_{hp}}{a \cdot P_{hp}}. \tag{6.30}$$

Since the queue system considered has a finite number of states *(N+1)*, the summation of the probabilities for the system to be in each state should be equal to one (which will allow the direct calculation of $p_e[N]$). Therefore, taking the system of equations (6.29) into account,

$$\sum_{i=0}^{N} p_e[i] = 1 \Leftrightarrow p_e[N] + \sum_{i=1}^{N} p_e[N-i] = 1 \Leftrightarrow p_e[N] +$$

$$+p_e[N] \cdot \frac{1-a}{a.P_{hp}} \cdot \sum_{i=0}^{N-k-1} (\text{ratio})^i + p_e[N] \cdot \frac{1-a}{a} \cdot (\text{ratio})^{N-k} \cdot \sum_{i=0}^{k-1} \left(\frac{s}{a}\right)^i = 1 \Leftrightarrow$$

$$\Leftrightarrow p_e[N] = \frac{1}{1 + \dfrac{1-a}{a} \cdot \left[\dfrac{1}{P_{hp}} \cdot \displaystyle\sum_{i=0}^{N-k-1} (\text{ratio})^i + (\text{ratio})^{N-k} \cdot \sum_{i=0}^{k-1} \left(\frac{s}{a}\right)^i\right]}. \tag{6.31}$$

In some cases, formula (6.31) can be simplified. In particular, if $N=k$ (in which case the approximated PBS mechanism is being considered), then formula (6.31) reduces itself easily to equation (6.3).

To determine the value of (6.11), the value of $p_e[N]$ is used as follows,

$$\sum_{i=k}^{N} p_e[i] = p_e[N] + \sum_{i=1}^{N-k} p_e[N-i] = p_e[N] + p_e[N] \cdot \frac{1-a}{a \cdot P_{hp}} \cdot \sum_{i=1}^{N-k} (\text{ratio})^{i-1} =$$

$$= p_e[N] \cdot \left[1 + \frac{1-a}{a . P_{hp}} \cdot \sum_{i=0}^{N-k\text{-}1} (\text{ratio})^{i} \right]. \tag{6.32}$$

6.4. Cell-Rate Simulation *versus* Fluid Flow Analysis

The previous Sections have described both the discretised fluid flow analysis developed in [Scho94] and the extension of this analysis in order to handle priorities using two versions of the same priority mechanism, Partial Buffer Sharing. The next step consists of comparing the developed approximate prioritised analysis with the prioritised cell-rate simulator described in Section 5.2; the necessity of this step has already been stressed in Section 5.3, as a second validation phase of the approximate priority mechanism implemented in the cell-rate simulator LINKSIM.

Since the exact priority mechanism was also added to the fluid flow analysis, it is also possible to evaluate the accuracy of the approximate prioritised analysis when compared with the exact prioritised analysis. Finally, it is also important to plot results from both the original cell-rate simulator and the fluid flow analysis for the case where no traffic with priorities is considered. A good agreement between the two techniques will show that the cell-rate simulator without priorities represents accurately the behaviour of a traffic source modelled by the fluid flow analysis; if this does not happen, then it will be meaningless to compare the results obtained with the prioritised simulator against the approximate fluid flow analysis with priorities.

This phase of the validation process referred to in Section 5.3 uses a traffic model that models only one traffic source; this is clearly a disadvantage, since the cell-rate simulator can be used to study the multiplexing of several traffic sources. This is because, until now, it has not been feasible to extend this fluid flow analysis in order to study the multiplexing of several traffic sources.

All the traffic experiments described in the next two Sections use traffic modelled as On/Off sources (according to Sections 6.2 and 6.3) and the considered mean *On* and *Off* state durations of the sources are geometrically distributed. Also, the results are always plotted by using the same logarithmic scale for the cell loss in order to make it easy to compare the several (simulation and analytical) approaches.

6.4.1. Traffic without Priorities

In order to evaluate the accuracy of the results obtained with the cell-rate simulator, experiments were carried out that compared the simulator with the fluid flow analysis without priorities. Table 6.1 gives all the parameters, variable and fixed ones, used to carry out a set of three experiments.

Common parameters	***Variable parameters***		
C = 500 cell/s	*load = 60%*	*load = 70%*	*load = 80%*
On = 7.5 ms / Off = 12.5 ms	R = 800 cell/s	R = 933 cell/s	R = 1067 cell/s
N = k	k = 4, 8, 12, ..., 28	k = 4, 8, 12, ..., 40	k = 4, 8, 12, ..., 40

Table 6.1 - Increase in the network load by increasing the input rate: traffic parameters.

The notation used (in Table 6.1) is the same already used in Sections 6.2 and 6.3 to describe the fluid flow analyses. In all three experiments, the link transmission rate *C* and average *On* and *Off* state durations were kept fixed. The variable part of these experiments comprised increasing the load of the system by increasing the peak bit rate of the On/Off source. Also, for each of the loads, a series of buffer lengths was considered with increments of 4 cells, in order to find a trend in the results obtained by both simulation and analysis. Notice also that for the lowest network load considered, the buffer length goes as far as 28 cells, although there are still analytical results for buffer lengths up to 40 cells; other graphs following will also appear to be incomplete because the cell-rate simulator used is not able, in reasonable computing time, to simulate cell losses lower than values in the order of 10^{-7} and produce good statistical measures (e.g., confidence intervals and correlation measures) at the same time.

Fig.6.4 presents the cell loss probability obtained for three different network loads, using both the cell-rate simulator and the fluid flow analysis without priorities. The confidence intervals are not shown for the simulation results because they would be meaningless if they had been added to the graph; in fact, the ratio of the confidence interval width to the corresponding cell loss value is of the order of 10^{-2} for all experiments (including the ones described in the next Section).

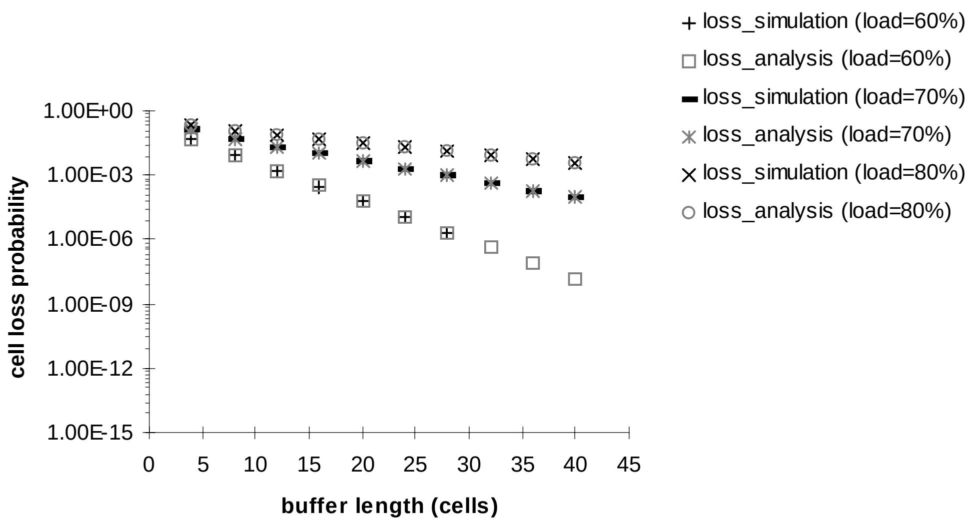


Fig.6.4 - Simulation *versus* approximate analysis, for traffic without priorities: increasing the network load by increasing the input rate (from Table 6.1).

The graph (see Fig.6.4) shows that there is a very good agreement between the cell loss probability results from cell-rate simulation (given by the lines labelled *loss_simulation*, where the different network loads are represented) and the fluid flow analysis results for cell loss probability (given by the lines labelled *loss_analysis*, each of these lines representing a different load).

Common parameters	***Variable parameters***		
	load = 60%	*load = 65%*	*load = 70%*
R = 1000 cell/s C = 700 cell/s N = k	On = 7.2 ms Off = 10 ms k = 4, 8, 12, ..., 24	On = 8.3 ms Off = 10 ms k = 4, 8, 12, ..., 32	On = 9.6 ms Off = 10 ms k = 4, 8, 12, ..., 40

Table 6.2 - Increase in the network load by increasing the *On* source time: traffic parameters.

It can be seen that, as the network load increases, the cell loss suffered by the On/Off source increases as well. It is also worth noticing that an increase in the network load does not appear to affect the accuracy of the simulation results.

Another similar set of simulations was undertaken which considers the *On* time of the used traffic source as varying and altering the network load, while the other source characteristics, as well as the system characteristics are kept as fixed parameters; this can be seen in Table 6.2. Again, the cell loss probability results obtained with both simulation and analytical approaches are almost coincident (see Fig.6.5), thus proving the accuracy of the cell-rate simulator.

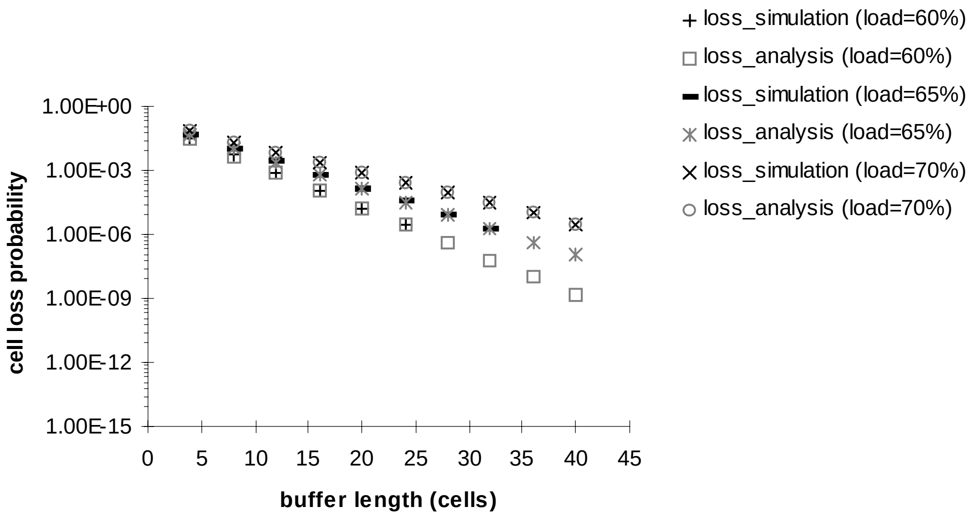


Fig.6.5 - Simulation *versus* approximate analysis, for traffic without priorities: increasing the network load by increasing the *On* source time (from Table 6.2).

The graph (see Fig.6.5) shows that an increase in the network load caused by an increase in the *On* average state duration of the source still has the same effect in terms of cell loss as when the network load increase is caused by a source input rate increase.

6.4.2. Traffic with Priorities

In this Section, two main comparisons take place: the first one is concerned with investigating whether the prioritised cell-rate simulator and the approximate fluid flow analysis produce the same type of results, under diverse conditions. The other main comparison consists of (see Section 6.4) evaluating the different characteristics of the approximate and exact fluid flow analyses. This will have an impact on the decision of when the prioritised cell-rate simulator should and should not be utilised.

Common parameters			***Variable parameters***
traffic source		queue	queue
R = 1000 cell/s Rlp = 0.6R	On = 1.8 ms Off = 10 ms	C = 190 cell/s k = 0.8N	k = 4, 8, 12, ..., 32

Table 6.3 - Increasing the queue's length: traffic parameters for approximate priorities.

The first experiment used to compare the prioritised simulator and the fluid flow analysis with the approximate priority mechanism involved considering a varying buffer length for the queue. However, the ratio of the threshold size to the queue length considered remains constant, as well as all source and other system parameters (see Table 6.3); the comparison was performed for a 80% network load. The cell loss values obtained for both the prioritised simulator and the approximate analysis are given in Fig.6.6.

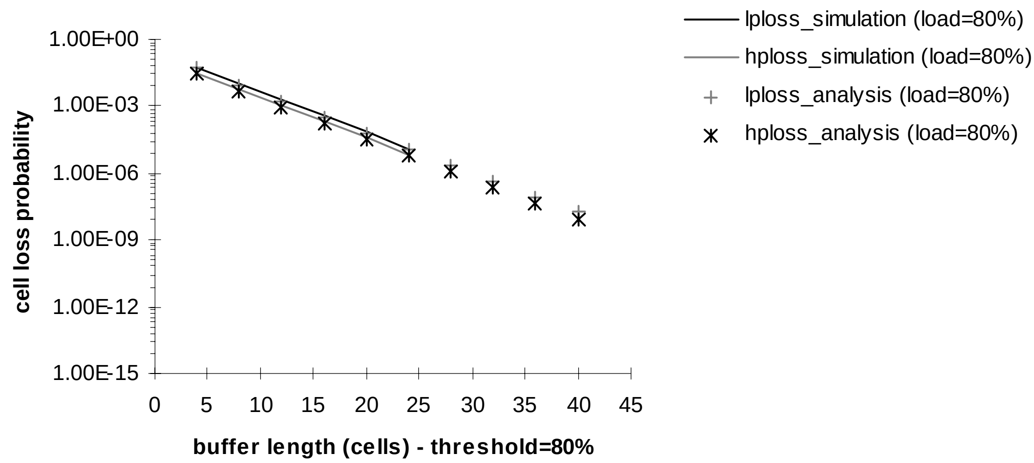


Fig.6.6 - Simulation *versus* approximate analysis, for traffic with priorities: increasing the queue's length (from Table 6.3).

The graph (see Fig.6.6) shows that the results obtained for a 80% network load with both simulation (given by lines *lploss_simulation* and *hploss_simulation* for low and high priority cell loss, respectively) and analytical techniques (given by lines *lploss_analysis* and *hploss_analysis* for low and high priority cell loss, respectively) are very similar. Notice the small difference between the low and high priority cell loss values, implying a poor improvement in terms of cell loss for the high priority traffic; this is because the proportion of high priority traffic to the total traffic is fairly high (40% in this case).

Common parameters	***Variable parameters***	
C = 200 cell/s	*load = 60%*	*load = 80%*
Rlp = 0.8R	R = 1120 cell/s	R = 1500 cell/s
On = 1.8 ms	Rlp = 896 cell/s	Rlp = 1200 cell/s
Off = 15 ms	Rhp = 224 cell/s	Rhp = 300 cell/s
k = 0.8N	k = 4, 8, 12, ..., 24	k = 4, 8, 12, ..., 40

Table 6.4 - Increase in the network load by increasing the input rate: traffic parameters for approximate priorities.

In fact, it was verified in Section 5.4 that "... the performance of priority mechanisms is better when the high priority traffic represents only a small proportion of the total traffic than in the case of the high priority traffic representing the bulk of the total traffic ...". In order to confirm the accuracy of the prioritised simulator when compared with the approximate analysis, two other sets of experiments were carried out; they are described next.

Another interesting way of comparing the prioritised simulator with the approximate analysis would be to vary the network load by increasing the input rate of the source. Table 6.4 shows the considered source and queue traffic parameters.

The results obtained from both techniques have been plotted in Fig.6.7 and Fig.6.8 for different network loads, so that it is easier to visualise the results; here, it is possible to see that, as expected, when the network load increases, both the low and the high priority traffic cell loss increase as well and in the same proportion. Again, there is a great similarity between the cell loss values obtained with the prioritised simulator and the ones obtained with the approximate fluid flow analysis. Moreover, this is verified for both low and high priority traffic cell losses.

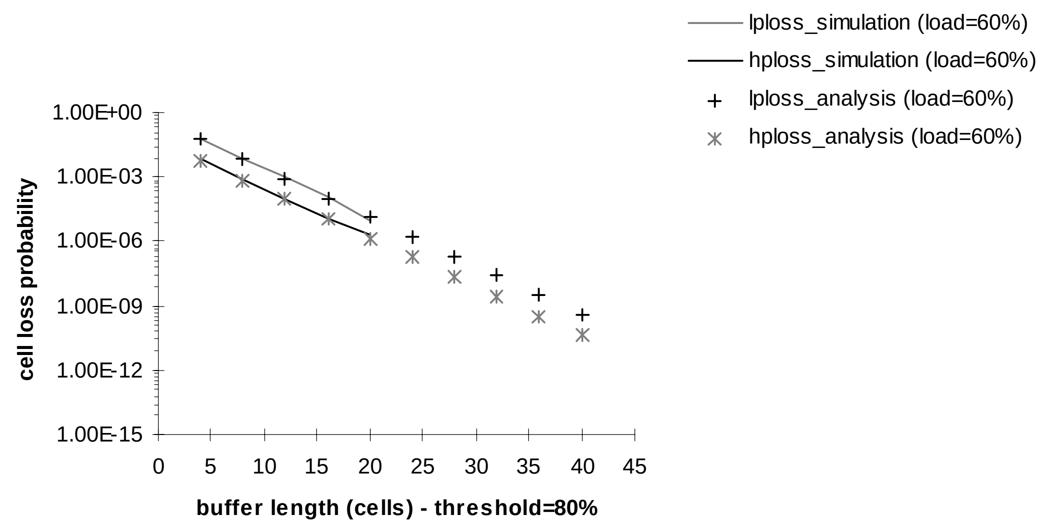


Fig.6.7 - Simulation *versus* approximate analysis, for traffic with priorities: increasing the input rate for a 60% network load (from Table 6.4).

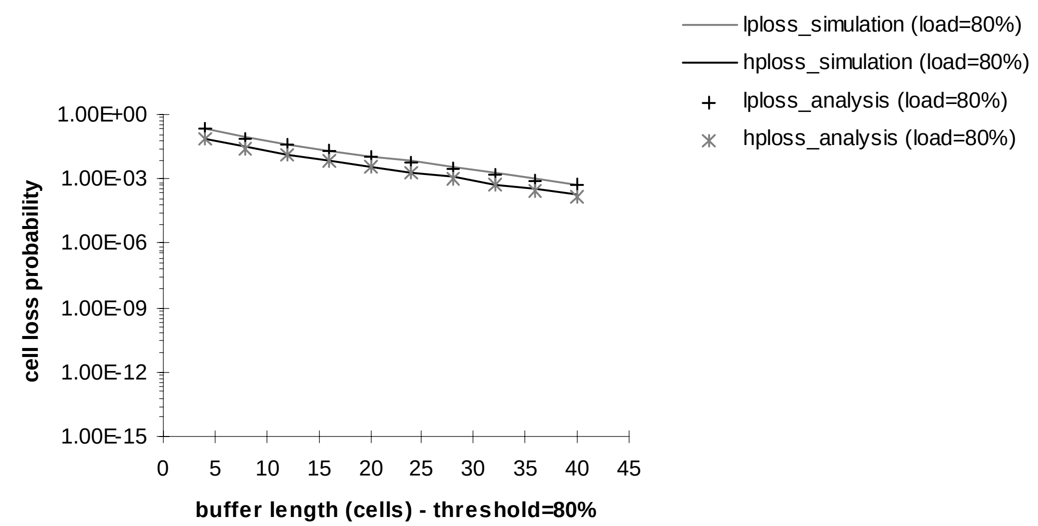


Fig.6.8 - Simulation *versus* approximate analysis, for traffic with priorities: increasing the input rate for a 80% network load (from Table 6.4).

Finally, another set of results was obtained by considering an On/Off traffic source for which the mean *On* state duration is varied, while all other conditions in the system remain fixed. A description of the traffic parameters used in these experiments is given in Table 6.5.

Common parameters	***Variable parameters***	
R = 1000 cell/s	*load = 80%*	*load = 90%*
Rlp = 0.8R	On = 1.8 ms	On = 2.1 ms
C = 190 cell/s	Off = 10 ms	Off = 10 ms
k = 0.8N	k = 4, 8, 12, ..., 28	k = 4, 8, 12, ..., 40

Table 6.5 - Increase in the network load by increasing the *On* source time: traffic parameters for approximate priorities.

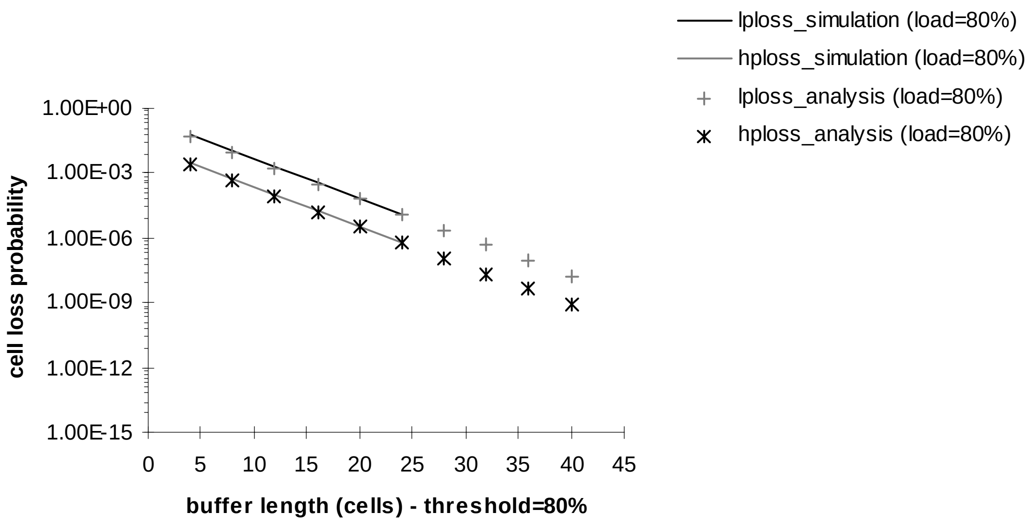


Fig.6.9 - Simulation *versus* approximate analysis, for traffic with priorities: increasing the *On* source time for a 80% network load (from Table 6.5).

In Fig.6.9 and Fig.6.10 (shown separately for different network loads), the varying threshold values (which nevertheless always represents 80% of the total size of the queue) are plotted against the cell loss probability for both low and high priority traffic (considered as 20% of the total traffic), which has been obtained with the prioritised cell-rate simulator and with the approximate analysis. The low and high priority cell loss results obtained for network loads of 80% and 90% show a similar pattern to that plotted in Fig.6.7 and Fig.6.8, respectively. This indicates that the prioritised cell-rate simulator is in agreement with the prioritised approximate analysis, which further validates the implementation of the approximate version of Partial Buffer Sharing in the cell-rate simulator (see Section 5.3).

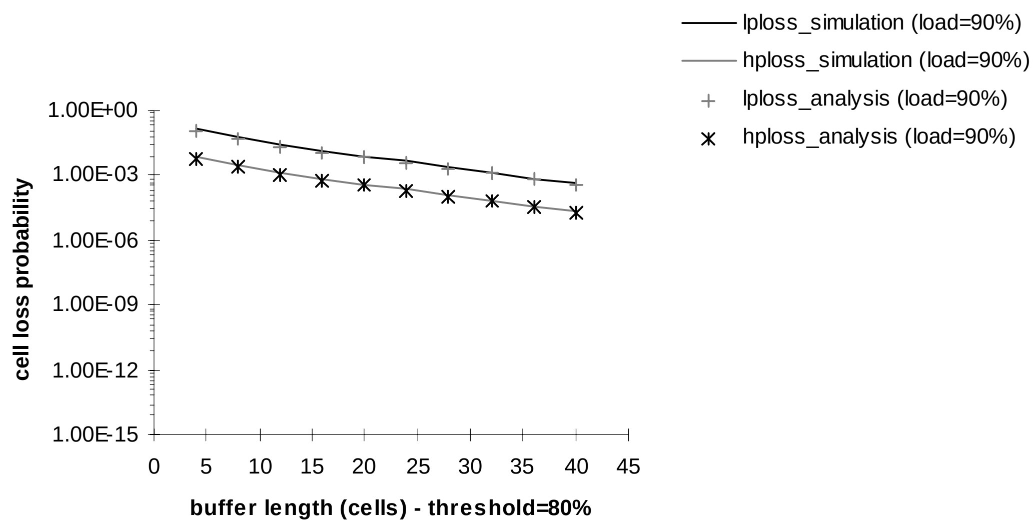


Fig.6.10 - Simulation *versus* approximate analysis, for traffic with priorities: increasing the *On* source time for a 90% network load (from Table 6.5).

As referred to in the beginning of this Section, it is also important to compare the two prioritised fluid flow analyses developed here (see Section 6.3): in one, an approximate version of the Partial Buffer Sharing mechanism is used; in the other analysis, the exact priority mechanism is included. Bearing this in mind, another set of tests was carried out.

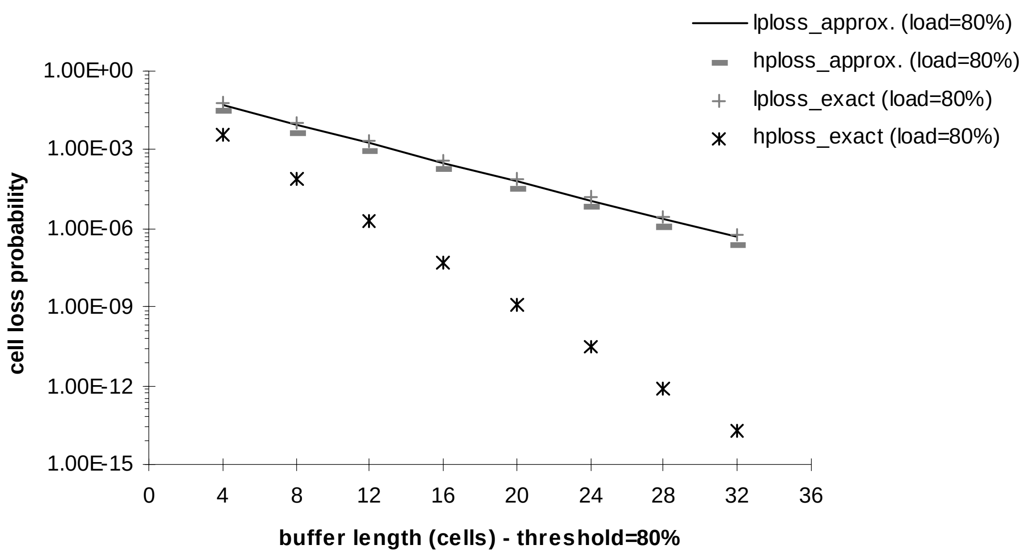


Fig.6.11 - Approximate analysis *versus* exact analysis, for traffic with priorities: increasing the queue’s length (from Table 6.3).

For the first test, an On/Off traffic source was used with traffic parameters described previously in Table 6.3 for a 80% network load. The only variable parameter is the threshold size of the queue, as for the experiment plotted in Fig.6.6.

The previous graph (in Fig.6.11) shows that, as the threshold size of the queue increases (and consequently, also the buffer length), the low priority cell loss results obtained with both the approximate and exact analytical methods are virtually identical; notice again the poor improvement in the high priority cell loss, compared with the low priority cell loss for the particular values chosen.

On the other hand, the high priority traffic experiences less cell loss with both analytical approaches, but the approximate results overestimate the high priority cell loss. This is to be expected because any high priority that has to be queued above the threshold with the exact analysis is assumed lost with the approximate analysis.

Common parameters	***Variable parameters***	
C = 200 cell/s	*load = 60%*	*load = 80%*
Rlp = 0.6R	R = 1120 cell/s	R = 1500 cell/s
On = 1.8 ms	Rlp = 672 cell/s	Rlp = 900 cell/s
Off = 15 ms	Rhp = 448 cell/s	Rhp = 600 cell/s
k = 0.6N	k = 3, 6, 9, ..., 30	k = 3, 6, 9, ..., 30

Table 6.6 - Increase in the network load by increasing the input rate: traffic parameters for approximate and exact priorities.

Apart from these experiments, it was also investigated whether the same behaviour would be observed for the low and high priority cell loss results from both analytical approaches when several network loads were considered, while maintaining the ratio of the threshold size to the buffer length and the proportion of the high priority traffic to the total traffic. The characteristics of the traffic source and the queue used in this experiment are given by Table 6.6.

A reading of the cell loss results plotted in Fig.6.12 and Fig.6.13 (shown separately for different network loads) shows that when the network load increases, the cell loss suffered by both low and high priority traffic increases as well, but the behaviour of the low priority cell loss relative to the high priority cell loss remains similar to that observed in Fig.6.11.

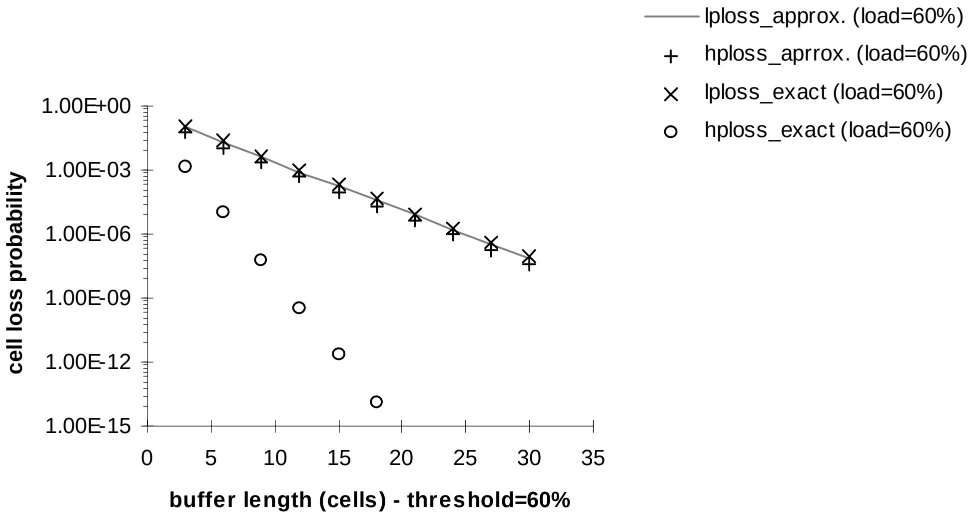


Fig.6.12 - Approximate analysis *versus* exact analysis, for traffic with priorities: increasing the input rate for a 60% network load (from Table 6.6).

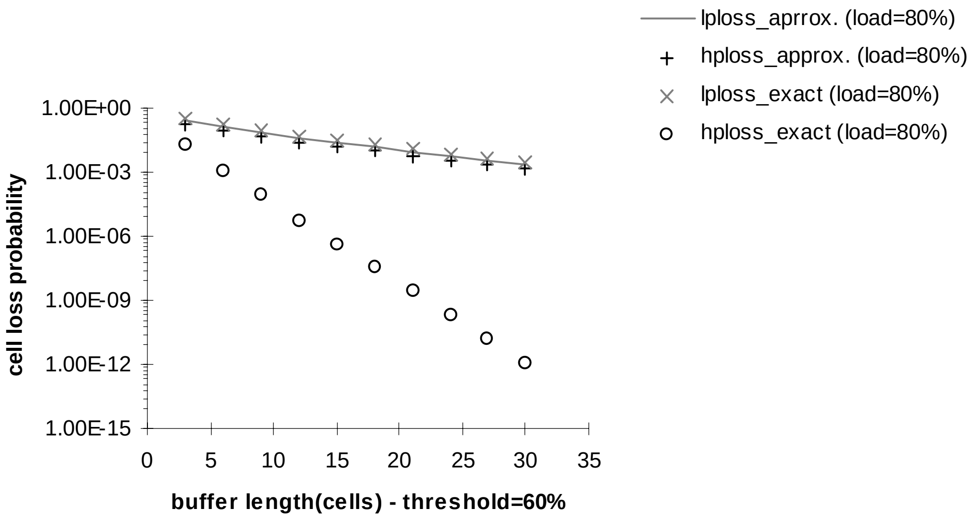


Fig.6.13 - Approximate analysis *versus* exact analysis, for traffic with priorities: increasing the input rate for a 80% network load (from Table 6.6).

The cell loss results given in Fig.6.12 for a 60% network load are represented by the lines labelled as *lploss_approx.* and *hploss_approx.* (for the low and high cell loss results with the approximate analysis, respectively), as well as *lploss_exact* and *hploss_exact* (for the low and high cell loss results with the exact analysis, respectively); the same applies to Fig.6.13, for a 80% network load.

Common parameters	***Variable parameters***	
R = 1000 cell/s	*load = 70%*	*load = 90%*
Rlp = 0.8R	On = 1.5 ms	On = 2.1 ms
C = 190 cell/s	Off = 10 ms	Off = 10 ms
k = 0.6N	k = 3, 6, ..., 30	k = 3, 6, 9, ..., 30

Table 6.7 **-** Increase in the network load by increasing the *On* source time: traffic parameters for approximate and exact priorities.

Finally, another test (similar to that of Fig.6.9 and Fig.6.10, respectively) was considered where the network load was varied by altering the mean *On* state duration of the used traffic source. Once more, it was expected to find a similar behaviour in the cell loss results to that observed in Fig.6.11. Table 6.7 describes the traffic parameters used. For this experiment, the threshold of the queue was always taken to be 40% smaller than the buffer length and the high priority traffic represents 20% of the total traffic.

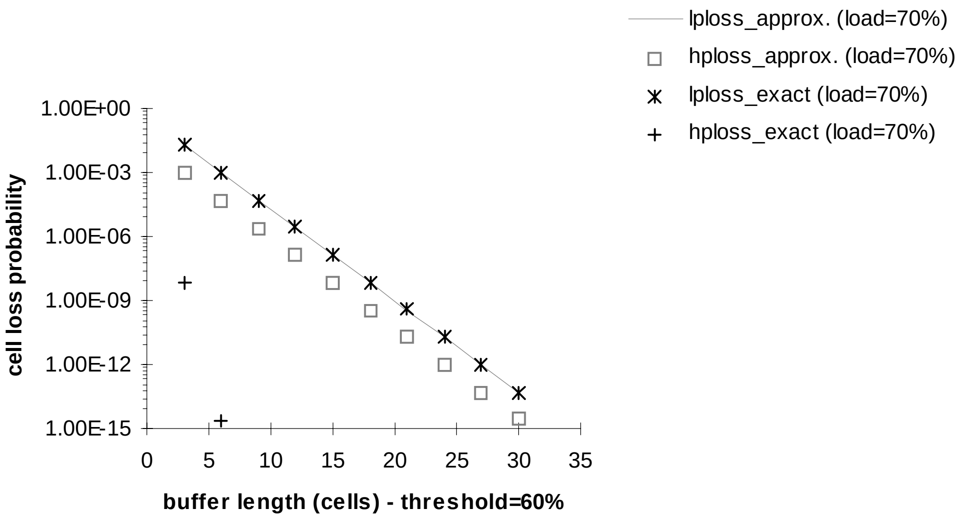


Fig.6.14 **-** Approximate analysis *versus* exact analysis, for traffic with priorities: increasing the *On* source time for a 70% network load (from Table 6.7).

Fig.6.14 and Fig.6.15 (shown separately for different network loads) give the low and high priority cell loss results obtained with both analytical approaches; they show that, under different network conditions, the low priority cell loss results are almost coincident using either analytical method. Notice the reasonable improvement in the high priority cell loss (compared to the low priority cell loss) obtained when using priorities; this is possible because of the small proportion (20% in this case) of the high priority traffic to the total traffic (see Table 6.7 and Section 5.4).

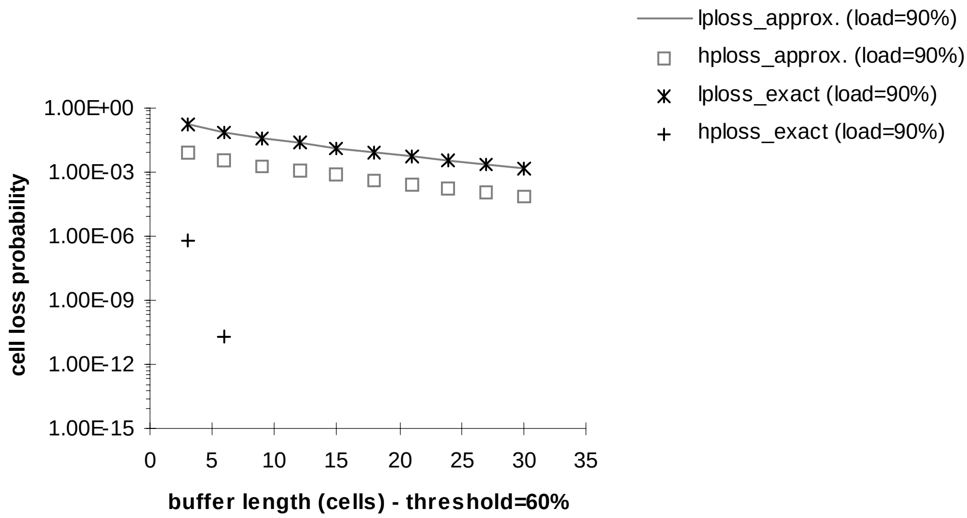


Fig.6.15 - Approximate analysis *versus* exact analysis, for traffic with priorities: increasing the *On* source time for a 90% network load (from Table 6.7).

The high priority cell loss results are overestimated when the approximate analysis is chosen. Therefore, taking into account the very similar behaviour of the prioritised cell-rate simulator and the approximate analysis, it is possible to conclude that, when studying priorities with the prioritised cell-rate simulator, the low priority cell losses will be very accurate, while the high priority cell loss results will be overestimated. This does not invalidate the usage of the prioritised cell-rate simulator since in most real cases, the high priority traffic is predicted to be a small proportion of the total traffic.

6.5. Conclusion

In this Chapter, a *fluid flow model* (for one On/Off modelled traffic source feeding an ATM buffer) initially developed in [Scho94] was extended to the case of traffic with priorities by using the mode of operation of a Partial Buffer Sharing mechanism. The space priority mechanism was considered both according to the approximation described in Section 5.2 and according to Section 3.8.2.2. By doing so, it was possible to complete the *validation process* (started in Section 5.3.1) of the priority mechanism described in Chapter 5. This consisted of comparing the prioritised cell-rate simulator described in Section 5.2 with the prioritised analysis; the results from both analytical and simulation techniques were found to be very similar, thus indicating that the prioritised cell-rate simulator represents correctly the behaviour of a network link where priorities are taken into account.

The two analytical approaches were also compared in order to evaluate the discrepancy in the results obtained with the approximate priority mechanism when compared to the prioritised analysis using the mechanism described in Section 3.8.2.2. In this case, it was found that the cell-rate simulator with priorities gives an accurate prediction of the cell loss for low priority traffic and it overestimates the prediction of the high priority cell loss. Although this is a particularly important drawback of the prioritised cell-rate simulator (because the whole rationale is to improve the high priority cell loss probability), the implementation of the approximate priority mechanism produces high priority cell loss results that are conservative.

7. UPC and Traffic with/without Priorities

Usage Parameter Control (UPC) was described in Section 3.6.2 as a mechanism that monitors the traffic accepted to a network, in order to protect the network from malicious or unintentional misbehaviour of the sources generating traffic and therefore ensure that the QoS of well-behaved sources is not affected. When the UPC finds traffic that is violating its traffic contract (i.e., the contract established at call set-up to be used by the CAC function), actions are taken on that traffic that can result in either the discarding or tagging of violating cells.

Tagging consists of marking cells as *not-so-important* so that they can be lost first in case of congestion. If tagging is the option adopted by the UPC, then the Cell Loss Priority (CLP) bit present in the header of ATM cells can be used to turn cells that are by default of high priority (i.e., with CLP=0) into low priority cells (i.e., with CLP=1), as proposed by ITU in [ITU94].

On the other hand, it is worth noticing that the original aim of introducing the CLP bit as a field in the header of ATM cells was purely to attribute priority levels to cells. This leads us to the present problem: can the CLP bit be used efficiently for two different purposes? And if not, where and when should it be used? In the next Sections, answers to these questions are sought.

To that end, this Chapter starts by describing both the UPC algorithm used (i.e., the Leaky Bucket in this case) and the four policing scenarios that were studied to investigate the problem. The scenarios investigated are:

* traffic without priorities being monitored by the UPC, and
  - cells are discarded when found to be non-compliant; or
  - misbehaving traffic is tagged by the UPC;
* traffic with priorities being policed by the UPC in two stages, the first of which acts only on the high priority traffic, while the second stage acts on the total traffic and always discards any violating traffic, and
  - violating high priority traffic is discarded in the first stage; or
  - misbehaving high priority traffic is tagged in the first stage.

Then, it is explained how the policing scenarios were implemented in the cell-rate simulator previously used in Chapter 5 to add the priority mechanism Partial Buffer Sharing. All the scenarios have used that prioritised cell-rate simulator to implement the Leaky Bucket algorithm. However, the policing scenario that considers the discarding UPC action for traffic without priorities does not make use of the inherent loss priority mechanism present in the prioritised cell-rate simulator.

The Chapter continues with a series of validation experiments on the policing mechanism implemented in the cell-rate simulator for the case of one On/Off modelled traffic source where priorities are not taken into account. The validation process uses a fluid flow model (see [Yin91]) for a single On/Off source (with exponentially distributed *On* and *Off* states) being policed by a Leaky Bucket mechanism. This is followed by the analysis of some traffic experiments carried out for each policing scenario (for traffic without priorities) with a view to suggesting the best actions to be taken by the UPC function on non-compliant traffic.

A similar approach (to that applied in the case of traffic without priorities) is followed to validate the policed cell-rate simulator for traffic with priorities. However, in this case, the approximate analysis used (and based in [Robe92, pp.150-152]) is valid for scenarios of multiple homogeneous On/Off modelled traffic sources. The validation process is complemented by a series of traffic experiments (using “mixtures” of traffic with priorities) to evaluate the performance of each policing scenario for traffic with priorities.

The Chapter ends with an evaluation of the processing speed reduction obtained with the policed cell-rate simulator for traffic with priorities, when compared with the simpler prioritised simulator. No comparison is made between the prioritised simulator and the policed simulator for traffic without priorities because the two scenarios for traffic without priorities can be viewed as particular cases of the policing scenarios for traffic with priorities. This can be seen by considering only high priority traffic in one of the policing scenarios for traffic with priorities.

7.1. Implementation of UPC in the Prioritised Cell-Rate Simulator

As mentioned in the beginning of this Chapter, the cell-rate simulator LINKSIM described in Chapter 5 has been used by the author to implement the UPC function by using the Leaky Bucket algorithm. This represents a new approach by the author to the study of policing algorithms since so far (and to the author's knowledge), policing has only been considered at cell level (see for example [EXPL94d]).

This Section describes how a policing algorithm was implemented in a cell-rate simulator and it represents new work by the author. The Leaky Bucket mechanism was chosen to perform policing in the cell-rate simulator because it is considered to be simple and its performance has been thoroughly analysed by different authors (see for example [Yama95]). Besides, the Generic Cell Rate Algorithm (GCRA) proposed by ITU (see [ITU95b]) is very similar to the algorithm of a Leaky Bucket mechanism.

7.1.1. Leaky Bucket

The principle of the Leaky Bucket algorithm is very simple (see Section 3.6.2.1 for a more detailed description); it considers a token pool (or *leaky bucket*) that is filled as a consequence of the arrival of cells and is emptied at a constant rate (the *leak rate*); cells are discarded when the token pool is full. This is the policing algorithm that will be used throughout the next Sections to study UPC. Schematically, the algorithm can be written as the following sequence of steps,

1. decrement a counter C by 1 every T seconds down to 0 if there are no cell arrivals; $(0 \le C \le M)$
2. increment the counter C by 1 for each transmitted cell;
3. discard (or tag) cells when C=M;

where M represents the maximum burst size (also called *bucket limit*) that the source is allowed to present and the reciprocal of T is the peak rate of the traffic being observed (also called *leak rate*). The increment in the bucket size produced by a cell arrival is called a *splash* (see [EXPL94a]).

7.1.2. UPC Actions on Misbehaving Traffic

Section 3.6.2.3 considered the state-of-the-art in terms of telecommunications' standards for UPC aspects. In particular, the actions that policing should take on any violating traffic were described. These are *discarding* or *tagging* (as referred to in the beginning of this Chapter), according to the network operator's choice (see [ITU94] and [ITU95b]). In order to put into practice either of these options, two scenarios have been put forward by ITU. Each scenario consists of carrying out two compliance tests: one test on the high priority traffic (i.e., traffic with CLP=0) and another on the global traffic (i.e., the CLP=0+1 traffic). A detailed description of these tests is given in Section 3.6.2.3. Here, only a diagram is presented (see Fig.7.1) to summarise the sequence of steps considered in each scenario.

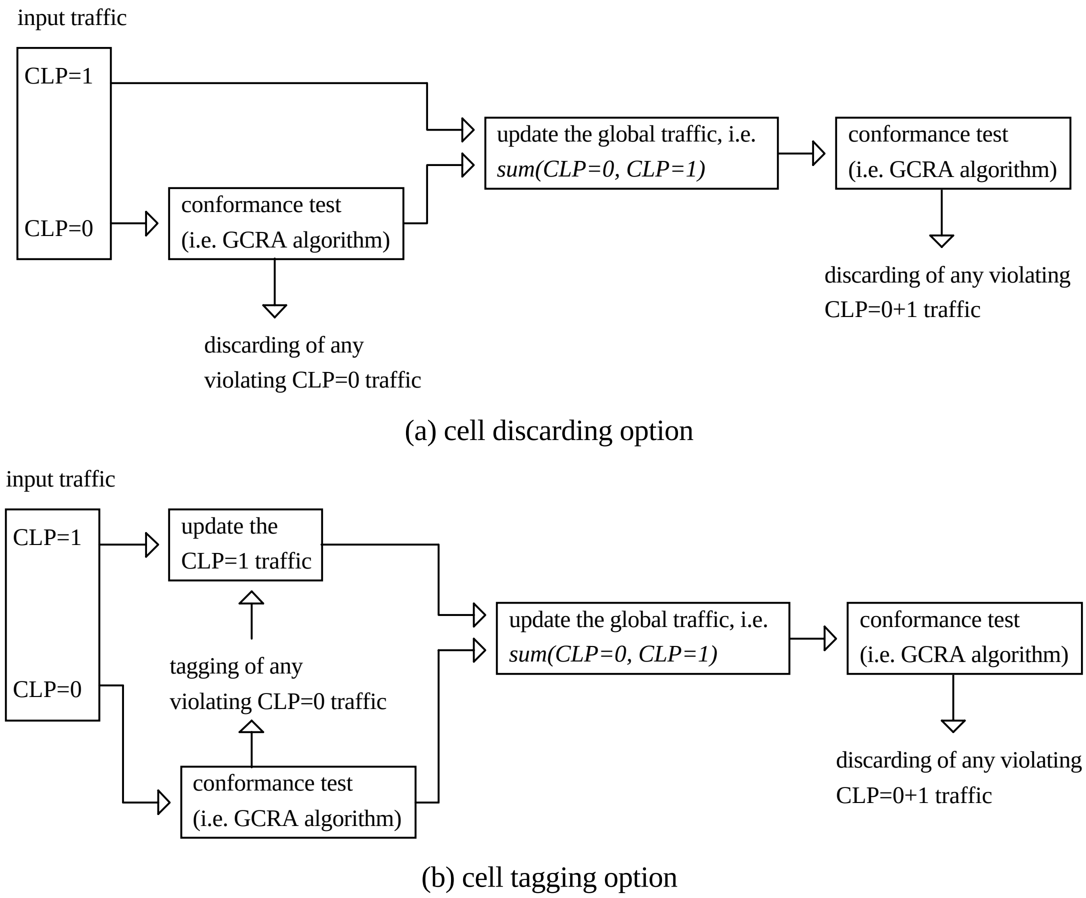


Fig.7.1 - UPC actions on misbehaving traffic (from [ITU95b]).

The implementation of the policing scenarios to be investigated in this thesis (see Section 7.1.3) will take into account the two compliance tests referred to above. Moreover, the compliance tests will use (as a decision mechanism) the Leaky Bucket algorithm that is very similar to the GCRA algorithm.

7.1.3. Policing Scenarios Studied

Four policing scenarios were considered to investigate the best actions that the UPC should take when it encounters any traffic violating the traffic contract established at call set-up. Their aim is to consider the case when traffic with priorities is present in the network, as well as when no priorities are considered. Also, two different UPC actions are analysed: *discarding* of cells and *tagging* (or marking) of cells. Taking this into account, the following situations were considered:

Scenario 1: Neither the UPC function nor the user can make use of the CLP bit. So, traffic sources can only have high priority cells (i.e., traffic is considered not to have priorities) and violating cells are discarded by the UPC mechanism.

Scenario 2: In this situation, only the UPC function can make use of the CLP bit. So, traffic sources can only have high priority cells (i.e., traffic is considered not to have priorities) and violating cells are tagged by the UPC mechanism.

Scenario 3: Here, only the user can make use of the CLP bit. The traffic sources can have low and high priority cells but violating high priority cells are discarded by the UPC mechanism in the first stage (according to Fig.7.1(a)). In the second stage, violating cells (be it of low or high priority) are discarded.

Scenario 4: In this case, both the UPC function and the user can make use of the CLP bit. The traffic can have cells of low and high priority and the UPC mechanism tags violating high priority cells in the first stage (see Fig.7.1(b)). Any violating cells are discarded in the second stage of the policing function.

Although *Scenario 2* does not consider traffic with priorities, the decision of tagging cells when they are found to be non-compliant implies the existence of some priority mechanism that can handle traffic with assigned priority levels. The reasoning applied to implement the policing scenarios in the prioritised cell-rate simulator is explained in the next Section.

In Section 7.1.2, the possible UPC actions on violating traffic (proposed by ITU) were indicated. It was seen that compliance tests are always carried out on both the *high priority traffic* (and all traffic is of high priority by default) and on the *global traffic* (which will be different from the high priority traffic if low priority traffic is considered). However, in the particular case of the first two policing scenarios defined above, only one test needs to be implemented. In fact, with *Scenario 1*, all the traffic is of high priority (since no low priority traffic is considered) and any violating traffic is discarded. This means that no test is necessary on the global traffic. The difference between *Scenarios 1* and *2* is that the latter tags all violating traffic. Therefore, the compliance test on the global traffic is still not necessary. The observation of these facts will imply a simplification in the implementation process.

In the case of *Scenarios 3* and *4*, two compliance tests must be performed, since traffic with priorities is considered.

7.1.4. Interpretation of Leaky Bucket in the Prioritised Cell-Rate Simulator

In order to understand how the several policing scenarios were implemented in the cell-rate simulator, it is worthwhile to briefly recall the simulator's mode of operation. LINKSIM works at burst level and the basic unit of traffic is a burst of cells, which means that during a certain time period, the inter-cell time remains constant. The events represent the time instants between bursts of cells with different cell rates (see Section 5.1).

On the other hand, the Leaky Bucket algorithm has been used (so far and to the author's knowledge) only at cell level and this is confirmed by its description in Section 7.1.1. Therefore, since the simulator works at burst level and the arrival of individual cells is no longer considered or analysed, a different reasoning must be applied to implement the policing algorithm. With the cell-rate simulator, *rates of cells* will be analysed.

The discussion of how the Leaky Bucket mechanism has been introduced in the prioritised cell-rate simulator is valid for both policing scenarios that consider traffic without priorities and policing scenarios for traffic with priorities. This is because each compliance test (for any given policing scenario) is a Leaky Bucket mechanism.

As mentioned before (see Section 5.1.1), each event in the cell-rate simulator represents a certain rate X of cells, which will last for a time period Y. At the beginning of each event, the rate X (at which traffic is being generated) is known. This is not necessarily true for the duration Y of the event, because the duration of the event will be predicted as a function of the input rate, the state of the queue (which is fed by the traffic source(s) considered) and the service rate of the queue (see Section 5.1.2 and [Pitt93]).

For the policing mechanism added to LINKSIM (and now referred to as Rate Based Leaky Bucket (RBLB)), VP policing is performed. This means that the judgement of whether traffic on a particular VP (that is being carried in one or more VCs) is respecting its contract or not is executed in the total traffic of that VP.

```
initialise system;
if (first event to process) then
    begin
        update the VC input rates for use in policing Test 0;
        for (each VP) do
            begin
                calculate the VP input rate for Test 0;
                if (VP input rate of Test 0 > VP peak cell rate of Test 0) then
                    begin
                        predict time of occurrence of bucket full event for Test 0;
                        insert time of occurrence of bucket full event for Test 0 in list of events;
                    end;
                update the generated VC input rates for policing Test 0;
            end;
    end
else while (there are events to process) do
    begin
        if (there is a system input rate change) then
            update the VC input rates for use in policing Test 0;
        for (each VP) do
            begin
                update the bucket level for Test 0;
                calculate the VP input rate for Test 0;
                if (VP input rate of Test 0 > VP peak cell rate of Test 0) then
                    begin
                        if (bucket level of Test 0 >=  bucket limit of Test 0) then
                            if (bucket full event for Test 0) then
                                process bucket full event for Test 0
                        else predict/insert in list of events, time of bucket full for Test 0;
                    end
                else reset discard rates for Test 0;
                if not (bucket full event for Test 0) then
                    update the generated VC input rates for policing Test 0;
            end;
    end;
```

Fig.7.2 - Policing algorithm for *Scenario 1*.

However, VC policing can also be studied by considering VPs that only contain one VC. Bearing this in mind, two parameters are important to execute policing on a given traffic: the total declared peak cell rate, which represents the *leak rate* (since a splash of one unit is assumed) and the *bucket's maximum size* (or bucket limit) for each VP in the system.

The simplest policing scenario is *Scenario 1* (see Section 7.1.3), where priorities are not taken into account and the UPC mechanism discards any violating traffic. In this case, the prioritised cell-rate simulator has been used, but the input traffic is specified as high priority traffic only. Because no low priority traffic enters the system at any point, no real use is made of the simulator's priority mechanism. Note that *Scenario 1* (and also *Scenario 2*, as seen later in this Section) only comprises one Leaky Bucket mechanism.

The implementation of the policing mechanism in the cell-rate simulator intends to show that it is possible to use burst level policing and this is validated in Section 7.2 (for the case of traffic without priorities) and in Section 7.4 (for policing traffic with priorities) by using analytical methods. The rate based policing algorithm for policing *Scenario 1* can be written as it is shown in Fig.7.2, where *Test 0* represents the policing mechanism applied to the input traffic (in this case, it is only high priority traffic). The initialisation of the system consists of giving initial values to variables such as the *bucket level* and the total *number of cells discarded* by the policing algorithm. The calculation of the input rate of each VP is simply the sum of the input rates for all the VCs belonging to a particular VP:

$$\mathrm{VP}(\mathrm{input})_{\mathrm{i}} = \sum_{\mathrm{j}=1}^{\mathrm{n_i}} \mathrm{VC}(\mathrm{input})_{\mathrm{j}} \ , \ \forall \mathrm{i} \tag{7.1}$$

where n_i represents the number of VCs that belong to the *ith* VP.

When the traffic being generated in a VP is greater than the maximum declared peak cell rate, there is a situation of violation of the traffic contract. In this situation, it is possible that, provided the same rate of cells is generated for a long period of time, the bucket will reach its limit. The time at which the case of a *bucket full* will occur can be predicted by the formula:

$$\mathrm{T(bucket\ full)_i} = \mathrm{T(current\ event)_i} + \frac{\mathrm{B(limit)_i - B(level)_i}}{\mathrm{VP(input)_i - VP(peak)_i}} \ , \ \forall \mathrm{i} \tag{7.2}$$

where for each VP, $T(bucket\ full)_i$ represents the time at which the bucket full event is predicted to occur, $T(current\ event)_i$ is the time at which the prediction is being performed, $B(limit)_i$ is the bucket size, $B(level)_i$ is the current value of the bucket, $VP(input)_i$ is the current input rate of the considered VP and $VP(peak)_i$ is the declared peak rate of that VP.

If the so called *bucket full event* occurs after the present event has terminated, then it will be ignored; it is an invalid event. The following example (see Fig.7.4) shows the difference between *valid* and *invalid bucket full events* for a Leaky Bucket mechanism. At the first event (named *event1*), that occurs at time *t1*, a bucket full event is predicted to happen at time *b_full1*. Similarly, the same happens at *event2*. However, the first bucket full event actually occurs after the termination time of *event1*. This means that, by the end of *event1*, the bucket did not yet reach its limit. So, although the traffic rate being generated was higher than the declared peak rate, no traffic was lost.

On the other hand, part of the traffic generated at *event2* is lost. This is because the bucket reaches its limit at time *b_full2* and traffic is still being generated from *event2*. The number of cells lost will be those corresponding to the area labelled *A* in Fig.7.4, that represents the product of the *excess rate* - defined in equation (7.4) - by the time period between *b_full2* and *t3*.

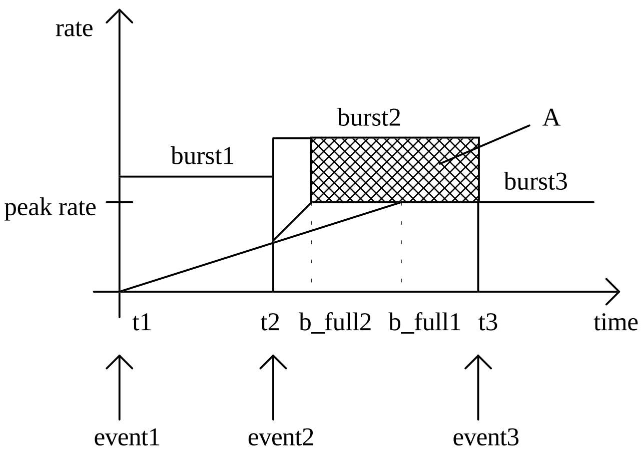


Fig.7.3 - Valid and invalid bucket full events (at *Test 0*).

After the calculation of the time instant at which the bucket full is predicted to occur, that event (or rather, the time at which it will occur) will be inserted in the queue of events in order to be processed in due time. The level of the bucket is updated for each VP at each new event by taking into account the time at which the last event occurred:

$$\mathrm{B(level)_i = B(level)_i -}$$
$$-\left[\mathrm{T(current\ event)_i - T(last\ event)_i}\right]\cdot\left[\mathrm{VP(peak)_i - VP(input)_i}\right]\ ,\ \forall \mathrm{i} \qquad (7.3)$$

where $T(last\ event)_i$ is the time at which the last event occurred.

When the event being analysed is a *valid bucket full event* (corresponding to a particular VP), then its processing consists of determining the *excess rate* (i.e., the difference between the VP input rate and the declared peak cell rate), followed by the evaluation of the discard rate of each VC belonging to that VP. Then,

$$\mathrm{VP(excess)_i = VP(input)_i - VP(peak)_i =}$$
$$= \sum_{\mathrm{j=1}}^{\mathrm{n_i}} \mathrm{VC(input)_j - VP(peak)_i}\ ,\ \forall \mathrm{i} \qquad (7.4)$$

and

$$\mathrm{VC(discard)_j} = \frac{\mathrm{VC(input)_j}}{\sum_{\mathrm{k=1}}^{\mathrm{n_i}} \mathrm{VC(input)_k}} \cdot \mathrm{VP(excess)_i}\ ,\quad \forall \mathrm{i};\ \forall \mathrm{j}:\ \mathrm{j} \in \{1,\dots,\mathrm{n_i}\} \qquad (7.5)$$

where $VP(excess)_i$ is the excess rate of the *ith* VP, $VC(input)_j$ is the input rate of the j*th* VC belonging to the *ith* VP and $VC(discard)_j$ represents the discard rate of each VC.

```
process bucket full event for Test0;
begin
    calculate the excess rates for the violating VP;
    update the generated VC input rates for Test 0;
    calculate the discard rates for the violating VP;
    update the VC input rates to the system's queue;
end;
```

Fig.7.4 - Algorithm for processing a bucket full event at *Test 0* (*Scenario 1*).

In other words, the excess rate of each VP is shared proportionally amongst all the VCs belonging to it. Fig.7.4 shows the algorithm for the processing of a valid bucket full event in *Scenario 1*.

As for the *number of cells discarded* per VC, its value is updated at every event by taking into account the time at which the last event occurred:

$$\mathrm{VC}(\mathrm{cellsdisc})_j = \mathrm{VC}(\mathrm{cellsdisc})_j + \mathrm{VC}(\mathrm{discard})_j \cdot$$
$$\cdot \sum_k \left[\mathrm{T}(\text{current event})_k - \mathrm{T}(\text{last event})_k \right],\ \forall j:\ j \in \{1,\ldots,n_i\} \qquad (7.6)$$

Scenario 2 considers traffic without priorities and the UPC function tags (or marks) violating traffic as low priority traffic. Similarly to the case of policing *Scenario 1*, this scenario also uses the prioritised cell-rate simulator. However, in the present case, the simulator's priority mechanism is used to process any tagged traffic by the UPC function. The rate based policing algorithm for *Scenario 2* can thus be written similarly to that of *Scenario 1*. However, every step previously related to the calculation of the number of cells discarded (see equations (7.4) to (7.6)) now refers to the number of cells tagged. This implies an update in the low priority input rates when traffic is found to be violating its traffic contract. Note that the low priority input rates were initially null because the user cannot make use of the CLP bit in this policing scenario.

The implementation of policing *Scenario 2* is similar to that of *Scenario 1*. The difference lies in the processing of the bucket full events. In the case of *Scenario 2*, the excess rate of each violating VP still has to be determined. However, no cells are discarded; instead, the excess rate is shared proportionally amongst the VCs of that particular VP and the rate tagged for each VC is calculated (this tagged traffic thus becomes low priority traffic). The number of cells tagged must also be updated at each new event, similarly to what is described in equation (7.6) for the number of cells discarded.

With policing *Scenarios 3* and *4*, traffic with priorities is considered and two policing tests must be performed, according to Fig.7.1. Fig.7.5 shows the main steps needed to be carried out in the execution of the two policing tests for traffic with priorities, when it is assumed that the test on the high priority traffic discards violating traffic (i.e., in the case of policing *Scenario 3*).

In Fig.7.5, *Test 0* represents the policing test on the high priority traffic and *Test 0+1* is the policing test on the global traffic. Each of the policing tests executed in the scenarios for traffic with priorities has the same main structure as the test presented in Fig.7.2. Therefore, the calculations described in equations (7.1) to (7.6) still apply.

```
initialise system;
if (first event to process) then
    begin
        update the VC input rates for use in policing Test 0 / Test 0+1;
        for (each VP) do
            begin
                calculate the VP input rate for Test 0 / Test 0+1;
                if (VP input rate of Test 0 / Test 0+1 > VP peak cell rate of Test 0 / Test 0+1) then
                    begin
                        predict time of occurrence of bucket full event for Test 0 / Test 0+1;
                        insert time of occurrence of bucket full event for Test 0 / Test 0+1 in list of events;
                    end;
                update the generated VC input rates for policing Test 0 / Test 0+1;
            end;
    end
else while (there are events to process) do
    begin
        if (there is a system input rate change) then
            update the VC input rates for use in policing Test 0 / Test 0+1;
        for (each VP) do
            begin
                update the bucket level for Test 0 / Test 0+1;
                calculate the VP input rate for Test 0 / Test 0+1;
                if (VP input rate of Test 0+1 > VP peak cell rate of Test 0+1) then
                    begin
                        if (bucket level of Test 0+1 >=  bucket limit of Test 0+1) then
                            if (bucket full event for Test 0+1) then
                                process bucket full event for Test 0+1
                        else predict/insert in list of events, time of bucket full for Test 0+1;
                    end
                else reset discard rates for Test 0+1;
                if (VP input rate of Test 0 > VP peak cell rate of Test 0) then
                    begin
                        if (bucket level of Test 0 >=  bucket limit of Test 0) then
                            if (bucket full event for Test 0) then
                                process bucket full event for Test 0
                        else predict/insert in list of events, time of bucket full for Test 0;
                    end
                else reset discard rates for Test 0;
                if not (bucket full event for Test 0) and not (bucket full event for Test 0+1) then
                    update the generated VC input rates for policing Test 0 / Test 0+1;
            end;
    end;
```

Fig.7.5 - Policing algorithm for *Scenario 3*.

However, it is important to notice that the two compliance tests (one on the high priority traffic and another on the global traffic) are not independent. This is because the output of the test on the high priority traffic affects the way in which the second test will monitor the global traffic. That phenomenon can be seen in the sequence of steps executed for each policing test when a valid bucket full event is found (see Fig.7.6). It is also worthwhile mentioning that, since the policing scenarios for traffic with priorities consider two Leaky Bucket mechanisms, characterising parameters need to be defined for each policing test (as seen in the beginning of this Section for policing *Scenarios 1* and *2*).

```
process bucket full event for Test0;
begin
    calculate the excess rates for the violating VP;
    update the generated VC input rates for Test 0;
    calculate the discard rates for the violating VP;
    update the VC input rates to the system's queue;
    update the generated VC input rates for Test 0+1;
    update the VP input rates for policing Test 0+1;
    if (VP input rate of Test 0+1 > VP peak cell rate of Test 0+1) then
        begin
            predict time of occurrence of bucket full event for Test 0+1;
            insert time of occurrence of bucket full event for Test 0+1 in list of events;
        end
    else reset discard rates for Test 0+1;
end;

process bucket full event for Test 0+1;
begin
    calculate the excess rates for the violating VP;
    update the generated VC input rates for Test 0 / Test 0+1;
    calculate the discard rates for the violating VP;
    update the VC input rates to the system's queue;
end;
```

Fig.7.6 - Algorithm for processing a bucket full event at *Test 0* / *Test 0+1* (*Scenario 3*).

In the case of a *Test 0* bucket full event, not only is it necessary to calculate the excess and discard rates for each of the VCs in the violating VP (as given by equations (7.4) and (7.5)), but also, it must be checked whether a new bucket full event for *Test 0+1* needs to be predicted. Fig.7.7 illustrates an example where, after a bucket full event is found for *Test 0*, a new bucket full event has to be predicted for *Test 0+1*. At time *t1*, a new rate of traffic is generated (for a given VP) and bucket full events are predicted for the test on the high priority traffic (i.e., at *Test 0*) and for the test on the global traffic (i.e., at *Test 0+1*). A bucket full event for *Test 0* occurs at time *b_full(0)*, which is earlier than the corresponding bucket full event for *Test 0+1* (this would occur at time *b_full(0+1)*). The processing of the bucket full for *Test 0* at time *b_full(0)* will imply the calculation of excess and discard high priority rates (as given by equations (7.4) and (7.5)). Therefore, from time *b_full(0)* to time *t2* (and if no other events occur), a portion of the high priority rate (i.e., the area labelled *A* in Fig.7.7) will be discarded that should not be accounted for by *Test 0+1*. In view of this, the global input rate should be recalculated at time *b_full(0)* and a new bucket full event for *Test 0+1* must be predicted, in the case of the new global input rate still exceeding the declared global peak cell rate (as was the case at time *t1*).

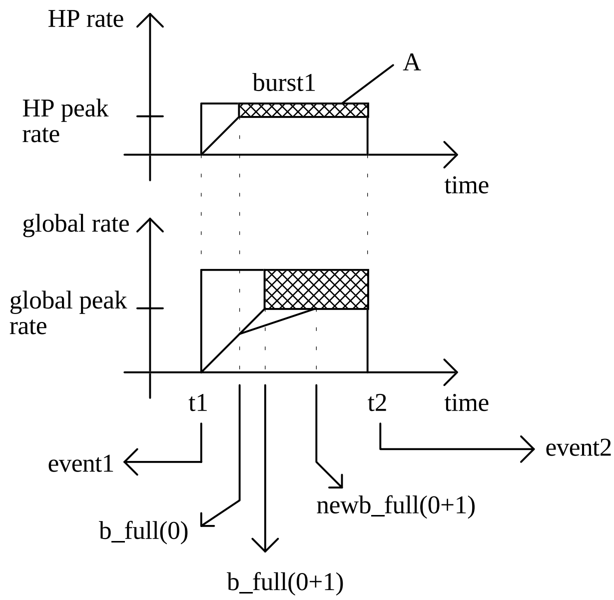


Fig.7.7 - The output of *Test 0* and its influence in *Test 0+1*.

The implementation *of Scenario 4* is analogous to that of *Scenario 3,* except that the high priority test now tags violating high priority traffic (instead of discarding it, as was the case with *Scenario 3*). This means that the rate based policing algorithm given in Fig.7.5 for *Scenario 3* (and in Fig.7.6 for the processing of bucket full events at policing *Test 0* and *Test 0+1*) will be identical in the case of *Scenario 4*. However, equations (7.5) and (7.6) are now used, not only for the calculation of the VCs' discarding rates and number of cells discarded (in policing *Test 0+1*), as well as for the tagged rates and number of cells tagged (in policing *Test 0*). Since the policing *Test 0* now tags violating high priority traffic, it is also necessary to update the low priority traffic rates when there is a bucket full event at *Test 0*. This is done by adding to each VC (within the same violating VP) the proportion of high priority excess rate (i.e., the tagged rate) detected by the policing *Test 0*.

7.2. Validation of the Policed LINKSIM for Traffic without Priorities

In Section 7.1.3, two scenarios were described under which traffic without priorities can be policed. One of policing scenarios, *Scenario 1*, considered the situation where the traffic being policed cannot make use of the CLP bit (i.e., the traffic has no assigned priority levels) and the policing mechanism discards any traffic found to be violating the traffic contract.

Here, experiments for that policing scenario have been carried out and compared with a fluid flow model described in [Yin91] for a single On/Off modelled traffic source with exponentially distributed *On* and *Off* state durations. The model provides closed-form expressions that relate the Leaky Bucket parameters (i.e., the *bucket size* and the *leak rate*) and the source characteristics to the cell discarding probability. It also gives closed-form expressions for the cell tagging probability (or ratio) of the source and it is valid for both the case of an unbuffered Leaky Bucket and a buffered Leaky Bucket. The work developed in this thesis only addresses the former situation.

This Section also includes some traffic experiments that consider the policing *Scenario 2*, where any violating traffic is tagged. Only cell-rate simulation is used to carry out the traffic experiments with this policing scenario, since there is yet no analytical method (to the author’s knowledge) that caters for both policing and prioritised traffic. However, the implementation of this scenario can be validated by using again the already mentioned analysis. This is possible since the fluid flow approximation method provides closed-form expressions for the cell tagging ratio.

7.2.1. The Fluid Flow Approximation

The concept of *fluid flow analysis* has already been described in Section 6.1. It refers to an approach whereby the traffic generated by sources is analysed at burst level and treated as a continuous stream of cells, thus ignoring the inherently slotted nature of traffic in ATM networks.

For the validation process of the policed cell-rate simulator, a fluid flow model introduced in [Yin91] has been used. The model analyses the Leaky Bucket algorithm for an On/Off modelled traffic source (with exponentially distributed *On* and *Off* state durations) that is characterised by its *average* and *peak rates*, as well as by its *average burst length*. The Leaky Bucket algorithm uses a fictitious queue to model the behaviour of the mechanism and the analysis can be applied both in the case of buffered and unbuffered Leaky Buckets. In this environment, the analysis described by Yin (see [Yin91]) then gives closed-form expressions for the cell discarding/marking probability and queuing delay (in the case of a buffered Leaky Bucket). The description of the fluid flow analysis given here merely indicates the formulas developed in [Yin91] that are used in this thesis.

The following definitions need to be taken into account in the calculation of the cell discarding/marking probability:

pbr	=	source peak bit rate
scr	=	permit generation rate or leak rate of the Leaky Bucket (also called sustainable cell rate)
max_b	=	maximum bucket size
abl	=	average burst length (in bits)
mbr	=	average (or mean) source bit rate
e_prob	=	equilibrium probability in the *On* state (i.e., the source utilisation)
lb_load	=	Leaky Bucket load.

The *Leaky Bucket load* and the *equilibrium probability in the On state* can be written as,

$$\text{lb\_load} = \frac{\text{mbr}}{\text{scr}} \qquad \text{and} \qquad \text{e\_prob} = \frac{\text{mbr}}{\text{pbr}}. \tag{7.7}$$

[Yin91] then derives an expression for the probability of the fictitious queue being full by considering the On/Off source modelled by a two-state Markov chain, as shown in Fig.7.8. Therefore, the analysis only considers the situation where the Leaky Bucket load (given by *lb_load*) is greater than the source utilisation (given by *e_prob*). In the opposite case, the Leaky Bucket "queue" would always remain empty and no traffic would be discarded or tagged.

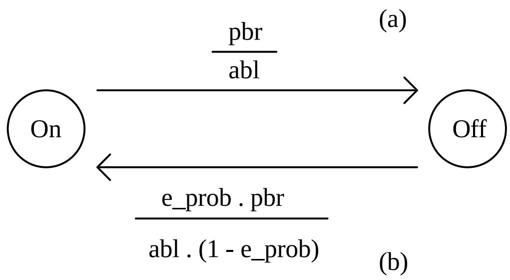


Fig.7.8 - The two-state Markov modulated rate process for the On/Off source;
(a) and (b) are the transition rates.

The *cell discarding probability* (and similarly for the case of the cell marking probability) is a function of both the *probability of the fictitious queue being full* and the *rate at which traffic is discarded* when the fictitious queue is full; in mathematical notation,

$$\begin{aligned} P_{\text{discard}} &= \frac{\text{pbr} - \text{scr}}{\text{mbr}} \cdot \text{Prob}\{\textit{fictitious queue being full}\} \\ &= \frac{(\text{lb\_load} - \text{e\_prob}) \cdot (1 - \text{e\_prob}) \cdot e^{\varepsilon\,\text{max\_b}}}{\text{lb\_load} \cdot (1 - \text{e\_prob}) \cdot \left(1 - \dfrac{\text{lb\_load} - \text{e\_prob}}{1 - \text{e\_prob}} \cdot e^{\varepsilon\,\text{max\_b}}\right)} \end{aligned} \qquad (7.8)$$

where ε is the non-zero system eigenvalue of the matrix that specifies the differential equations for the system (see [Yin91]) and it is given by the expression,

$$\varepsilon = \frac{\text{lb\_load} \cdot (\text{lb\_load} - 1)}{\text{abl} \cdot (1 - \text{e\_prob}) \cdot (\text{lb\_load} - \text{e\_prob})}, \text{ for } \text{lb\_load} < 1. \qquad (7.9)$$

7.2.2. Policed Cell-Rate Simulation *versus* Fluid Flow Approximation: Discarding Violating Traffic

In this Section, some traffic experiments have been carried out with the policed cell-rate simulator for the particular case of one On/Off source with exponentially distributed *On* and *Off* state durations. The cell discarding probability results thus obtained were then compared with those obtained by using the fluid flow approximation described in Section 7.2.1. It is worth noticing that similar experiments were also carried out in an ATM testbed (see [EXPL94d]) and compared with the same fluid flow model. The outcome of comparing the two techniques (i.e., cell-rate simulation and fluid flow analysis) will help evaluate the cell-rate simulator's accuracy in traffic studies that involve the policing of traffic.

Each set of experiments considers one On/Off modelled traffic source being policed by a function that uses the Leaky Bucket mechanism to take action (in this case, by discarding cells) when the traffic being generated by the source is violating its traffic contract. Although the sources used have been modelled as On/Off (with the *On* and *Off* state durations exponentially distributed), the parameters were varied so as to analyse three different situations:

1. traffic with high burstiness;
2. traffic with high peak cell rate;
3. traffic with both low peak cell rate and low burstiness.

The experiments concentrate on policing the two traffic parameters *peak cell rate* (given by *pbr* in Section 7.2.1) and *sustainable cell rate* (given by *scr* in Section 7.2.1), which have previously been defined in Section 3.6.2.3. The sustainable cell rate is defined for these experiments as a function of the two parameters *mean cell rate* (given by *mbr* in Section 7.2.1) and *peak cell rate*; the sustainable cell rate was chosen to take the following values:

$$\frac{1}{\mathrm{scr}} = \begin{cases} \dfrac{1}{\mathrm{mbr}} + \dfrac{1}{\mathrm{n}} \cdot \left(\dfrac{1}{\mathrm{pbr}} - \dfrac{1}{\mathrm{mbr}} \right); & \mathrm{n} = 2,4,8 \\ \dfrac{2}{\mathrm{pbr}} \end{cases} \tag{7.10}$$

The policing mechanism takes two parameters: the *leak rate* (corresponding to the sustainable cell rate, in this case) at which the bucket is emptied and the *bucket limit* (given by *max_b*, as in the description of the fluid flow approximation in Section 7.2.1). This last parameter is chosen to be given as a function of the *maximum burst size* (mbs), the *sustainable cell rate* and the *peak cell rate*:

$$\mathrm{max\_b} = \mathrm{mbs} \cdot \left(1 - \frac{\mathrm{scr}}{\mathrm{pbr}} \right) \tag{7.11}$$

The *mbs* parameter represents the maximum burst tolerance (or cell delay variation tolerance) that the traffic source is allowed to present. It is therefore a measure of the burstiness allowed into the network, since it indicates the maximum number of cells that can pass the UPC function at the peak cell rate without the occurrence of cell discards. The maximum burst size is chosen to be calculated as a function of the average burst size (given by *abl* in the description of the analytical model in

Section 7.2.1) of the source:

$$\text{mbs} = x \cdot \text{abl} \ ; \quad x = 1,2,5,10 \tag{7.12}$$

where *x* is a tolerance factor in the calculation of the maximum burst size. With these parameters and their possible values, the *cell discarding ratio* (cdr) is obtained for each configuration as a function of the considered sustainable cell rate and maximum burst tolerance. The cell discarding ratio calculated by the policed cell-rate simulator is given by the quotient,

$$\text{cdr} = \frac{\text{discarded cells}}{\text{passed cells} + \text{discarded cells}} \ . \tag{7.12}$$

The first set of experiments considers one traffic source of *type A* (described in Table 7.1), which has a high peak bit rate (or equivalently, high peak cell rate) of 31.1 Mbit/s, being policed at four different rates that are given by equation (7.10).

Type A pbr = 31.1 Mbit/s		***Type B*** pbr = 7.78 Mbit/s		***Type C*** pbr = 1.94 Mbit/s	
state	duration	state	duration	state	duration
On	20 ms	On	10 ms	On	50 ms
Off	80 ms	Off	190 ms	Off	50 ms

Table 7.1 - Sources' traffic parameters for experiments with policing.

For each of the considered policing rates, a series of four bucket limits is then calculated, using equations (7.11) and (7.12).

Fig.7.9 shows the values for the cell discarding ratio obtained from the fluid flow analysis (given by lines labelled as *cdr_analysis*) and the policed cell-rate simulator (in the lines labelled as *cdr_simulation*), when varying the sustainable cell rate (i.e., the policing rate) and maintaining the value of the ratio given in equation (7.11). The graph (see Fig.7.9) shows that there is a good agreement between the results obtained with the policed cell-rate simulator and the fluid flow analysis. No confidence intervals are shown for the simulated results as they are very close to the analytical ones. It can be seen that, as the policing rate approaches the peak cell rate of the source (for any given maximum burst tolerance), the cell discarding ratio tends to decrease. This reduction in the cell discarding ratio becomes more noticeable when the maximum burst tolerance given to the source is also high. This is explained by the fact that, as the sustainable cell rate is increased for a given burst tolerance, more traffic will be allowed to pass the policing mechanism, therefore reducing the cell discarding ratio.

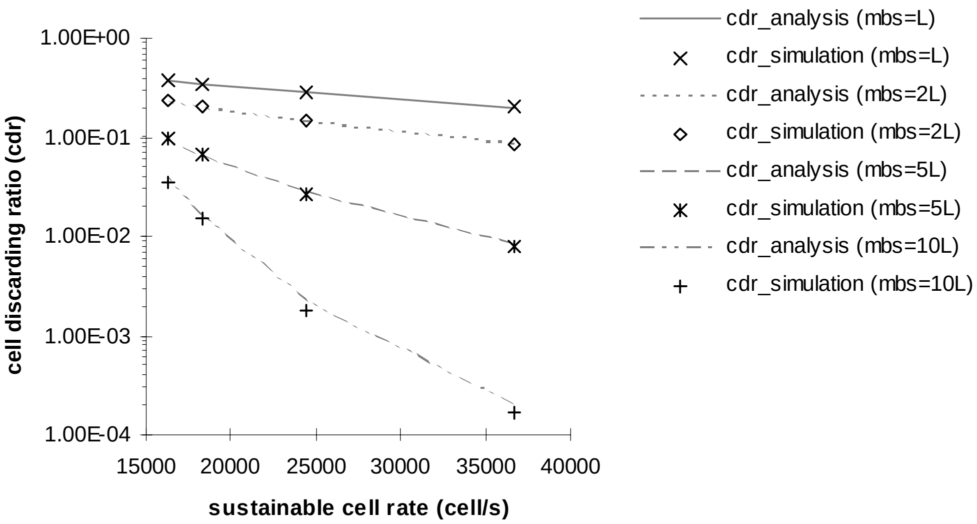


Fig.7.9 - Policed simulation *versus* fluid flow approximation: traffic with high peak cell rate (*source type A*). (*) L=abl

The next set of experiments considers the policing of a *type B* traffic source (described in Table 7.1), which has a burstiness of 20 (where the burstiness has been taken to be the peak-to-mean ratio - see Section 3.3). A similar approach to that used in Fig.7.9 has been applied to this traffic type when presenting the cell discarding ratios obtained from both techniques. The comparison results can be found in Fig.7.10.

The similarity of the obtained results shown in both this graph (Fig.7.10) and the previous one (i.e., Fig.7.9) is obvious. It appears that the policed cell-rate simulator is fairly accurate when compared with an analytical model in the situation of the traffic involved being either very bursty or requiring a large bandwidth. Notice however that no simulated value is given for the case in which both the highest sustainable cell rate and the highest burst tolerance are considered. This is because the policed cell-rate simulator produced a null value for the cell discarding ratio.

The last set of experiments with policed traffic used a *type C* traffic source. Again, different policing rates (given by equation (7.10)) were used and for each policing rate, the bucket limit was varied. The results are shown in Fig.7.11.

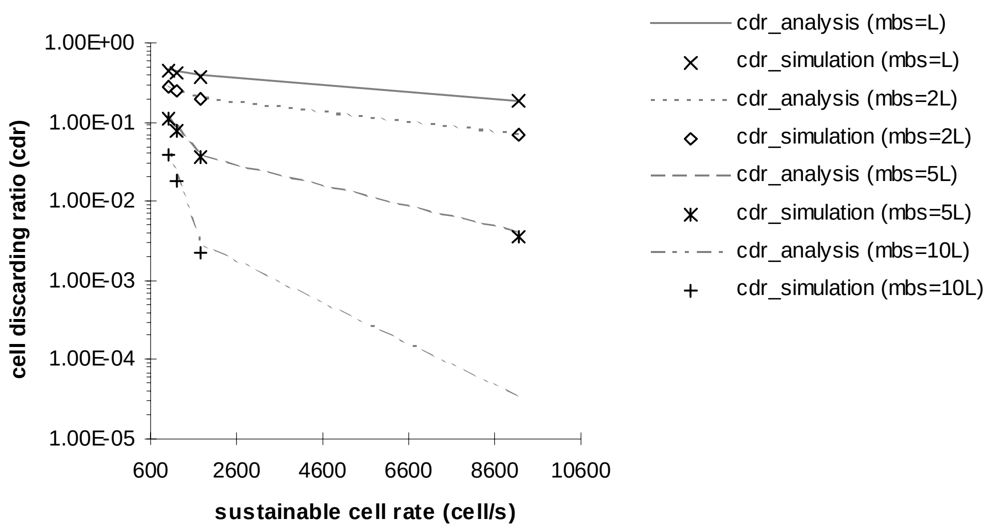


Fig.7.10 - Policed simulation *versus* fluid flow approximation: traffic with high burstiness (*source type B*). (*) L=abl

Again, the two techniques (cell-rate simulation and fluid flow analysis) produce almost coincident results for the cell discarding ratio of the *type C* traffic source. Also, the trend in the discarding ratios obtained for given policing rates and maximum burst tolerances follows a similar pattern to that of Fig.7.9.

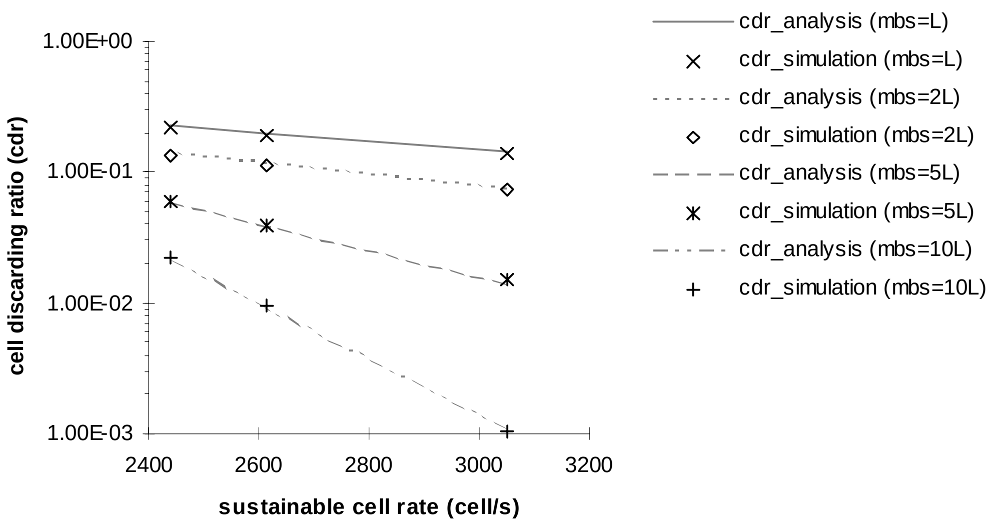


Fig.7.11 - Policed simulation *versus* fluid flow approximation: traffic with low peak cell rate and low burstiness (*source type C*). (*) L=abl

As a summary of what was verified with the sets of traffic experiments in this Section, it can be said that the cell-rate simulator with policing (for traffic without priorities and discarding of violating traffic) is fairly accurate, not only when the traffic has a high peak rate or is very bursty, but also when the source has both a low peak rate and a low burstiness. This validates the RBLB mechanism implemented in the prioritised cell-rate simulator for the case of *Scenario 1* (see Section 7.1.3).

7.2.3. Policed Cell-Rate Simulation *versus* Fluid Flow Approximation: Tagging Violating Traffic

The second policing scenario (i.e., *Scenario 2*) implemented in the prioritised cell-rate simulator (see Section 7.1.3) considers tagging as the action to be taken on any traffic found to be non-compliant. The implementation of this scenario was validated with the fluid flow analysis by Yin (see [Yin91]) that has been briefly described in Section 7.2.1. Yin's fluid flow analysis gives a closed-form formula to calculate the cell tagging probability of an On/Off modulated traffic source with *On* and *Off* exponentially distributed states when policed according to a Leaky Bucket mechanism.

In order to validate the RBLB mechanism implemented in the prioritised cell-rate simulator for *Scenario 2*, two main sets of experiments were carried out with the aim of testing the policed cell-rate simulator in diverse situations. All experiments consider the particular case of a single On/Off traffic source being policed and three types of sources have been considered, according to their burstiness. The cell tagging probability results obtained from the simulation experiments are compared with those obtained with Yin's fluid flow analysis (see [Yin91]).

Before giving a description of the two main sets of experiments, it is important to introduce two concepts that will be used in the following. One of the concepts is the *excess burst size* (ebs). It represents the part of the *average burst size* (abs) of a source that exceeds the declared rate of that source (see Fig.7.12). If all the traffic that represents an excess in the declared peak rate is discarded/marked (i.e., when the bucket size limit of the policing function has a null size), then the source will "lose" as many cells as there are in the excess burst size. In most cases however, the bucket size limit of the policing function has a non-null value, so as to cater for the effects of CDV in the traffic (see Section 3.6.2.2). In this case, a proportion of the excess burst size will be passed by the UPC (see Fig.7.12); that proportion of the excess burst size is called the *excess burst passed* (ebp).

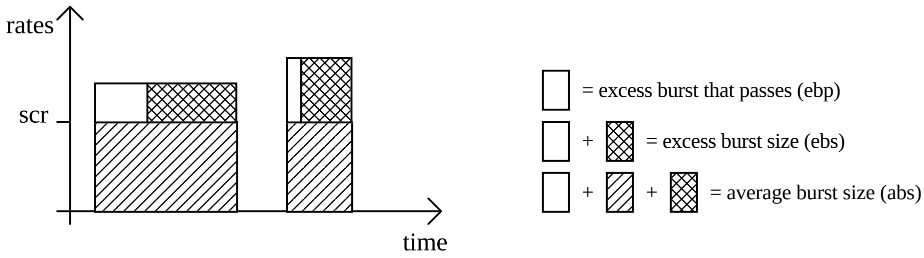


Fig.7.12 - Identification of traffic parameters for policing experiments (traffic without priorities).

The first set of experiments considers three different sources with the same *mean bit rate* (mbr) and average burst size and where the following ratio is considered to be a constant for all the sources (see Fig.7.12):

$$\frac{\mathrm{ebp_i}}{\mathrm{abs_i}} = \text{constant}, \ \forall \mathrm{i}\,. \tag{7.13}$$

This means that the excess burst size will also be constant for all three types of sources. Taking this into account, the time cycle (i.e., the sum of the average durations of the *On* and *Off* states) of the sources can be calculated by using the expression,

$$\mathrm{mbr_i} = \mathrm{mbr} = \frac{\mathrm{abs_i}}{\mathrm{Tcycle_i}}, \ \ \mathrm{Tcycle_i} = \mathrm{Ton_i} + \mathrm{Toff_i}, \ \ \forall \mathrm{i} \tag{7.14}$$

where $Tcycle_i$ represents the time cycle of the *ith* source type. Having the values of the mean and peak bit rates for each source, it is possible to calculate the duration of the *On* and *Off* states by using the expression,

$$\mathrm{mbr_i} = \frac{\mathrm{On_i}}{\mathrm{On_i} + \mathrm{Off_i}} \cdot \mathrm{pbr_i}, \ \ \forall \mathrm{i} \tag{7.15}$$

The value of the excess burst size can then be determined by the formula,

$$\mathrm{ebs_i} = \mathrm{On_i} \cdot \left(\mathrm{pbr_i} - \mathrm{scr_i}\right), \ \ \forall \mathrm{i} \tag{7.16}$$

Adding up to this setting, the experiments also consider variable degrees of traffic violation for each source type; this enables the calculation of the leak rate - or sustainable cell rate (scr), as defined in Section 7.2.2 - of the UPC. With this information, the value of the bucket size can be deduced for a given traffic violation by using the equation,

$$\text{bucket\_size}_i = \text{rate\_duration}_i \cdot (\text{pbr}_i - \text{scr}_i), \quad \forall i \tag{7.17}$$

where $rate\_duration_i$ corresponds to the time that is necessary for the source to generate the total number of cells that are passed in an average burst. It is given by,

$$\text{rate\_duration}_i = \frac{\text{On}_i \cdot \text{total\_cellspassed}_i}{\text{abs}_i} \tag{7.18}$$

$$= \frac{\text{On}_i \cdot \left[\text{ebp}_i + (\text{abs}_i - \text{ebs}_i)\right]}{\text{abs}_i}, \quad \forall i \tag{7.19}$$

In summary, the common parameters for all the first set experiments are as indicated by Table 7.2 and Table 7.3 gives each source's parameters.

Common parameters for policing with tagging (experiments' set 1)			
mean bit rate (mbr)	average burst size (abs)	cycle time	excess burst passed (ebp)
2 Mbit/s	400 cells	84.8 ms	10 cells

Table 7.2 - Tagging traffic without priorities: common traffic parameters to all sources and set 1 experiments.

Therefore, the variable parameters for this set of experiments are the *sustainable cell rate*, the *excess burst size* and the *bucket size*. These parameters depend on the traffic violation (which is taken to vary from 5% to 25%, with increments of 5%) considered for a particular source type.

	Source X1	***Source Y1***	***Source Z1***
peak bit rate (pbr)	4.5 Mbit/s	10 Mbit/s	22 Mbit/s
burstiness	2.25	5	11
On time	37.7 ms	16.96 ms	7.7 ms
Off time	47.1 ms	67.84 ms	77.1 ms

Table 7.3 - Tagging traffic without priorities: traffic parameters for low, medium and high burstiness sources (set 1 experiments).

Using the input data from the two previous Tables (i.e., Table 7.2 and Table 7.3) and executing some calculations to determine the UPC parameters with equations

(7.14) to (7.19), simulation results were obtained for the cell tagging probability. In Fig.7.13, results are shown for the cell tagging probability obtained with the fluid flow analysis described in Section 7.2.1 by using equation (7.8). The same figure (i.e., Fig.7.13) also gives the difference (in terms of percentage) between the simulated and analytical results for the source types indicated in Table 7.3. The graphs in Fig.7.13 show that the policed cell-rate simulator produces results for the cell tagging probability that are almost coincident with those obtained by using Yin's analysis. Moreover, this observation is true both for diverse levels of traffic violation and for traffic sources with different burstiness measures.

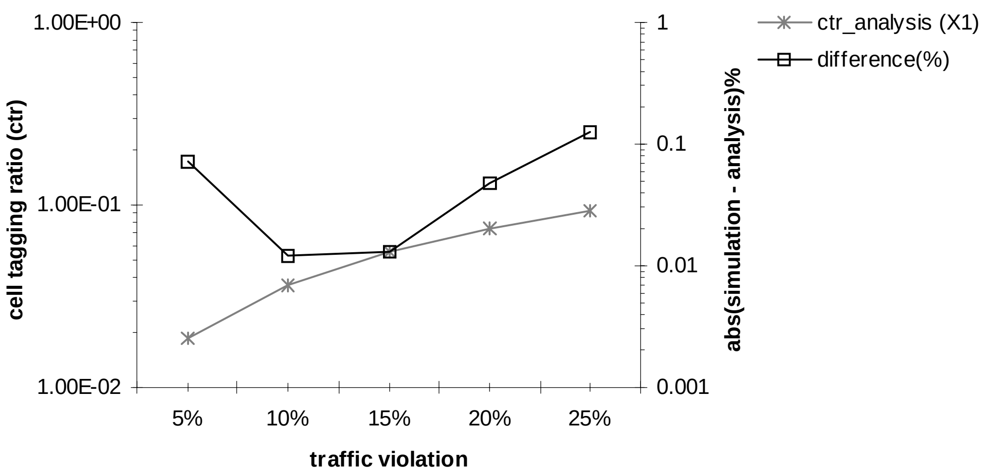
The second set of experiments again considered three different traffic sources that have the same peak and mean bit rates as those described for the first set of experiments in Table 7.3. However, the main characteristic of this second set of experiments is that the excess burst size remains constant for all types of sources used (see Fig.7.12). Therefore, the following relationship is kept constant for all experiments:

$$\frac{\mathrm{ebp_i}}{\mathrm{ebs_i}} = \text{constant}, \ \forall \mathrm{i}\,. \tag{7.20}$$

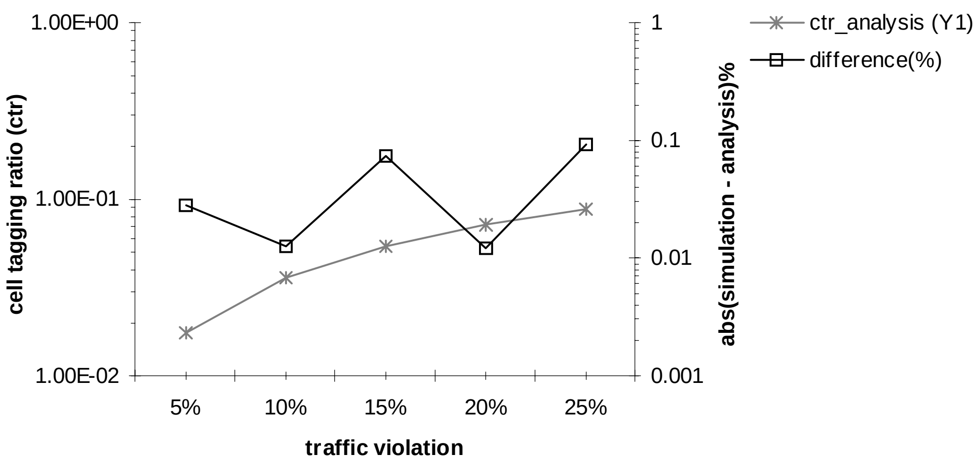
Once again, the mean bit rate is also kept constant for the three types of sources. However, in this case, equation (7.14) does not imply that the average burst size and cycle time of the sources will also remain fix. Therefore, although formulas (7.14) to (7.19) are still used, the common parameters for this second set of experiments (given in Table 7.4) are now different from those indicated in Table 7.2. Therefore, the variable parameters for this set of experiments are the *sustainable cell rate*, the *bucket size*, the *cycle time* and the *On* and *Off* state durations. These parameters depend on the traffic violation (which is taken to vary from 5% to 25%, with increments of 5%) considered for a particular source type.

Common parameters for policing with tagging (experiments' set 2)		
mean bit rate (mbr)	excess burst size (ebs)	excess burst passed (ebp)
2 Mbit/s	40 cells	10 cells

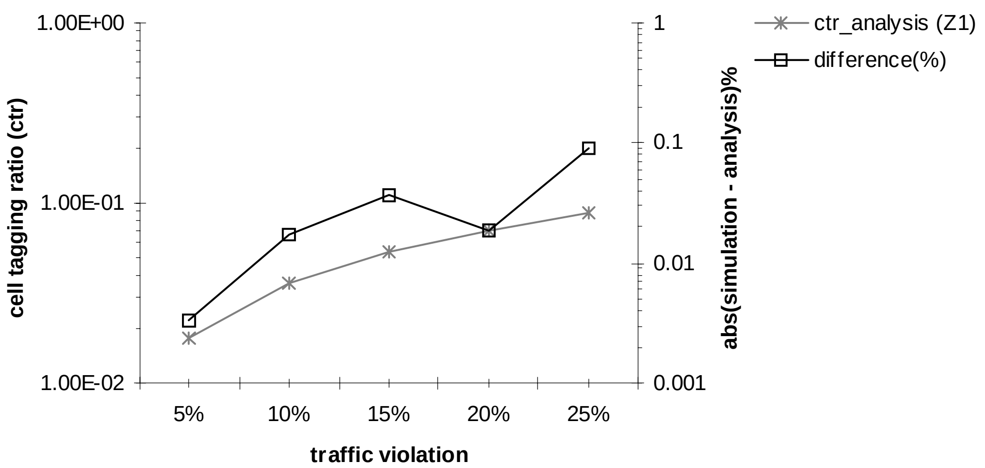
Table 7.4 - Tagging traffic without priorities: common traffic parameters to all sources and set 2 experiments.



(a) source type *X1* - low burstiness



(b) source type *Y1* - medium burstiness



(c) source type *Z1* - high burstiness

Fig.7.13 - Policed simulation *versus* fluid flow analysis: cell tagging probability (set 1 experiments - fix average burst size).

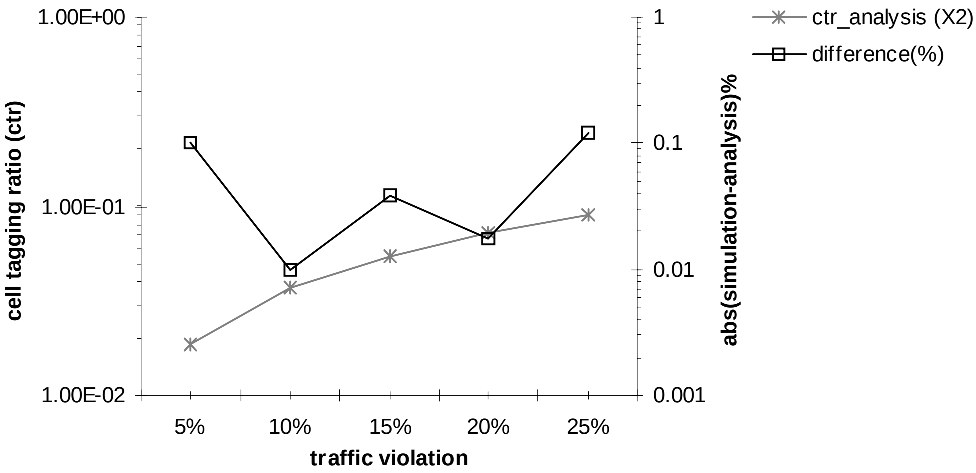
	Source X2	***Source Y2***	***Source Z2***
peak bit rate (pbr)	4.5 Mbit/s	10 Mbit/s	22 Mbit/s
burstiness	2.25	5	11

Table 7.5 - Tagging traffic without priorities: traffic parameters for low, medium and high burstiness sources (set 2 experiments).

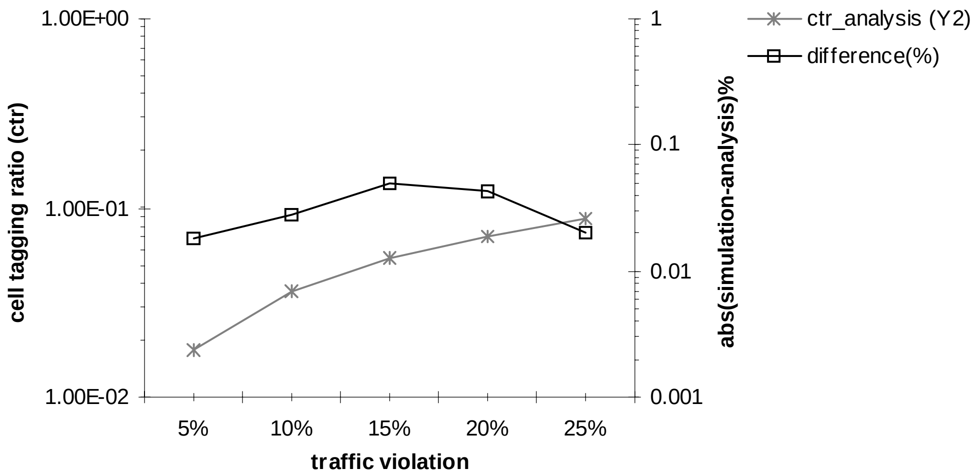
With the data from Table 7.4 and Table 7.5 and executing some calculations to determine the UPC parameters with equations (7.14) to (7.19), simulation results were obtained for the cell tagging probability. Fig.7.14 gives the cell tagging probability results obtained with the fluid flow analysis and it presents the difference (in terms of percentage) between the simulated and analytical results for the source types indicated in Table 7.5.

The analysis of the graphs in Fig.7.14 indicates that the policed cell-rate simulator also gives accurate results for the cell tagging probability under the settings of this second set of experiments. Again, the degree of accuracy in the results does not depend on the burstiness of the sources nor on the level of traffic violation of a particular source.

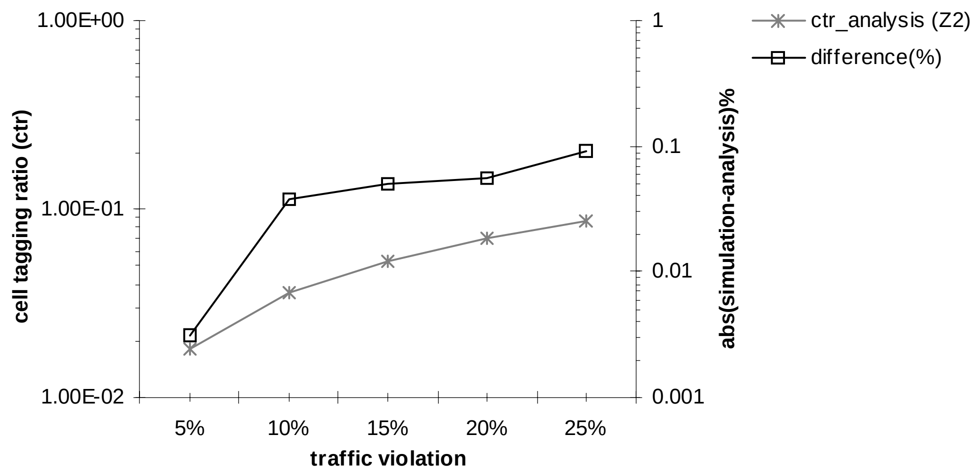
In summary, the policing *Scenario 2* implemented in the prioritised cell-rate simulator was exposed to the two sets of experiments described in this Section. The experiments were intended to represent diverse situations to investigate the accuracy of the policed simulator when compared with a fluid flow analysis for the particular case of one On/Off source being policed. The results obtained with both the simulation experiments and the analysis are almost coincident (this is the reason for not showing confidence intervals with the simulation results). As a consequence, it can be said that the implementation of the policed cell-rate simulator for traffic without priorities and tagging of violating traffic is fairly accurate under diverse system configurations and therefore valid.



(a) source type X2 - low burstiness



(b) source type Y2 - medium burstiness

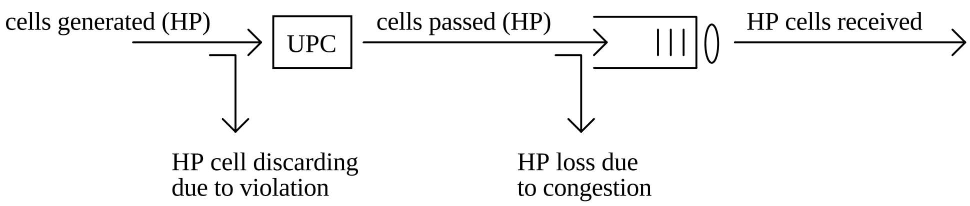


(c) source type Z2 - high burstiness

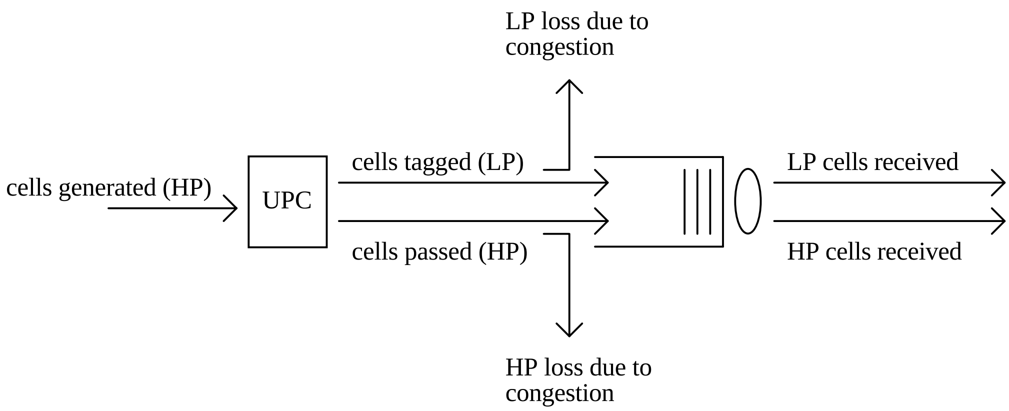
Fig.7.14 - Policed simulation *versus* fluid flow analysis: cell tagging probability (set 2 experiments - fix excess burst size).

7.3. Policing Traffic without Priorities: Discarding or Tagging?

Sections 7.2.2 and 7.2.3 validated *Scenarios 1* and *2* that were implemented by the author in the prioritised cell-rate simulator LINKSIM. Therefore, the UPC actions implemented for each of the policing scenarios can now be investigated and answers to the question raised in the beginning of this Chapter can be found. The question referred to whether the CLP bit can be used efficiently for two different purposes: assignment of priority levels and UPC action on violating traffic. An answer to this question can be found by comparing the performance of the two policing scenarios already validated.



(a) UPC discarding option



(b) UPC tagging option

Fig.7.15 - Traffic loss with policing *Scenarios 1* and *2*: discarding and tagging of traffic without priorities.

Fig.7.15 gives a schematic view of what happens with each policing scenario, i.e., it indicates where there can be traffic loss, either due to violation of traffic parameters at the UPC level or because of congestion in the system's queue. Taking this into account, two main comparisons have been investigated to decide which is the more appropriate UPC action to take on violating traffic without priorities. One of the comparisons involves evaluating the loss of high priority traffic (labelled *HP*) at the system's queue for both scenarios. The other comparison concerns the calculation of the global cell loss probability of the system for both policing scenarios.

Before describing the traffic experiments used to evaluate the performance of policing *Scenarios 1* and *2*, it is important to introduce the measures of interest for each policing scenario.

Let us assume, without the loss of generality, that all the traffic entering the system considered in the scenarios of Fig.7.15 belongs to the same VP. Moreover, suppose that only one VP is present in the system. In this case, the calculation of the cell loss ratios at both the UPC and the queue for the UPC discarding option (see Fig.7.15(a)) takes into account that for each VC,

$$\mathrm{cells\_generated}_{\mathrm{disc}}(\mathrm{i}) = \mathrm{cells\_passed}_{\mathrm{disc}}(\mathrm{i}) + \mathrm{cells\_discarded}_{\mathrm{disc}}(\mathrm{i}),\ \forall \mathrm{i} \quad (7.21)$$

$$\mathrm{cells\_passed}_{\mathrm{disc}}(\mathrm{i}) = \mathrm{cells\_received}_{\mathrm{disc}}(\mathrm{i}) + \mathrm{cells\_lost}_{\mathrm{disc}}(\mathrm{i}),\ \forall \mathrm{i} \quad (7.22)$$

where the index *i* represents the *ith* VC and the subscript “disc” is to distinguish the measures calculated under this policing scenario from those determined with policing *Scenario 2* (that will have the subscript “tag”). Therefore, the global cell loss probability (named *global_clp*) and the HP loss at the queue (*HP_loss*) are calculated by the expressions,

$$\mathrm{global\_clp}_{\mathrm{disc}}(\mathrm{i}) = \frac{\mathrm{cells\_lost}_{\mathrm{disc}}(\mathrm{i}) + \mathrm{cells\_discarded}_{\mathrm{disc}}(\mathrm{i})}{\mathrm{cells\_generated}_{\mathrm{disc}}(\mathrm{i})},\ \forall \mathrm{i} \quad (7.23)$$

$$\mathrm{HP\_loss}_{\mathrm{disc}}(\mathrm{i}) = \frac{\mathrm{cells\_lost}_{\mathrm{disc}}(\mathrm{i})}{\mathrm{cells\_passed}_{\mathrm{disc}}(\mathrm{i})},\ \forall \mathrm{i} \quad (7.24)$$

On the other hand, for policing *Scenario 2* (given in Fig.7.15(b)), the calculation of the loss ratios is slightly different. In this case, we have a similar expression to (7.21) to calculate the number of cells generated,

$$\mathrm{cells\_generated}_{\mathrm{tag}}(\mathrm{i}) = \mathrm{cells\_passed}_{\mathrm{tag}}(\mathrm{i}) + \mathrm{cells\_tagged}_{\mathrm{tag}}(\mathrm{i}) \quad (7.25)$$

$$= \mathrm{cells\_received}_{\mathrm{tag}}(\mathrm{i}) + \mathrm{cells\_lost}_{\mathrm{tag}}(\mathrm{i}),\ \forall \mathrm{i} \quad (7.26)$$

where *cells_received* and *cells_lost* (for each VC) represent the total cells received and lost (respectively) of both low and high priority at the output of the queue. Thus, the global cell loss probability and the losses at the queue (for low and high priority traffic) are calculated by the equations,

$$\mathrm{global\_clp}_{\mathrm{tag}}(\mathrm{i}) = \frac{\mathrm{cells\_lost}_{\mathrm{tag}}(\mathrm{i})}{\mathrm{cells\_generated}_{\mathrm{tag}}(\mathrm{i})},\ \forall \mathrm{i} \quad (7.27)$$

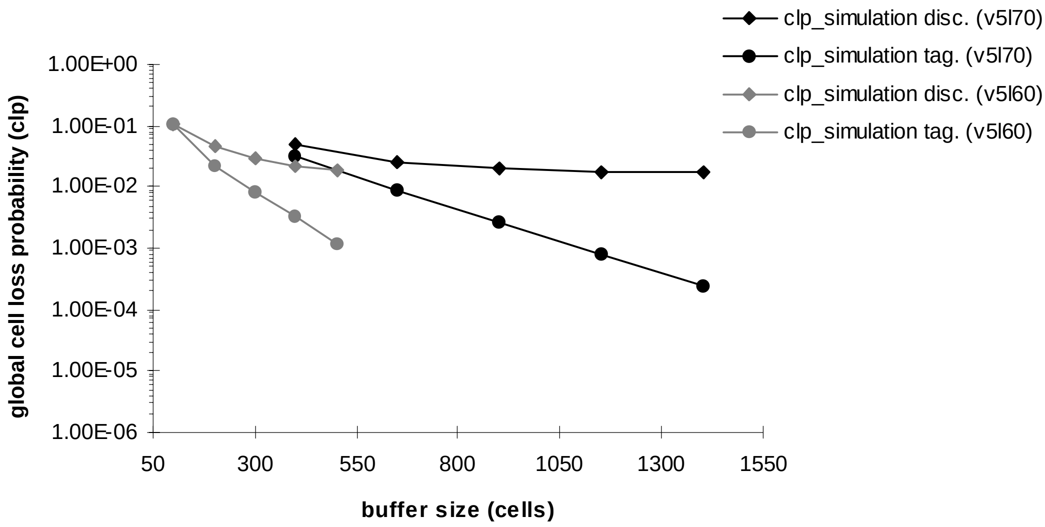
$$\mathrm{HP\_loss}_{\mathrm{tag}}(\mathrm{i}) = \frac{\mathrm{HPcells\_lost}_{\mathrm{tag}}(\mathrm{i})}{\mathrm{cells\_passed}_{\mathrm{tag}}(\mathrm{i})}, \quad \forall \mathrm{i} \tag{7.28}$$

$$\mathrm{LP\_loss}_{\mathrm{tag}}(\mathrm{i}) = \frac{\mathrm{LPcells\_lost}_{\mathrm{tag}}(\mathrm{i})}{\mathrm{cells\_tagged}_{\mathrm{tag}}(\mathrm{i})}, \quad \forall \mathrm{i} \tag{7.29}$$

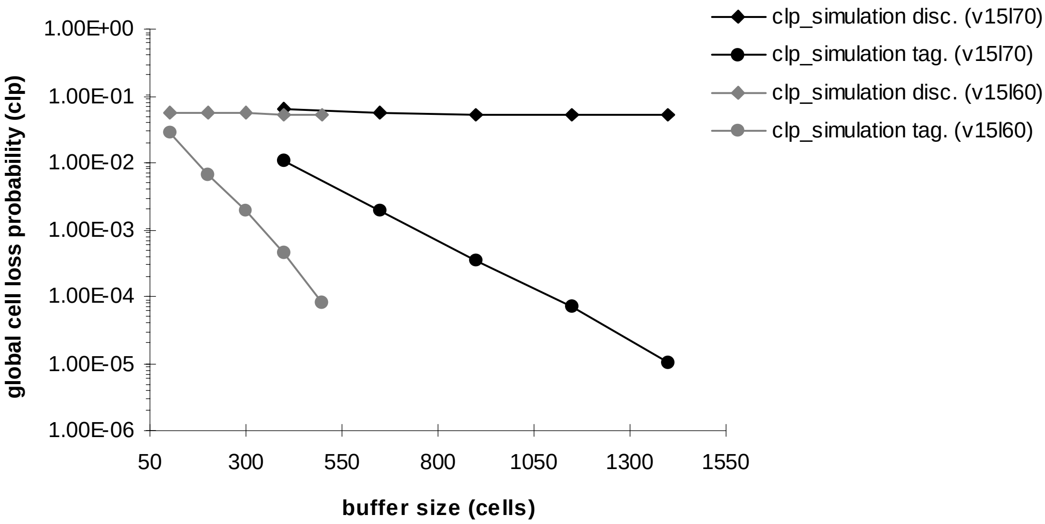
Notice that, although the implementation of the policing scenarios was made on a burst level simulator, the calculation of the performance measures is being presented at cell level. This is because it is easier to interpret the meaning of each measure for presentation purposes. In the policed cell-rate simulator, performance measures such as the number of cells discarded for a given VC are taken as proportions (see Section 7.1.4), since the simulator works with rates of cells, instead of individual cells.

A set of experiments has been carried out to evaluate and compare the measures given by equations (7.23) to (7.24) and (7.27) to (7.29) for the two policing scenarios described in Section 7.1.3. The experiments consider one On/Off modelled traffic source with *On* and *Off* exponentially distributed state durations that is policed under a series of different system configurations. The traffic source types used are *Source X1* and *Source Y1* described in Table 7.3. Moreover, the UPC parameters and the general experimental conditions are also those given in Table 7.2. However, for each source used, only two levels of traffic violation (5% and 15%) are considered. The difference between these experiments and those described in Section 7.2.3 is that for each experiment, both the global load of the system and the buffer size of the system's queue are varied. For source type *X1*, loads of 60% and 70% are considered, while for source type *X2*, the network loads taken are 30% and 40%.

Fig.7.16 gives the results obtained with both policing scenarios for the global cell loss probability of traffic source type *X1* under two different traffic violations. The lines in the graphs that contain the abbreviation "disc." refer to results obtained with the discarding option, while the abbreviation "tag." refers to the tagging option results. For example, the label *clp_simulation disc. (v5l70)* refers to the global cell loss results obtained with the discarding option, for a 5% source traffic violation and a 70% network load. It can be seen that, for a given traffic violation and variable buffer size, the global cell loss probability decreases with both tagging and discarding UPC options. However, the decrease in the global cell loss probability is more noticeable when tagging is used instead of discarding. This pattern in the results also occurs for different traffic loads (60% and 70% in this case). In the case of high traffic violations, the global cell loss probability obtained with the discarding option remains almost constant for varying buffer sizes and different network loads.



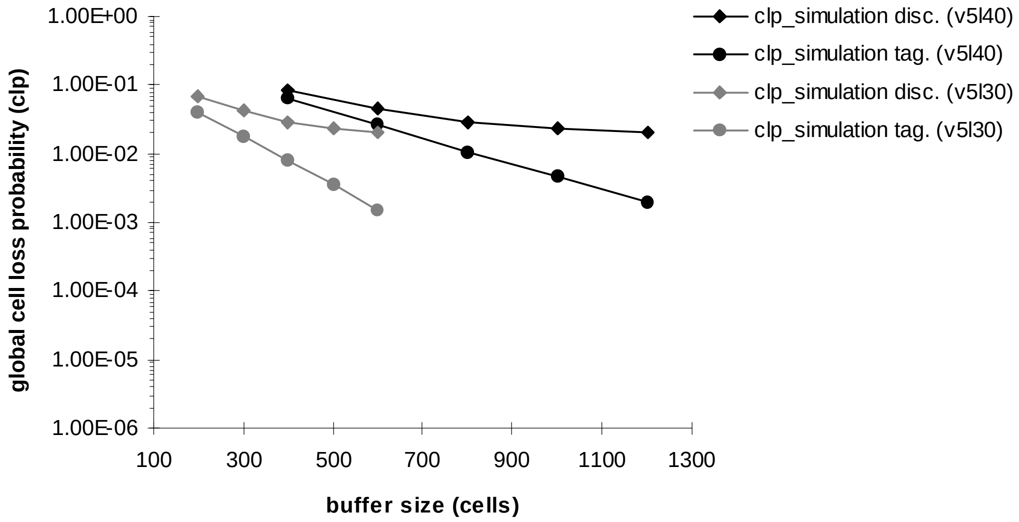
(a) traffic violation = 5%



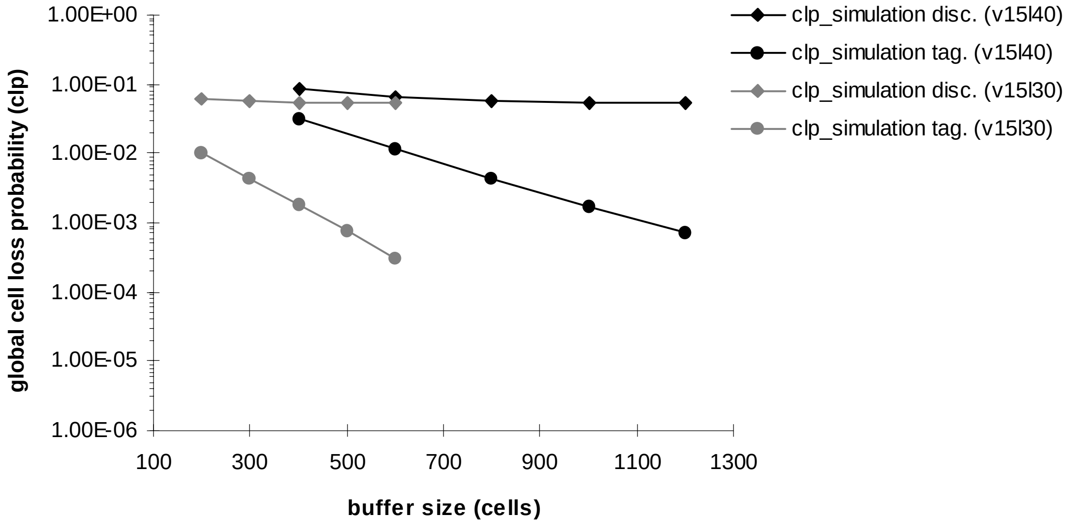
(b) traffic violation = 15%

Fig.7.16 - Discarding *versus* tagging (for traffic without priorities):
global cell loss probability for source type *X1*.

The graphs (in Fig.7.16) also show that there is an improvement in the global cell loss probability obtained by using tagging (instead of discarding) in the case of high traffic violations (and taking into account the given queue configurations). This is because high traffic violations imply a greater tagging rate being input to the queue as low priority traffic. In turn, this results in a better performance of the priority mechanism since there is a better balance of the traffic (of low and high priority) in the queue. For low traffic violations, the amount of traffic tagged (i.e., turned into low priority traffic) at the UPC is a very small percentage of the total input traffic to the system. Thus, there will be a higher probability of that traffic (i.e., low priority traffic) being lost at the queue, where the traffic passed by the UPC (i.e., the high priority traffic) has priority over it.



(a) traffic violation = 5%



(b) traffic violation = 15%

Fig.7.17 - Discarding *versus* tagging (for traffic without priorities):
global cell loss probability for source type *Y1*.

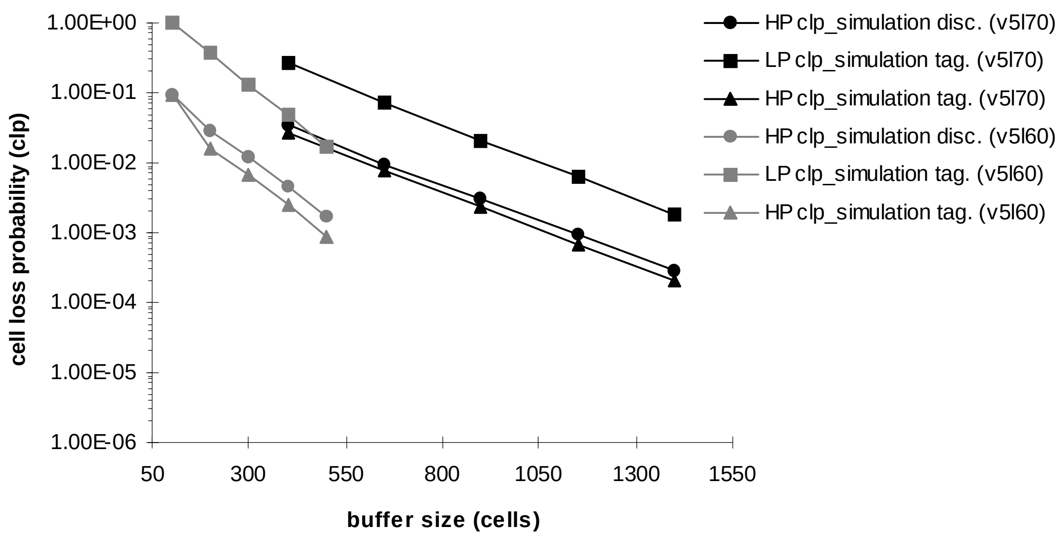
Obviously, the previous statements concerning the performance of the UPC mechanism when tagging is used are true under the circumstances (i.e., the traffic experimental conditions) considered. It is possible that, given other (less restrictive) link configurations (both in terms of bandwidth capacity and buffer size), slightly different results would be observed. A deeper study, using a wider range of traffic types, as well as policing and queue parameters, should be carried out to obtain a more accurate conclusion for the performance of the UPC mechanism when using either tagging or discarding.

Fig.7.17 presents similar results of global cell loss probability for source type *X2*, under varying network loads (30% and 40% in this case) and for varying traffic violations (5% and 15%). The lines plotted in this graph (Fig.7.17) present similar patterns to those of Fig.7.16. Thus, the same type of conclusions can be made for Fig.7.17 as for Fig.7.16. This means that the burstiness of the traffic does not affect the performance of the policing scenarios. In other words, the graphs in Fig.7.17 show that the use of the tagging option produces better global cell loss results, especially in situations of traffic with a high violation percentage.

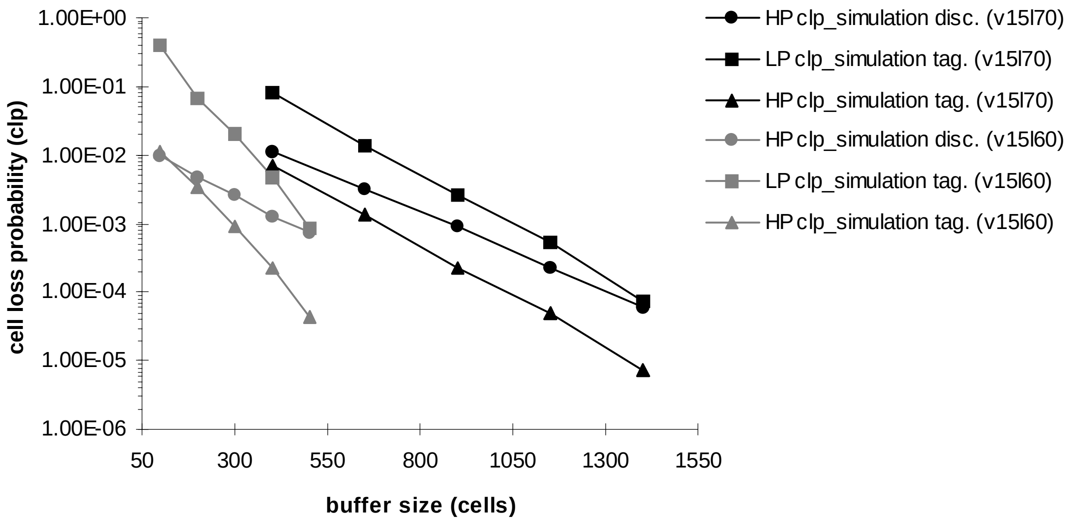
The performance of the two policing scenarios was also tested on what concerns the loss of low priority traffic (i.e., the traffic tagged by the UPC) and high priority traffic (i.e., the input traffic by default, since no priorities are assigned to the traffic at the system's input) at the entrance of the system's queue.

Fig.7.18 presents the results obtained with both tagging and discarding UPC options for the HP loss and LP loss at the queue. The results shown relate to source type *X1*.

Under a 5% traffic violation and varying traffic loads (60% and 70% in this case), the graph in Fig.7.18(a) shows that the HP loss at the queue obtained with both discarding and tagging options is very similar for a wide range of buffer sizes. For a higher traffic violation (as shown in Fig.7.18(b)), the HP losses obtained with the tagging option are much lower than when using the discarding option. This is because high traffic violations lead to a better performance of the priority mechanism in the system's queue as referred to in the discussion of Fig.7.16.



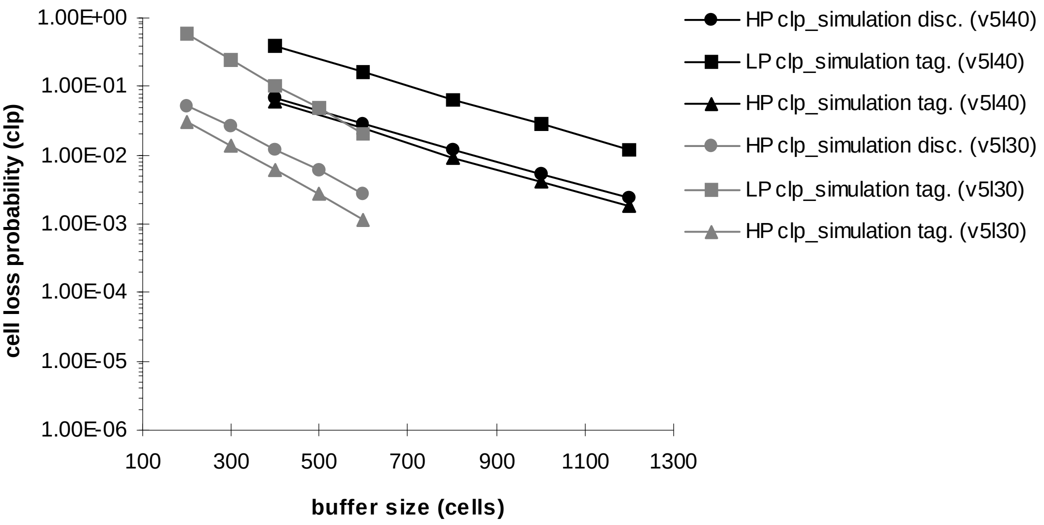
(a) traffic violation = 5%



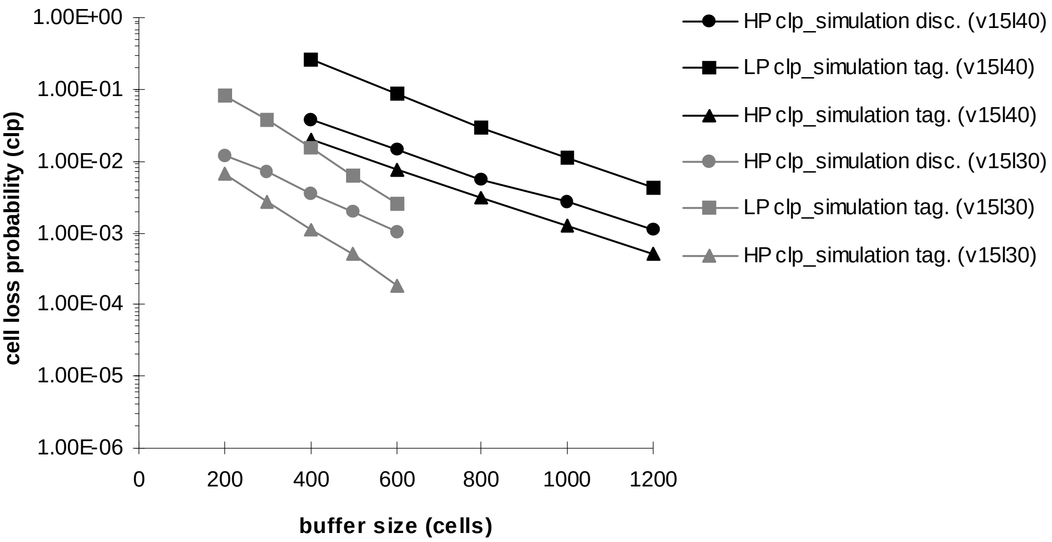
(b) traffic violation = 15%

Fig.7.18 **-** Discarding *versus* tagging (for traffic without priorities): LP cell loss and HP cell loss at the queue for source type *X1*.

Fig.7.18 also contains results obtained for the loss of low priority traffic at the queue (i.e., the traffic tagged by the UPC mechanism that is lost at the queue). It shows that the low priority loss is always greater (in all circumstances studied) than the corresponding HP loss obtained with the tagging option. In other words, most of the traffic tagged by the UPC is lost at the queue.



(a) traffic violation = 5%



(b) traffic violation = 15%

Fig.7.19 - Discarding *versus* tagging (for traffic without priorities): LP cell loss and HP cell loss at the queue for source type *Y1*.

Fig.7.19 gives the same type of results as Fig.5.18 but for source type *Y1*. Once more, the burstiness of the traffic does not seem to affect the results obtained for the HP loss and LP loss at the system's queue.

In summary, the performance of the two policing scenarios implemented (i.e., tagging and discarding for traffic without priorities) does not seem to depend on the burstiness of the traffic. However, the percentage of traffic violation considered is an important factor to take into account in the choice of either policing scenario. In fact, *tagging* appears to perform better for high traffic violations, leading to both smaller LP losses and HP losses than for low traffic violations. In the case of low traffic violations, the HP loss results from both policing scenarios are very similar. Choosing the tagging UPC option in this situation leads to a greater complexity of the system (because of the handling of the priority mechanism in the system's queue), which is not compensated for in terms of performance. However, this is not an exhaustive treatment of the issue.

7.4. Validation of the Policed LINKSIM for Traffic with Priorities

Two policing scenarios for traffic with priorities have been previously described in Section 7.1.3: they were named policing *Scenario 3* and policing *Scenario 4*. Both scenarios consider the situation of traffic with priorities being monitored in two phases by Leaky Bucket mechanisms. In the second phase (now called *Test 0+1*), any misbehaving traffic (i.e., be it low or high priority traffic) is discarded. The first phase (called *Test 0*) only monitors the high priority traffic and it can either discard (in the case of *Scenario 3*) or tag (in the case of *Scenario 4*) violating high priority traffic.

This Section describes how policing *Scenarios 3* and *4* were validated by using an approximate analysis, adapted to the case of traffic with priorities. The analysis (see [Robe92, pp.150-152]) used provides an exact formula to calculate the loss probability for multiple homogeneous On/Off modelled traffic sources in a link where the queue is assumed to have a zero size. The analysis thus provides an upper bound for the cell loss probability. From this simple analysis, two approximate analytical methods were derived by the author in order to validate the policing mechanisms implemented in each of the scenarios for traffic with priorities. This represents new work by the author.

The two new analytical methods give upper bounds for the global cell loss probability at each of the policing tests (i.e., the policing test on the high priority traffic and the policing test on the global traffic) for *Scenarios 3* and *4*.

7.4.1. The Approximate Analysis

As referred to in the previous Section, the policed cell-rate simulator for traffic with priorities has been validated by adapting an analysis for multiple homogeneous On/Off sources. The analysis, reported in [Robe92, pp.150-152] and briefly described here, considers the superposition of *N* homogeneous and independent On/Off modelled traffic sources, characterised by their mean and peak bit rates (represented by *mbr* and *pbr*, respectively). Taking into account the following notation,

N	=	total number of On/Off sources
mbr	=	mean bit rate of each source
pbr	=	peak bit rate of each source
prob_on	=	probability of each source being On
C	=	service capacity of the link
Nmax	=	maximum number of sources (at peak rate) that can fit into the link's service capacity
P(n)	=	probability of *n* sources being active, out of a total of *N* sources
Ploss	=	cell loss probability

it is then possible to calculate the exact cell loss probability for a bufferless link, given that,

$$\mathrm{prob\_on} = \frac{\mathrm{mbr}}{\mathrm{pbr}} \tag{7.30}$$

and

$$\mathrm{Nmax} = \frac{\mathrm{C}}{\mathrm{pbr}}. \tag{7.31}$$

Since it is assumed that the sources are independent, a Binomial distribution is used to calculate the probability *P(n)* of, at any arbitrary moment having *n* sources active, out of the total *N* sources. Therefore, *P(n)* is given by the expression,

$$\mathrm{P(n)} = \frac{\mathrm{N!}}{\mathrm{n!}\cdot(\mathrm{N}-\mathrm{n})!}\cdot \mathrm{prob\_on}^{\mathrm{n}}\cdot(1-\mathrm{prob\_on})^{\mathrm{N-n}}, \quad \mathrm{n} = 0,\ldots,\mathrm{N} \tag{7.32}$$

and the global cell loss probability will be

$$\mathrm{Ploss} = \frac{1}{\mathrm{N}\cdot\mathrm{prob\_on}}\cdot\sum_{\mathrm{k=ceil(Nmax)}}^{\mathrm{N}}\mathrm{P(k)}\cdot(\mathrm{k}-\mathrm{Nmax}) \tag{7.33}$$

where *ceil(Nmax)* represents the smallest integer that is greater than or equal to *Nmax*.

Taking into account the analysis just described, two approximate analytical methods were derived by the author to validate policing *Scenarios 3* and *4*. Bearing in mind the diagrams shown in Fig.7.1 (see Section 7.1.2), it can be seen that it is necessary to evaluate the loss probability (or rather, the discarding or tagging probabilities) at each of the policing tests (i.e., one test on the high priority traffic, *Test 0*, and another on the global traffic, *Test 0+1*). Therefore, the distribution of the traffic going through each of the tests should be known, at least in an approximate way. If we assume that at each policing test, no CDV tolerance is provided to the incoming traffic (i.e., the maximum bucket size is considered to be null), then it is possible to use the analysis above to calculate the discarding/tagging probability at each policing test. Indeed, although the mode of operation of a Leaky Bucket is slightly different from that of a normal queue, they work virtually the same way when no queuing is considered.

In the next two Sections, two approximate analytical methods are described by the author that cater for the situations of discarding and tagging at the first policing test, as well as discarding in the second policing test. The two approximate analytical methods represent new work by the author.

7.4.1.1. Analysis for Discarding

One of the policing scenarios for traffic with priorities that was implemented in the prioritised cell-rate simulator, *Scenario 3*, considers the situation whereby traffic with priorities is monitored in two stages. In the first stage, all the incoming high priority traffic to the system is monitored. Any high priority traffic found to be violating the traffic contract will be discarded. So, the first policing test (i.e., *Test 0*) will provide as input to the second policing test (see Fig.7.1(a)) all the compliant high priority traffic (i.e., all the passed high priority traffic). In its turn, the second policing test (i.e., *Test 0+1*) will discard any violating traffic (be it of low or high priority), after updating what is considered to be the global traffic. The updated global traffic that is input to the second policing test will be the sum of the low priority traffic and the passed high priority traffic. At the entrance to the queue (that follows the second policing test), a similar procedure takes place and traffic is now lost whenever its total input rate amounts to more than the queue's service capacity (since a zero queue size is assumed).

Let us assume, for simplicity reasons, that the system's traffic is comprised of independent low and high priority traffic sources, all modelled as On/Off sources and all with the same traffic parameters, namely the mean bit rate (represented by *mbr*, as seen in the previous Section) and the peak bit rate (given by *pbr*). Moreover, let us take a similar notation to that used in Section 7.4.1 and define,

N_lp (N_hp)	=	total number of low (high) priority sources
prob_on	=	probability of each source being On, as given by equation (7.30)
C_hp	=	leak rate of *Test 0*
C_glob	=	leak rate of *Test 0+1*
C_queue	=	service capacity of the queue
Nmax_hp	=	maximum number of high priority sources (at peak rate) that can fit into the *Test 0* leak rate
Nmax_glob	=	maximum number of sources (at peak rate) that can fit into the *Test 0+1* leak rate
Nmax_queue	=	maximum number of sources (at peak rate) that can fit into the queue's service capacity
P_lp(n)	=	probability of *n* low priority sources being active, out of a total of *N_lp* sources
P_hp(n)	=	probability of *n* high priority sources being active, out of a total of *N_hp* sources
globdist_0+1(n)	=	distribution for the traffic at the input of *Test 0+1*
queuedist(n)	=	distribution for the traffic at the input of the queue
Pdiscard_0	=	cell discarding probability at *Test 0*
Pdiscard_0+1	=	cell discarding probability at *Test 0+1*
Ploss	=	cell loss probability at the queue.

With this notation and applying the reasoning of the analysis described in Section 7.4.1, it is easy to calculate the cell discarding probability for the high priority traffic at *Test 0*. It will be given by a formula that is identical to equation (7.33),

$$\text{Pdiscard\_0} = \frac{1}{\text{N\_hp} \cdot \text{prob\_on}} \cdot \sum_{\text{i=ceil(Nmax\_hp)}}^{\text{N\_hp}} \text{P\_hp}(\text{i}) \cdot (\text{i} - \text{Nmax\_hp}). \qquad (7.34)$$

At the output of *Test 0*, the distribution of the high priority traffic will no longer be defined by *P_hp(n)*. Instead, it is now described by the following function

$$\mathrm{newP\_hp}(n)=\begin{cases}\mathrm{P\_hp}(n)\,, & n=0,\ldots,\mathrm{ceil}(\mathrm{Nmax\_hp})-1\\ \displaystyle\sum_{i=\mathrm{ceil}(\mathrm{Nmax\_hp})}^{\mathrm{N\_hp}}\mathrm{P\_hp}(i)\,, & n=\mathrm{ceil}(\mathrm{Nmax\_hp})\end{cases} \tag{7.35}$$

Equation (7.35) illustrates the fact that when at least *Nmax_hp* sources are active, then there is an excess rate state and traffic is discarded by *Test 0*.

To calculate the cell discarding probability *at Test 0+1*, it is necessary to know the distribution of the global traffic at the input to that test. Therefore, the distribution of the high priority traffic at the output of *Test 0* (given by equation (7.35)) and the distribution of the low priority traffic must be convolved. This will produce the global distribution at the input of *Test 0+1*, defined as *globdist_0+1(n)* and given by the formula,

$$\mathrm{globdist\_0}+1(n)=\sum_{k=0}^{n}\mathrm{P\_lp}(k)\cdot\mathrm{newP\_hp}(n-k)\,, \tag{7.36}$$

$$n=0,\ldots,\mathrm{N\_lp}+\mathrm{ceil}(\mathrm{Nmax\_hp})$$

With formula (7.36), it is now possible to determine the cell discarding probability at *Test 0+1*, taking into account that the mean input rate to the second policing test is

$$\left(\mathrm{N\_lp}+\mathrm{N\_hp}\cdot(1-\mathrm{Pdiscard\_0})\right)\cdot\mathrm{mbr}\,. \tag{7.37}$$

Pdiscard_0+1 will therefore be given by,

$$\mathrm{Pdiscard\_0}+1=\frac{\displaystyle\sum_{i=\mathrm{ceil}(\mathrm{Nmax\_glob})}^{\mathrm{N\_lp}+\mathrm{ceil}(\mathrm{Nmax\_hp})}\mathrm{globdist\_0}+1(i)\cdot(i-\mathrm{Nmax\_glob})}{\left(\mathrm{N\_lp}+\mathrm{N\_hp}\cdot(1-\mathrm{Pdiscard\_0})\right)\cdot\mathrm{prob\_on}}\,. \tag{7.38}$$

Finally, in order to calculate the input distribution for the system's queue and the cell loss probability at the queue, similar approaches are taken to those of equations (7.35) and (7.38), respectively. So, for the queue's input distribution, the distribution at the input to *Test 0+1* takes into account that *Test 0+1* will be in an excess rate state whenever the total input rate combines to more than the leak rate of that policing test. In view of this, *queuedist(n)* will be given by,

$$\mathrm{queuedist}(n)=\begin{cases}\mathrm{globdist\_0}+1(n)\,, & n=0,\ldots,\mathrm{ceil}(\mathrm{Nmax\_glob})-1\\ \displaystyle\sum_{i=\mathrm{ceil}(\mathrm{Nmax\_glob})}^{\mathrm{N\_lp}+\mathrm{ceil}(\mathrm{Nmax\_hp})}\mathrm{globdist}(i)\,, & n=\mathrm{ceil}(\mathrm{Nmax\_glob})\end{cases} \tag{7.39}$$

As for the cell loss probability at the queue, *Ploss*, it is determined by taking into account that the mean input rate to the queue is

$$\left(\mathrm{N\_lp} + \mathrm{N\_hp} \cdot \left(1 - \mathrm{Pdiscard\_0}\right)\right) \cdot \left(1 - \mathrm{Pdiscard\_0+1}\right) \cdot \mathrm{mbr}\,. \tag{7.40}$$

So, *Ploss* will be given by the formula

$$\mathrm{Ploss} = \frac{\sum\limits_{i=\mathrm{ceil}(\mathrm{Nmax\_queue})}^{\mathrm{ceil}(\mathrm{Nmax\_glob})} \mathrm{queuedist}(i) \cdot \left(i - \mathrm{Nmax\_queue}\right)}{\left(\mathrm{N\_lp} + \mathrm{N\_hp} \cdot \left(1 - \mathrm{Pdiscard\_0}\right)\right) \cdot \left(1 - \mathrm{Pdiscard\_0+1}\right) \cdot \mathrm{prob\_on}}\,. \tag{7.41}$$

It is worthwhile noticing that the value of cell loss probability obtained for the queue is a global value, i.e., with this approach, it is not possible to know the individual low and high priority values of cell loss probability at the queue (or at *Test 0+1*, for that matter). So, with this analysis, the processing of the priority mechanism present in the queue is not taken into account. Although this is a drawback of the analysis used, it does not invalidate the results obtained because the use of priority mechanisms does not provide a means of reducing the global cell loss probability of a system, it only redistributes the loss according to the priority of the different traffic streams in a system.

7.4.1.2. Analysis for Tagging

The policing *Scenario 4* described in Section 7.1.3 and whose implementation in the prioritised cell-rate simulator was explained in Section 7.1.4, caters for traffic with priorities and the high priority traffic is tagged when violating the traffic contract. The monitoring of the global traffic for this scenario results in the discarding of any misbehaving traffic. In other words, *Test 0* tags violating high priority traffic (and therefore, no traffic is ever lost at *Test 0* when this option is used) and, after updating the total traffic at the input to *Test 0+1*, this test discards any violating traffic. The updated input traffic to *Test 0+1* will be the sum of the low priority traffic, the tagged high priority traffic (now viewed as low priority traffic) and the high priority traffic passed at *Test 0*. At the entrance to the queue, the incoming traffic will be the sum of the low and high priority traffic that was passed at *Test 0+1*. Once more, loss at the queue will occur whenever the total incoming input rate is greater than the queue's service capacity (because no buffering is allowed to the excess traffic).

Taking into account the description of the approximate analysis for policing *Scenario 3* (see Section 7.4.1.1), it is easy to see that the same reasoning can be applied to *Scenario 4*, as far as the first policing test (i.e., the test on the high priority traffic, *Test 0*) is concerned. Therefore, if a similar mathematical notation to that used for *Scenario 3* (see Section 7.4.1.1) is taken into account, then equation (7.34) is still valid. However, that equation now relates to the probability of tagging high priority traffic (labelled as *Ptag_0*) in *Test 0*. In other words, *Ptag_0* will be given by,

$$\mathrm{Ptag\_0} = \frac{1}{\mathrm{N\_hp} \cdot \mathrm{prob\_on}} \cdot \sum_{\mathrm{i=ceil(Nmax\_hp)}}^{\mathrm{N\_hp}} \mathrm{P\_hp}(\mathrm{i}) \cdot (\mathrm{i} - \mathrm{Nmax\_hp}). \tag{7.42}$$

At the input of *Test 0+1*, no traffic (of either priority type) will have been lost, so the global traffic is still the same as before the execution of *Test 0*. Thus, it can be said that no change has occurred in the distribution of both the low and high priority traffic streams in the system. This simplifies the calculation of the distribution for the total traffic at the input of *Test 0+1*, since it will be sufficient to determine the convolution of the low and high priority traffic distributions. Taking this into account, *globdist_0+1(n)* will be given by,

$$\mathrm{globdist\_0+1}(\mathrm{n}) = \sum_{\mathrm{k=0}}^{\mathrm{n}} \mathrm{P\_lp}(\mathrm{k}) \cdot \mathrm{P\_hp}(\mathrm{n}-\mathrm{k}), \tag{7.43}$$

$$\mathrm{n} = 0,\ldots,\mathrm{N\_lp} + \mathrm{N\_hp}$$

Having determined the distribution for the global traffic at *Test 0+1*, it is now possible to know how much traffic is discarded (if any) at that policing test. That calculation again takes into account the value of the mean global input rate at the input to *Test 0+1*, which is given by

$$(\mathrm{N\_lp} + \mathrm{N\_hp}) \cdot \mathrm{mbr}, \tag{7.44}$$

since no traffic has yet been lost/discarded. So, similarly to equation (7.38), the value of *Pdiscard_0+1* is calculated for policing *Scenario 4*, taking (7.44) into account:

$$\mathrm{Pdiscard\_0+1} = \frac{\displaystyle\sum_{\mathrm{i=ceil(Nmax\_glob)}}^{\mathrm{N\_lp+N\_hp}} \mathrm{globdist\_0+1}(\mathrm{i}) \cdot (\mathrm{i} - \mathrm{Nmax\_glob})}{(\mathrm{N\_lp} + \mathrm{N\_hp}) \cdot \mathrm{prob\_on}}. \tag{7.45}$$

Finally, the formula for the distribution of the global traffic at the input to the queue is also similar to the expression used in policing *Scenario 3*, except that in the present case no traffic is considered to be lost before the execution of *Test 0+1*. Therefore, *queuedist(n)* will given by,

$$\text{queuedist}(n) = \begin{cases} \text{globdist\_0}+1(n)\,, & n = 0,\ldots,\text{ceil}(\text{Nmax\_glob})-1 \\ \displaystyle\sum_{i=\text{ceil}(\text{Nmax\_glob})}^{\text{N\_lp+N\_hp}} \text{globdist}(i)\,, & n = \text{ceil}(\text{Nmax\_glob}) \end{cases} \tag{7.46}$$

and the global cell loss probability for the queue will again be calculated by taking into account the mean input rate at the entrance of the queue; this is,

$$(\text{N\_lp}+\text{N\_hp})\cdot(1-\text{Pdiscard\_0}+1)\cdot \text{mbr}\,. \tag{7.47}$$

The global cell loss probability will then be given by the following expression,

$$\text{Ploss} = \frac{\displaystyle\sum_{i=\text{ceil}(\text{Nmax\_queue})}^{\text{ceil}(\text{Nmax\_glob})} \text{queuedist}(i)\cdot(i-\text{Nmax\_queue})}{(\text{N\_lp}+\text{N\_hp})\cdot(1-\text{Pdiscard\_0}+1)\cdot \text{prob\_on}}\,. \tag{7.48}$$

7.4.2. Policed Cell-Rate Simulation *versus* Analysis: Discarding Violating High Priority Traffic

In Section 7.4.1.1, an approximate analysis was derived by the author to validate the implementation of policing *Scenario 3* in the prioritised cell-rate simulator. The analysis gave an upper bound for,

- the cell discarding probability at policing *Test 0* and policing *Test 0+1*,
- the cell loss probability at the queue,

for a network link where it is assumed that no CVD tolerance is given to the traffic (i.e., the buckets in policing *Test 0* and *Test 0+1* have null size) and no queuing is allowed in the system's queue (i.e., the queue has a zero size). Here, a series of traffic experiments are carried out that consider the multiplexing of homogeneous On/Off modelled traffic sources; they aim at validating the implementation of policing *Scenario 3* and evaluating the accuracy of the simulator. Results are obtained for the traffic experiments by using both the policed cell-rate simulator and the approximate analysis described in Section 7.4.1.1.

The traffic experiments use homogeneous On/Off sources for which parameters such as the burstiness are varied and VP policing is assumed. The first set of experiments concerns the multiplexing of low burstiness On/Off sources (some of which are taken to be high priority traffic, while the rest represent low priority traffic) that represent an overall network load of about 53%. Table 7.6 gives the traffic parameters used in the experiments, where the leak rate of policing *Test 0* was varied. Notice that, in order for the policed simulator's software to work, non-null values had to be considered for the sizes of the buckets and the size of the queue. They are the minimum integer values allowed by the cell-rate simulator.

Low burstiness On/Off sources (burstiness = 5)			
peak rate (pbr)	mean rate (mbr)	On state duration	Off state duration
10000 cell/s	2000 cell/s	25 ms	100 ms

(a) sources' parameters

Number of sources: 25 LP sources; 25 HP sources					
Test 0		*Test 0+1*		*Queue*	
bucket (0)	leak rate (0)	bucket (0+1)	leak rate (0+1)	buffer size	link capacity
1	100000 cell/s to 180000 cell/s (*)	1	200000 cell/s	1 cell	190000 cell/s

(b) system parameters

Table 7.6 - Parameters of experiments for policing low burstiness sources (*Scenario 3*);
(*) increments of 20000 cells.

Fig.7.20 gives the results for the cell discarding probability (at the two policing tests, *Test 0* and *Test 0+1*) and the cell loss probability at the queue that were obtained with both the policed cell-rate simulator and the approximate analysis described in Section 7.4.1.1.

The graph in Fig.7.20 shows that there is a good agreement between the analysis and the policed simulator for varying leak rates in *Test 0*. Indeed, the analysis always provides upper bounds for both the cell discarding and cell loss probabilities, and the simulation results follow the same trend as the analytical ones. It can also be seen that, as the amount of passed high priority traffic at *Test 0* increases, so does the cell discarding probability (cdr) at *Test 0+1* and the cell loss probability (clp) at the queue. However, the great reduction observed in the cell discarding probability at *Test 0* is not shown in the second policing test. Instead, these loss values tend to stabilise. This can be explained by the fact that the load at the queue is always the same (since the leak rate of *Test 0+1* remains constant). The two missing values in the graph (i.e., in Fig.7.20) for the cell discarding probability at *Test 0* were found to be null with the policed cell-rate simulator.

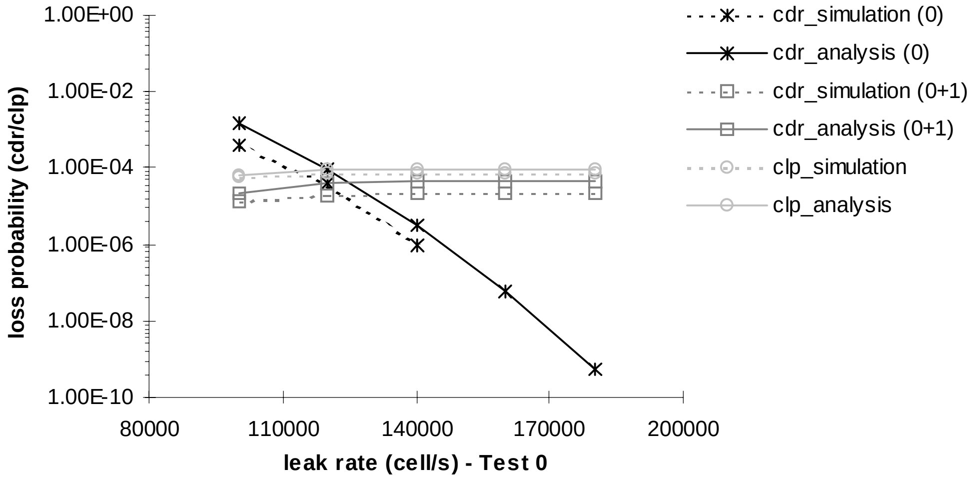


Fig.7.20 - Policed cell-rate simulator with discarding at policing *Test 0 versus* approximate analysis: low burstiness sources (*Scenario 3*).

It is worth noticing that when smaller bucket values are considered with the policed simulator, the simulation results tend to be closer to the analytical results, thus confirming the validation of the policed simulator for traffic with priorities and discarding at *Test 0*. For example, if the bucket sizes for policing *Test 0* and *Test 0+1* are both set to 0.001 with a leak rate in *Test 0* of 100000 cell/s (i.e., the values corresponding to the first set of points in Fig.7.20), then the cell discarding/loss probability values increase between 0.63% and 2.5% relative to the corresponding simulation values in Fig.7.20 for bucket sizes of 1.

The second set of traffic experiments carried out to validate policing *Scenario 3* considers the multiplexing of high burstiness homogeneous On/Off traffic sources. Again, some of the sources are taken to represent low priority traffic while others will be high priority traffic, and the leak rate of *Test 0* is varied. The total number of sources in these experiments now represents a network load of 75%. Table 7.7 gives the details of the traffic sources used, as well as the policing and queue parameters.

Low burstiness On/Off sources (burstiness = 10)			
peak rate (pbr)	mean rate (mbr)	On state duration	Off state duration
30000 cell/s	3000 cell/s	0.11 sec	1 sec

(a) sources' parameters

Number of sources: 25 LP sources; 40 HP sources					
Test 0		*Test 0+1*		*Queue*	
bucket (0)	leak rate (0)	bucket (0+1)	leak rate (0+1)	buffer size	link capacity
1	210000 cell/s to 290000 cell/s (*)	1	350000 cell/s	1 cell	260000 cell/s

(b) system parameters

Table 7.7 - Parameters of experiments for policing high burstiness sources (*Scenario 3*);

(*) increments of 20000 cells.

The results for the simulation and analytical approaches under the experimental conditions described by Table 7.7 are shown in Fig.7.21. The plotted lines again show that the policed cell-rate simulator produces fairly accurate results for both policing *Test 0* and *Test 0+1*. The results of this graph (i.e., Fig.7.21) are shown on the same logarithmic scale as those of Fig.7.20, for a more exact evaluation of the policed simulator's accuracy.

The graph (see Fig.7.21) shows that for increasing leak rates of the policing *Test 0*, there is an increase in both the cell discarding probability at *Test 0+1* and the cell loss probability at the queue, which is more noticeable for the cell discarding probability at *Test 0+1*.

In summary, the two sets of experiments described in this Section have considered the multiplexing of homogeneous sources with low and high burstiness values. Also, the experiments were carried out for two different network loads and varying policing rates for the high priority traffic. The results from the policed cell-rate simulator have proved to be fairly accurate (when compared with the upper bound analysis described in Section 7.4.1.1) under the conditions of the two sets of experiments. This validates the implementation of policing *Scenario 3*.

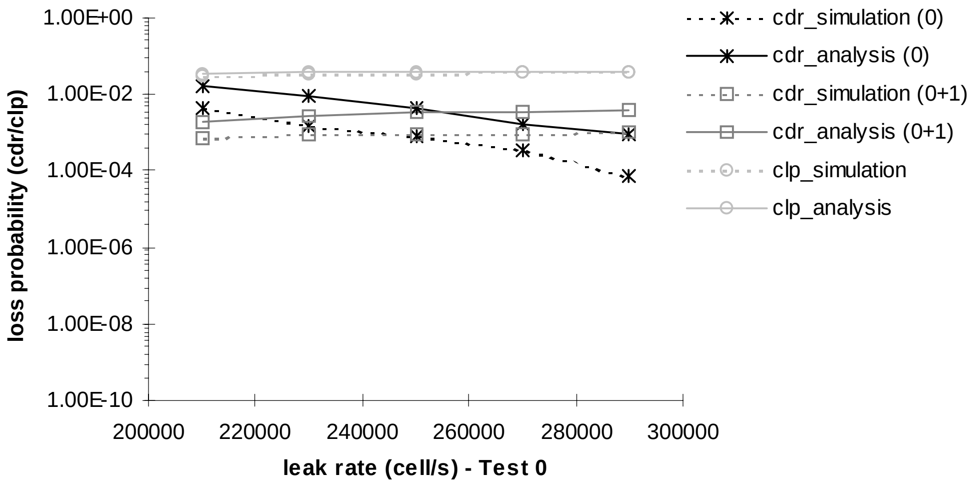


Fig.7.21 - Policed cell-rate simulator with discarding at policing *Test 0 versus* approximate analysis: high burstiness sources (*Scenario 3*).

7.4.3. Policed Cell-Rate Simulation *versus* Analysis: Tagging Violating High Priority Traffic

The policing *Scenario 4* has been described in Section 7.1.3 and Section 7.1.4 as allowing the study of traffic with priorities where the UPC action taken on violating high priority traffic is tagging. On the other hand, in Section 7.4.1.2, an approximate analysis was described by the author that provides upper bounds for the cell tagging probability (ctr) at *Test 0*, the cell discarding probability (cdr) at *Test 0+1* and the cell loss probability (clp) at the queue of a network link. In this Section, that approximate analysis is used to validate policing *Scenario 4*. To this end, two sets of experiments (similar to those of Section 7.4.2) are carried out to compare the policed cell-rate simulator with the upper bound analysis and evaluate the simulator's accuracy.

The first set of experiments used homogeneous On/Off modelled sources with a low burstiness coefficient that have the same characteristics as those described in Table 7.6. Again, the network load considered is 53% and no queuing nor CDV tolerance are allowed. Although this set of experiments is mainly the same as that given by Table 7.6, in this case, *Test 0* tags any violating high priority traffic, instead of discarding it (as happened in Section 7.4.2).

Fig.7.22 gives the comparative results for the cell tagging probability at *Test 0*, cell discarding probability at *Test 0+1* and cell loss probability at the queue, when using simulation and analysis. The results obtained are very similar to those of

Fig.7.20, especially for the cell tagging probability values. This is because, under the same experimental conditions, the tagging UPC option at the policing *Test 0* is equivalent to the discarding option. In other words, tagging and discarding are the same operation, they select and quantify the violating traffic. The difference is that, with tagging, the violating traffic is not lost (as with the discarding option), it only becomes low priority traffic. Again, the comparison of the simulation and analytical results indicates that the policed cell-rate simulator performs as intended and is fairly accurate for varying policing rates at *Test 0*.

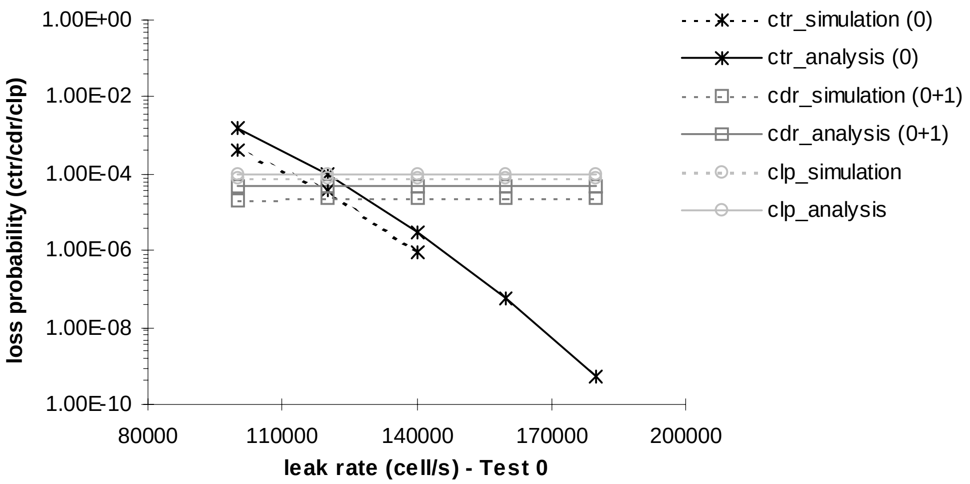


Fig.7.22 - Policed cell-rate simulator with tagging at policing *Test 0 versus* approximate analysis: low burstiness sources (*Scenario 4*).

The second set of experiments used the high burstiness sources' and system configuration given in Table 7.7 and the network load considered was 75%. In this case, any violating high priority traffic is tagged by policing *Test 0*. Fig.7.23 shows the tagging, discarding and loss probability results (at *Test 0*, *Test 0+1* and the queue, respectively) obtained with both the policed simulator and the upper bound analysis. The graph (in Fig.7.23, for the UPC tagging option at *Test 0*) again produces similar results to those of Fig.7.21 for the discarding UPC option on the high priority traffic.

In conclusion, the comparison of the policed cell-rate simulator for policing *Scenario 4* and the corresponding approximate analysis indicates that the policed simulator produces accurate results at both policing tests and the system's queue. The analysis always overestimates the tagging/discarding/loss values, which is explained by the fact that no queuing nor CVD tolerance are accounted for. This means that the implementation of policing *Scenario 4* in the prioritised cell-rate simulator is valid.

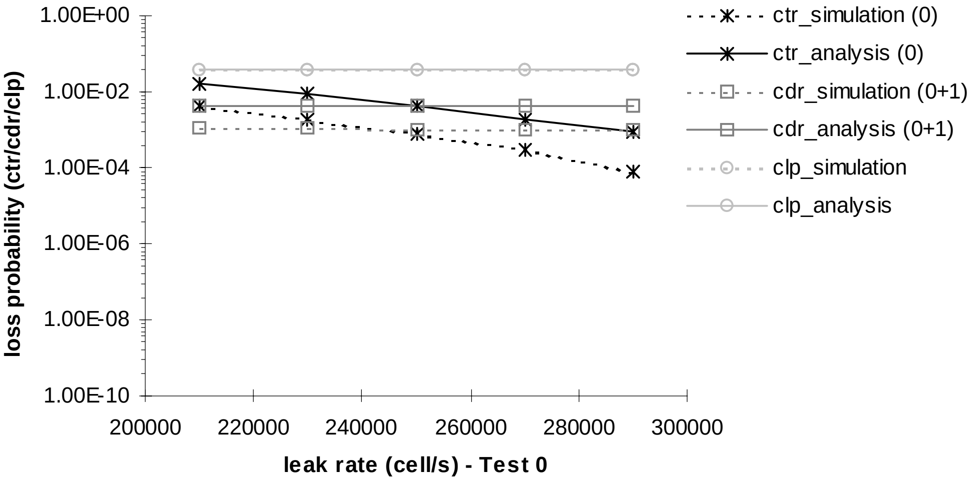


Fig.7.23 - Policed cell-rate simulator with tagging at policing *Test 0 versus* approximate analysis: high burstiness sources (*Scenario 4*).

7.5. Policing Traffic with Priorities: Discarding or Tagging of High Priority Traffic?

The implementation (in the prioritised cell-rate simulator LINKSIM) of the policing *Scenarios 3* and *4* for traffic with priorities, has been validated in Section 7.4.3 and Section 7.4.2, respectively, by making use of a bufferless analysis, adapted by the author to study traffic with priorities. In view of this, it is now possible to compare the performance of both policing scenarios for traffic with priorities, so that conclusions can be taken as to which UPC action is preferable to apply on violating traffic.

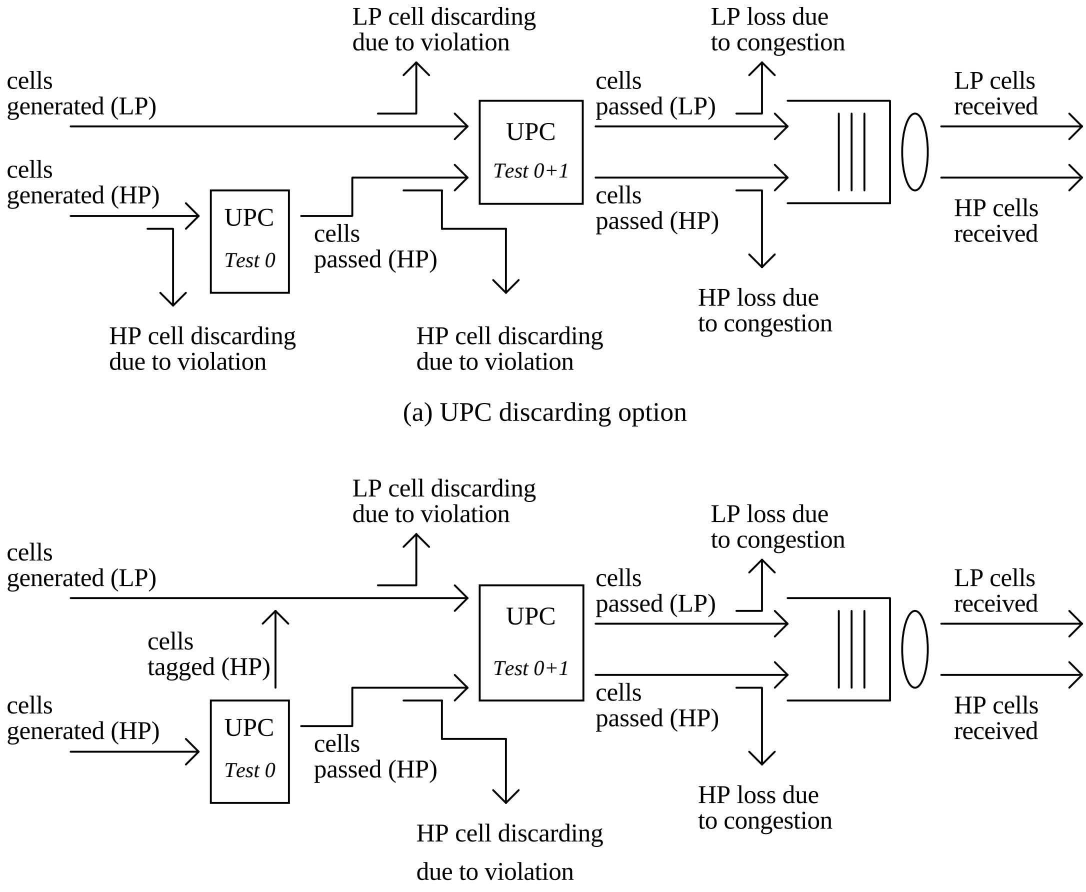
Fig.7.24 shows two diagrams that illustrate the steps through which low and high priority traffic types undergo when in the case of a system where either policing *Scenario 3* or policing *Scenario 4* has been implemented, as proposed by ITU in [ITU95b].

In a similar manner to the description in Section 7.3 for the performance evaluation of policing *Scenarios 1* and *2,* this Section will consider two main comparisons to help suggesting the best UPC actions on violating traffic with priorities:

1. calculation of the system's global cell loss probability for policing *Scenarios 3* and *4* (i.e., when *Test 0* discards or tags violating high priority

traffic, respectively);

2. evaluation of the loss for low and high priority traffic, when either tagging or discarding of high priority traffic is applied in the case of traffic contract violation.



(b) UPC tagging option

Fig.7.24 - Traffic loss with policing *Scenarios 3* and *4*: discarding and tagging of high priority traffic.

Before describing and analysing the traffic experiments used in the performance evaluation of the two policing scenarios for traffic with priorities, the measures of interest for each scenario will be introduced.

Let us assume that all the input traffic to the systems considered in Fig.7.24 belongs to the same VP. Suppose also that all the traffic sources in the system have the same characteristics (i.e., traffic parameters) and that some represent low priority traffic sources, while others are high priority traffic sources. Then, the calculation of the cell loss ratios for policing *Scenario 3* (see Fig.7.24(a)) at the UPC *Test 0* (i.e., policing of the high priority traffic), at *Test 0+1* (i.e., policing of the global traffic) and at the queue takes into account that for each VC,

LP traffic

$$\mathrm{LPcells\_gen_{disc}(i) = LPcells\_pass01_{disc}(i) + LPcells\_disc01_{disc}(i),\ \ \forall i} \tag{7.49}$$

$$\mathrm{LPcells\_pass01_{disc}(i) = LPcells\_rec_{disc}(i) + LPcells\_lost_{disc}(i),\ \ \forall i} \tag{7.50}$$

HP traffic

$$\mathrm{HPcells\_gen_{disc}(i) = HPcells\_pass0_{disc}(i) + HPcells\_disc0_{disc}(i),\ \ \forall i} \tag{7.51}$$

$$\mathrm{HPcells\_pass0_{disc}(i) = HPcells\_pass01_{disc}(i) + } \\ \mathrm{+HPcells\_disc01_{disc}(i),\ \ \forall i} \tag{7.52}$$

$$\mathrm{HPcells\_pass01_{disc}(i) = HPcells\_rec_{disc}(i) + HPcells\_lost_{disc}(i),\ \ \forall i} \tag{7.53}$$

where the index *i* refers to the *ith* VC and the subscript "disc" is used to distinguish the measures calculated with *Scenario 3* (that considers discarding of violating high priority traffic) from those referring to *Scenario 4* (that considers tagging of violating high priority traffic). For simplicity reasons, an abbreviated notation has been used in equations (7.49) to (7.53) - and will continue to be used throughout this Section. For example, the notation $HPcells\_pass01_{disc}(i)$ refers to the number of high priority cells of the *ith* VC for *Scenario 3* that the policing *Test 0+1* allowed to pass (i.e., low priority cells that were not discarded by the second policing test). The abbreviations "gen", "disc", "tag" and "rec" refer to *cells generated* (at the system's input), *cells discarded* (at either policing test), *cells tagged* (at the first policing test) and *cells received* (at the output of the queue), respectively.

Using equations (7.49) to (7.53), the global cell loss probability (labelled as *global_clp*) is then determined by,

$$\mathrm{global\_clp_{disc}(i) = \frac{cells\_lost_{disc}(i) + cells\_disc_{disc}(i)}{cells\_gen_{disc}(i)},\ \ \forall i} \tag{7.54}$$

where,

$$\mathrm{cells\_lost_{disc}(i) = LPcells\_lost_{disc}(i) + HPcells\_lost_{disc}(i),\ \ \forall i} \tag{7.55}$$

$$\mathrm{cells\_disc_{disc}(i) = LPcells\_disc01_{disc}(i) + HPcells\_disc0_{disc}(i) + } \\ \mathrm{+HPcells\_disc01_{disc}(i),\ \ \forall i} \tag{7.56}$$

$$\mathrm{cells\_gen_{disc}(i) = LPcells\_gen_{disc}(i) + HPcells\_gen_{disc}(i),\ \ \forall i} \tag{7.57}$$

On the other hand, the total low and high priority losses, named *LP_loss* and *HP_loss*, respectively, are given by,

$$\mathrm{LP\_loss_{disc}(i)} = \frac{\mathrm{LPcells\_disc01_{disc}(i) + LPcells\_lost_{disc}(i)}}{\mathrm{LPcells\_gen_{disc}(i)}}, \quad \forall \mathrm{i} \tag{7.58}$$

$$\mathrm{HP\_loss_{disc}(i)} = \frac{1}{\mathrm{HPcells\_gen_{disc}(i)}} \cdot \big(\mathrm{HPcells\_disc0_{disc}(i)} +$$

$$+\mathrm{HPcells\_disc01_{disc}(i) + HPcells\_lost_{disc}(i)}\big), \quad \forall \mathrm{i} \tag{7.59}$$

For policing *Scenario 4* (see Fig.7.24(b)), the calculation of the loss ratios is similar to the calculations used for *Scenario 3*. Indeed, the formulas for the total number of cells generated (of either priority type) are as follows,

LP traffic

$$\mathrm{LPcells\_gen_{tag}(i) = LPcells\_pass01_{tag}(i) + LPcells\_disc01_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.60}$$

$$\mathrm{LPcells\_pass01_{disc}(i) = LPcells\_rec_{tag}(i) + LPcells\_lost_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.61}$$

HP traffic

$$\mathrm{HPcells\_gen_{tag}(i) = HPcells\_pass0_{tag}(i) + HPcells\_tag0_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.62}$$

$$\mathrm{HPcells\_pass0_{tag}(i) = HPcells\_pass01_{tag}(i)} +$$

$$+\mathrm{HPcells\_disc01_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.63}$$

$$\mathrm{HPcells\_pass01_{tag}(i) = HPcells\_rec_{tag}(i) + HPcells\_lost_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.64}$$

Taking into account equations (7.60) to (7.64), the global cell loss probability, and the low and high priority loss ratios for policing *Scenario 4* are then,

$$\mathrm{global\_clp_{tag}(i)} = \frac{\mathrm{cells\_lost_{tag}(i) + cells\_disc_{tag}(i)}}{\mathrm{cells\_gen_{tag}(i)}}, \quad \forall \mathrm{i} \tag{7.65}$$

where,

$$\mathrm{cells\_lost_{tag}(i) = LPcells\_lost_{tag}(i) + HPcells\_lost_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.66}$$

$$\mathrm{cells\_disc_{tag}(i) = LPcells\_disc01_{tag}(i) + HPcells\_disc01_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.67}$$

$$\mathrm{cells\_gen_{tag}(i) = LPcells\_gen_{tag}(i) + HPcells\_gen_{tag}(i)}, \quad \forall \mathrm{i} \tag{7.68}$$

and

$$\mathrm{LP\_loss_{tag}(i)} = \frac{\mathrm{LPcells\_disc01_{tag}(i) + LPcells\_lost_{tag}(i)}}{\mathrm{LPcells\_gen_{tag}(i)}}, \quad \forall \mathrm{i} \tag{7.69}$$

$$\mathrm{HP\_loss}_{\mathrm{tag}}(\mathrm{i}) = \frac{\mathrm{HPcells\_disc01}_{\mathrm{tag}}(\mathrm{i}) + \mathrm{HPcells\_lost}_{\mathrm{tag}}(\mathrm{i})}{\mathrm{HPcells\_gen}_{\mathrm{tag}}(\mathrm{i})}, \quad \forall \mathrm{i} \tag{7.70}$$

Using the formulas given by equations (7.54), (7.58) and (7.59) - for policing *Scenario 3* - and equations (7.65), (7.69) and (7.70) - for policing *Scenario 4*, two sets of experiments were carried out that considered the multiplexing of homogeneous On/Off modelled traffic sources. The traffic experiments are the same as those described in Section 7.4.2 and Section 7.4.3 for the validation of *Scenarios 3* and *4*, respectively. The details of the sources and systems used can be found in Table 7.6 and Table 7.7.

Fig.7.25 gives the results for the global cell loss probability obtained with both *Scenario 3* and *Scenario 4*, for a multiplexing of homogeneous low and high burstiness sources (taken separately), when the Leaky Bucket load in the policing *Test 0* (i.e., the test on the high priority traffic) is varied. The lines in the graph (see Fig.7.25) that contain the abbreviation "disc/HB" refer to results obtained for a multiplexing of high burstiness sources with policing *Scenario 3* (i.e., where discarding is considered to be the UPC action on violating high priority traffic). Similarly, the lines that contain the abbreviation "tag/LB" refer to results obtained with policing *Scenario 4* (i.e., where *Test 0* tags violating high priority traffic) for a multiplexing of homogeneous low burstiness sources.

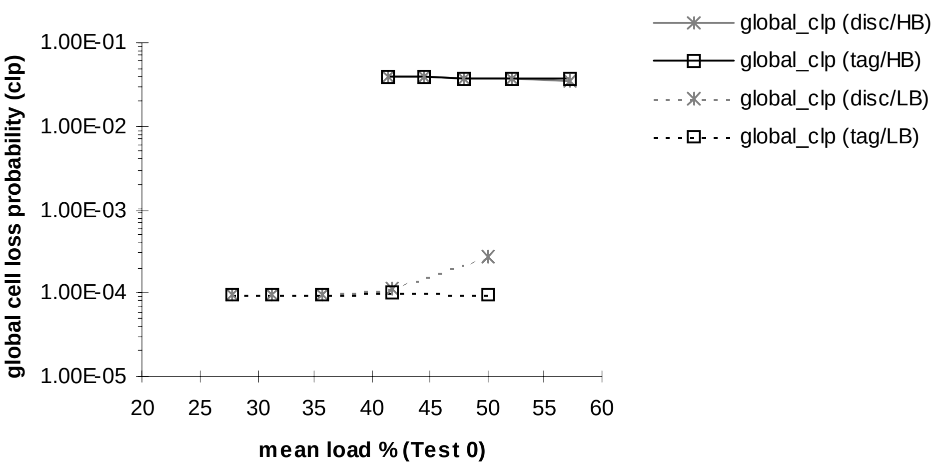
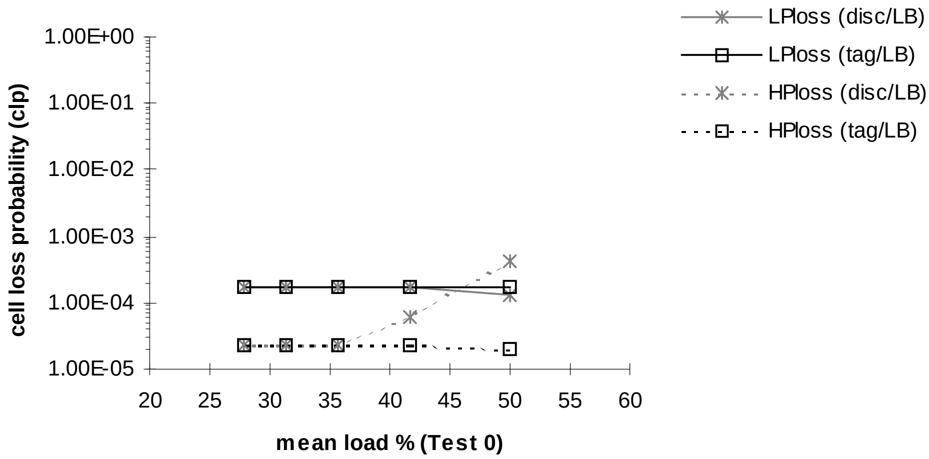
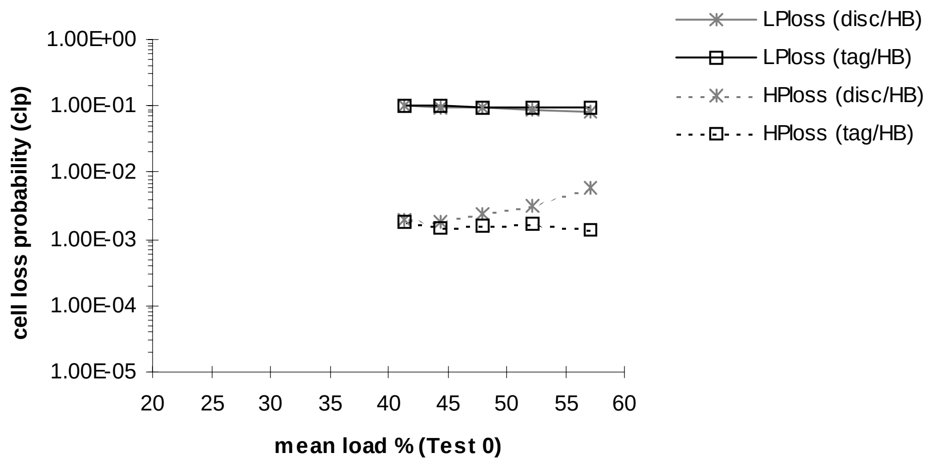


Fig.7.25 - Discarding *versus* tagging (for traffic with priorities): global cell loss probability for low and high burstiness sources.

The graph in Fig.7.25 shows that for a multiplexing of high burstiness sources, the global cell loss probability value remains almost constant for varying mean loads at policing *Test 0*. In this case, it can also be seen that the cell loss probability value obtained with policing *Scenario 3* and policing *Scenario 4* is virtually the same for increasing mean loads at *Test 0*. The similarity of the global cell loss probability results obtained with policing *Scenarios 3* and *4* can again be found, to a certain extent, with the multiplexing of low burstiness sources. However, for very high mean loads at *Test 0,* the global cell loss probability obtained for policing *Scenario 3* (under a multiplexing of high burstiness sources) becomes higher than the corresponding value for the policing *Scenario 4*.



(a) multiplexing of low burstiness sources



(b) multiplexing of high burstiness sources

Fig.7.26 - Discarding *versus* tagging (for traffic with priorities): LP cell loss and HP cell loss.

In Fig.7.26, results are given for the low and high priority loss probabilities under varying loads at policing *Test 0*. The results consider both the multiplexing of low and high burstiness sources for policing *Scenarios 3* and *4*. It can be seen that, for both graphs in Fig.7.26 (i.e., for both low and high burstiness sources), the use of tagging (i.e., the use of policing *Scenario 4*) appears to produce a reasonable improvement in the high priority cell loss that results in a very small degradation of the low priority cell loss. This is more noticeable in the case of high mean loads at the policing test on the high priority traffic. This indicates that the tagging option on the high priority traffic produces overall better results than using policing *Scenario 3*.

In conclusion, the observation of the results obtained in Fig.7.25 and Fig.7.26 indicates that, under the experimental conditions considered, the policing *Scenario 4* performs better overall, especially for high mean loads at the policing *Test 0*. It is also worth noticing that this does not depend on the burstiness level of the traffic sources used. However, this is a very limited study of a very wide subject. For a better understanding of the impact of the dimensioning of policing *Test 0* on the policing test for the global traffic, a deeper study is needed. For example, other experimental conditions should be considered, including the varying of parameters such as the queue's buffer size, the leak rate and maximum bucket size for policing *Test 0+1*, as well as the study of VC policing.

7.6. Processing Speed: Prioritised LINKSIM *versus* Policed LINKSIM

The implementation of the policing mechanism Leaky Bucket (for policing *Scenarios 1* to *4*, as described by Section 7.1) in the prioritised cell-rate simulator LINKSIM has brought more complexity to the already enhanced simulator. Indeed, when compared with the prioritised cell-rate simulator, the simulator with policing now has a source code with approximately 7200 lines (this represents an increase of about 30% over the prioritised version of LINKSIM) and an executable code size of the order of 300 Kbytes (i.e., an increase of about 36%).

Policing traffic with the policed LINKSIM implies additional calculations (apart from those necessary with the prioritised simulator) at the policing tests (i.e., at *Test 0* for high priority traffic and at *Test 0+1* for the global traffic, as seen in Section 7.1.4). Thus, it is expected that a reduction in the processing speed of the policed LINKSIM will become apparent when comparing it with the prioritised cell-rate simulator. This observation follows that in Section 5.5, where the processing speeds of both the prioritised cell-rate simulator and the original simulator were compared. It was then seen that there is a decrease in the processing speed of the prioritised cell-rate simulator.

Taking into account the extra complexity introduced in the prioritised cell-rate simulator when adding the implementation of the policing mechanism Leaky Bucket, it becomes necessary to investigate how much is really lost in processing speed when using the policing LINKSIM to study the policing of traffic with priorities. To this end, a set of traffic experiments with homogeneous On/Off modelled traffic sources was carried out where two network loads are considered. For each network load, the *On* and *Off* state durations of the sources are scaled so as to vary the burst length of the sources while maintaining the network load. Each traffic experiment is run both with the prioritised cell-rate simulator (always considering the high priority traffic to represent 30% of the total traffic) and with the policed simulator, under the same main experimental conditions. These concern,

- the number of sources used,
- the sources' traffic parameters,
- the proportion of high priority traffic to the total traffic, and
- the queue parameters,

so that the global load end-to-end (i.e., without considering policing) is the same with both LINKSIM versions. The experiments run with the policed simulator have fixed traffic parameters for the two policing mechanisms.

Table 7.8 shows the parameters used in the traffic experiments for both the prioritised and the policed cell-rate simulators, where *bucket_0* (*bucket_0+1*) and *leakrate_0* (*leakrate_0+1*) represent the maximum bucket size and the leak rate given to the policing *Test 0* (*Test 0+1*), respectively.

It is worth noticing that the traffic experiments described in Table 7.8 refer to VP policing. If VC policing had been considered, then the processing speed observed with the policed cell-rate simulator would be slightly lower. This is because more events have to be processed with VC policing (one event per active source when violating the traffic contract) than with VP policing.

Traffic Sources & System	
On state mean duration	2 ms to 1250 ms
Off state mean duration	18 ms to 11250 ms
peak bit rate	25000 cell/s
mean bit rate	2500 cell/s
burstiness	10
mean burst length	50 cells to 31250 cells
bucket_0, bucket_0+1	5, 10
leakrate_0, leakrate_0+1	120000 cell/s, 300000 cell/s
buffer length, threshold size	5 cells, 5 cells
queue capacity	200000 cell/s
network load	45% to 90%

Table 7.8 - Processing speed of policed cell-rate simulator: sources and system characteristics.

The graph in Fig.7.27 gives the results obtained for the speed increase when using the prioritised cell-rate simulator instead of the policed LINKSIM. It can be seen that up to reasonably high source mean burst lengths, both the prioritised simulator and the policed simulator have very similar processing speeds, for low and high network loads.

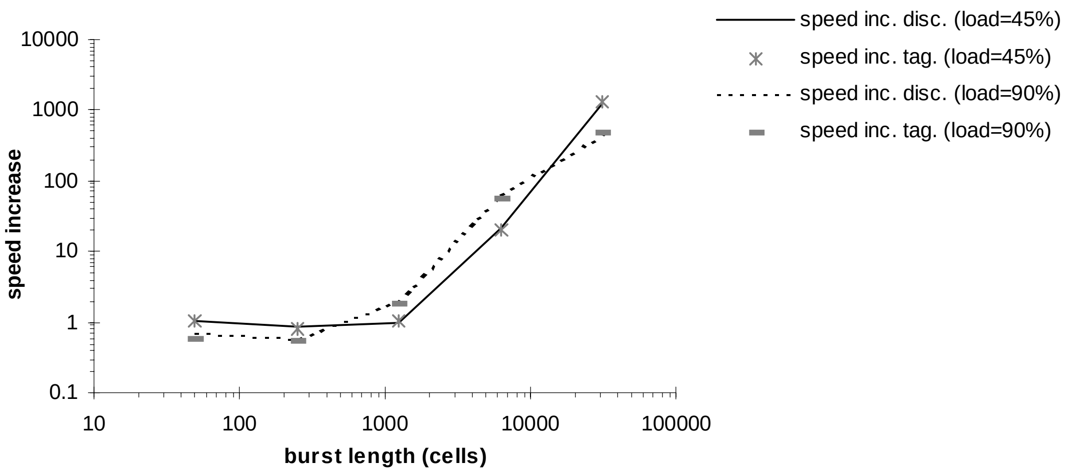


Fig.7.27 - Speed increase of prioritised cell-rate simulator *versus* source's mean burst length.

However, for very high burst lengths, the reduction in the processing speed of the policed simulator is quite noticeable, both for low and high network loads. This can be easily explained, especially for the case of long burst lengths and high network loads. Indeed, in this case, there is a high probability of having cell loss, thus implying both a great amount of processing in the queue and the two policing tests, and a longer overall processing time of the simulator.

Fig.7.27 also shows that the versions of the policed cell-rate simulator for tagging and discarding high priority traffic (labelled as *speed inc. tag.* and *speed inc. disc.*, respectively) have very similar processing speeds at both low and high network loads and for a wide range of mean burst length values.

In conclusion, and taking into account the results obtained in both Section 5.5 (in the comparison of the prioritised and original cell-rate simulators) and in this Section (for comparing the prioritised and policed simulators), it can be said that the policed cell-rate simulator is useful (i.e., it is reasonably fast) in the study of policing traffic with priorities when the traffic sources involved do not have very high mean burst lengths, regardless of the load they produce in the network.

7.7. Conclusion

The *possible UPC actions on misbehaving traffic* (i.e., discarding and tagging) and their *impact on traffic* in general were investigated in this Chapter. To that end, two *policing scenarios for traffic without priorities* and two other *scenarios for traffic with priorities were implemented* by the author in the prioritised cell-rate simulator described in Section 5.2. One of the scenarios for traffic without priorities considers the discarding of violating traffic and the other scenario tags all misbehaving traffic. The scenarios for traffic with priorities comprise two policing tests, one on the high priority traffic (which can be either tagged or discarded when found to be non-compliant) and another on the global traffic (this test always discards any violating traffic).

The UPC mechanism used was the *Leaky Bucket* and the implementation of the UPC mechanism in the prioritised cell-rate simulator had to be adapted to the burst level characteristic of the simulator. To the author's knowledge, this is the first application of cell-rate simulation to this problem.

The two *policing scenarios for traffic without priorities* implemented in the cell-rate simulator were *validated* for the particular case of one On/Off modelled traffic source with *On* and *Off* exponentially distributed state durations. The results obtained with the policed simulator were compared with a fluid flow analysis developed in [Yin91] and briefly described in Section 7.2.1. Both implemented scenarios for traffic without priorities proved to be fairly accurate under a wide range of experimental conditions, as seen in Sections 7.2.2 and 7.2.3.

A *validation process* was also carried out for the policing *scenarios using traffic with priorities,* which was described in Section 7.4.2 and Section 7.4.3. This is was possible by adapting (to the study of priorities) an upper bound analysis previously introduced in [Robe92, pp.150-152]. The validation considered the situation of multiplexing homogeneous On/Off modelled low burstiness sources. These policing scenarios were also tested for the case of multiplexing high burstiness sources and were seen to be fairly accurate.

The validation of the policing scenarios was followed by an evaluation of the policing scenarios' performance in terms of cell loss probability suffered by the input traffic. Again, different experimental conditions were considered to compare the global cell loss probability of the several policing scenarios, as well as the cell loss probability of high and low priority traffic at the system's queue.

The experiments carried out produced results for traffic without priorities that suggest the *use of the tagging scenario in situations of high traffic violations*; in this case, the reduction in the global cell loss probability is significant when compared with the discarding scenario. In other situations, the tagging scenario only produces slightly better results than the discarding scenario. Taking into account that the tagging of traffic implies the existence of some priority mechanism in the system and therefore a greater complexity of the network, tagging may not be worthwhile in situations of low traffic violations.

In the case of traffic with priorities, the values obtained appear to indicate that *tagging* the violating high priority traffic (i.e., using policing Scenario 4) produces overall better results for the low and high priority cell losses, especially in the case of *high mean loads at the policing Test 0*.

Finally, Section 7.6 addressed the evaluation of the reduction in the policed cell-rate simulator's processing speed, when compared with the prioritised simulator. Several traffic experiments were carried out under the same conditions using both versions of the cell-rate simulator. Although only VP policing was used to compare the processing speeds of the two simulators, it was observed that the *expected reduction in the processing speed when using the policed simulator* (due to the greater complexity introduced with the two policing tests) only becomes noticeable when in situations of traffic sources with very high values of mean burst length (in the order of about a thousand cells).

One final remark should be made that concerns the traffic experiments carried out with the policing scenarios and the suggestions for use of the policing scenarios. It has to do with the fact that no VC policing was considered for the scenarios using traffic with priorities. Moreover, the experiments only considered the policing of one traffic source for policing *Scenarios 1* and *2*. Therefore, the conclusions presented for the performance of the policing scenarios studied should be interpreted with *caution* since they are mere indicators of what will happen in the case of system configurations handling multiple (heterogeneous) sources with/without priorities.

8. Application to an ATM Testbed

In previous Chapters, a description was given of the theoretical and experimental work developed by the author. The traffic experiments that were then introduced and carried out considered both artificial traffic (in the sense that traffic source models were used to represent the behaviour of real services) and either analytical methods or software simulation tools. These methods have the disadvantage of, in most cases, not taking into consideration important network variables such as the presence of Operation and Maintenance (OAM) cells in the traffic circulating in real networks or the processing delays inherently associated with network switches. On the other hand, the use of analytical and especially simulation methods will be beneficial, whether there is a real network available or not. In the second situation, simulation methods are useful in the sense that they can predict how a network will behave, given certain characteristics. When there is a real network operating, a network simulator is still very useful. In this case, the simulator can primarily be validated by comparison with the real network. Once the network simulator has been validated, it can then again be used as a predictor, by incorporating new network features and evaluating the network behaviour under specific conditions, such as a scaling of the network size in terms of load and customers.

While there are no real ATM networks operating, an intermediate stage would be to have a so called *testbed*, which would be mainly hardware based (as real ATM networks will be) and providing some basic services like voice and data, whilst still considering also some artificial traffic sources. This is the case of the ATM testbed considered and used in this Chapter.

Both simulators and ATM testbeds have disadvantages, some of which have been indicated in the beginning of this Chapter. Another drawback of using simulators concerns the almost systematic use of pure ATM interfaces to represent the line transmission technology; this is a limitation, since most existing testbeds consider transmission technologies such as Synchronous Digital Hierarchy (SDH), which is already standardised for a maximum transmission rate of 155.52 Mbit/s (see [Cuth93, pp.33-34] and [Sait94, pp.5-8], respectively). As far as testbeds are concerned, and although they cater for the existence of network variables like signalling and the existence of OAM cells (thus producing results which are closer to reality concerning the users' perceived QoS for given services), they have also some limitations, such as the degree of accuracy of the artificial traffic generators used.

This Chapter compares results from the cell-rate simulator used throughout the thesis with an ATM testbed. The experiments described aim at validating the prioritised simulator thus confirming it as a useful predictor. Moreover, the study of the results points out to some traffic aspects that can lead to future improvements in the cell-rate simulator.

The testbed used is that belonging to the RACE project EXPLOIT in which QMW is a partner. Located in a Swiss PTT building in Basel, it provides one of the most extensive ATM test localities in Europe.

8.1. Experiments without Priorities

The experiments carried out in this Section concern traffic where cells do not have an assigned priority level. The aim of the experiments is to evaluate the accuracy of the cell-rate simulator LINKSIM described in Chapter 5 by comparing its outputs with results obtained in an ATM testbed. To this end, two main types of artificial traffic sources are considered in the experiments: On/Off and GMDP modelled sources (see Section 4.1.3 for a description of these source models). The *On* and *Off* states of the On/Off sources used are assumed to be exponentially distributed, while each state of the GMDP sources considered is taken to be geometrically distributed.

The raw data used to reproduce the testbed experiments with the cell-rate simulator have been extracted from two documents (see [EXPL94b] and [EXPL94c]) produced by the research project EXPLOIT, under the European research programme Research in Advanced Communications in Europe (RACE). The equipment used to execute the traffic experiments in the ATM testbed included artificial traffic generators, traffic analysis tools and network switches.

Experiments for homogeneous and heterogeneous traffic scenarios are treated separately in Sections 8.1.1 and 8.1.2, respectively.

8.1.1. Homogeneous Scenarios

The first set of experiments carried out for this type of scenario considers a mixture of 3 On/Off sources with the same traffic parameters, but where the mean *Off* state duration of the sources is progressively increased; the traffic mixture is multiplexed in an ATM queue (i.e., a network switch) with a fixed service rate and buffer size. This allows an evaluation of how well the cell-rate simulator reacts to changes in the experimental conditions. The next Table (Table 8.1) gives the traffic parameters associated with the On/Off sources for each experiment, as well as the multiplexer's characteristics.

	On state		***Off state***
exp#	peak bit rate	mean burst length	mean silence length
1	55 Mbit/s	100 cells	100 cells
2	55 Mbit/s	100 cells	250 cells
3	55 Mbit/s	100 cells	500 cells
4	55 Mbit/s	100 cells	1000 cells
5	55 Mbit/s	100 cells	7500 cells
buffer size		27 cells	
link capacity		103.68 Mbit/s	

Table 8.1 - Traffic parameters for experiments with different mean silence periods (network load: 80% down to 2%).

It should be mentioned that although the original experiments considered a mixture of On/Off sources (under the same conditions as described in the previous paragraph), another type of traffic, CBR, was also included (see [EXPL94b]). In order to adapt the experiments so that no CBR traffic would be considered, a scaling of the available link bandwidth had to be performed. Since CBR traffic requires a fixed portion of bandwidth to be carried by a network, the final link bandwidth value was obtained by subtracting the bit rate required by the CBR traffic source to the original bandwidth value of 155.52 Mbit/s. The results obtained by repeating the traffic experiments with the cell-rate simulator described in Chapter 5 are shown in Fig.8.1. The two plotted lines, *clp (testbed)* and *clp (simulation)*, refer to the cell loss results obtained in the testbed and with the simulator, respectively; the simulated results include confidence intervals for each experiment. Note also that the cell loss results given in the graph represent averaged values for all the traffic sources involved.

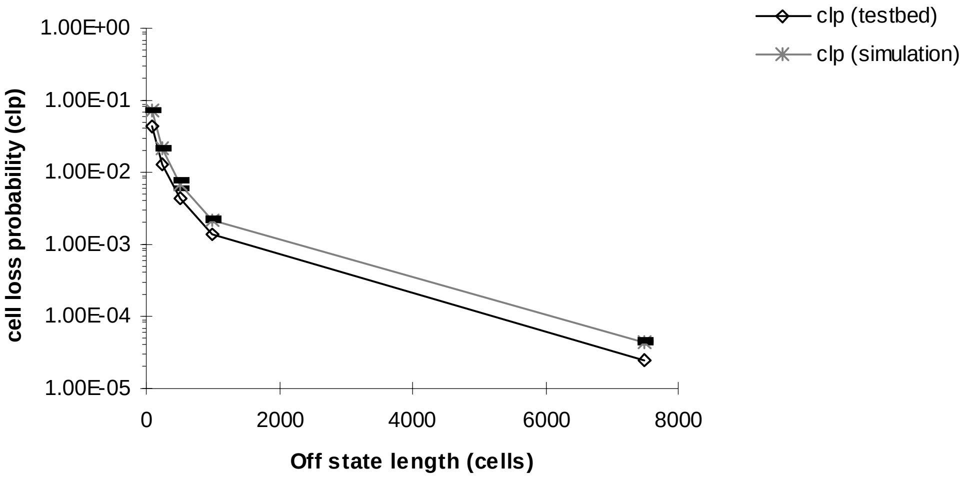


Fig.8.1 - Simulation *versus* ATM testbed: varying the sources' mean silence duration (network load: 80% down to 2%).

The graph (see Fig.8.1) shows that the cell-rate simulator slightly overestimates the sources' cell loss when compared with the cell loss results obtained in the testbed. This is to be expected, since the traffic analysis tools used in the testbed work at cell level, while the simulator operates at burst level.

	On state		***Off state***
exp#	peak bit rate	mean burst length	mean silence length
1	55 Mbit/s	1500 cells	1500 cells
2	55 Mbit/s	1500 cells	2500 cells
3	55 Mbit/s	1500 cells	5000 cells
4	55 Mbit/s	1500 cells	7500 cells
5	55 Mbit/s	1500 cells	10000 cells
buffer size		27 cells	
link capacity		124.416 Mbit/s	

Table 8.2 - Traffic parameters for experiments with different mean silence periods (network load: 66% down to 12%).

Note however that the cell loss overestimation always amounts to far less than one order of magnitude.

The second set of experiments again considered 3 On/Off modelled sources feeding a queue with the same buffer size taken in the previous set of experiments, but where the available link bandwidth is now greater than before. Once more, the *Off* state duration was varied throughout the several experiments, while maintaining the mean burst length of the sources. The sources' peak bit rate was also maintained. Table 8.2 gives the traffic sources' characteristics for each experiment, along with the buffer size and link bandwidth used.

In the next graph (Fig.8.2), cell loss results obtained with the cell-rate simulator are compared with the results from the ATM testbed. The two lines plotted represent the same type of data described for the results shown in Fig.8.1. The cell loss values obtained in the two different environments are very close. Again, the simulator overestimates the cell loss, but the overestimation becomes smaller for increasing *Off* state durations (i.e., for decreasing network loads). This gives an indication that the cell-rate simulator performs better for low network loads.

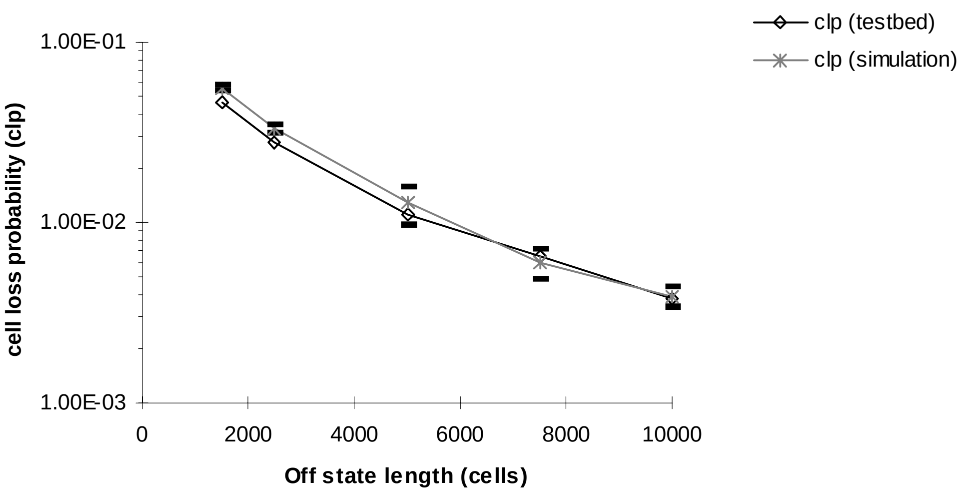


Fig.8.2 - Simulation *versus* ATM testbed: varying the sources' mean silence duration (network load: 66% down to 12%).

The next three sets of experiments described in the following have been taken from the document [EXPL94c]. Again, several traffic sources (modelled as On/Off sources in the first two cases and as GMDP sources in the last situation) are statistically multiplexed and a measurement of the resulting network performance is taken in terms of cell loss ratio. The same document uses the results thus obtained to investigate the ability of different Connection Admission Control (CAC) algorithms to control specific network performance objectives. This means that the cell-rate simulator described in this thesis can also be a useful tool to execute identical studies, provided it proves to be sufficiently accurate.

Traffic Sources & System	
On state mean duration	20 ms
Off state mean duration	80 ms
peak bit rate	31.1 Mbit/s
buffer size	48 cells
link capacity	155.52 Mbit/s
number of sources	8 to 16 (increments of 2)

Table 8.3 - Traffic parameters for experiments with an increasing number of multiplexed sources (network load: 32% to 64%).

In the first two sets of experiments that follow, the peak bit rate of the traffic sources, as well as the available link capacity and the buffer size of the queue are maintained. On the other hand, both the *On* and *Off* mean durations of the traffic sources and the number of sources multiplexed are varied.

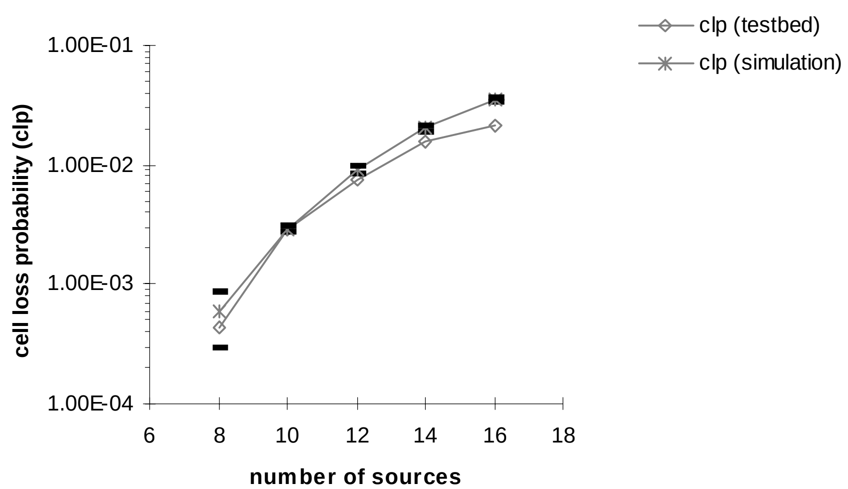


Fig.8.3 - Simulation *versus* ATM testbed: varying the number of sources multiplexed (network load: 32% to 64%).

Table 8.3 describes the parameters of the traffic sources used in the several experiments of the first set.

The cell loss results obtained with the cell-rate simulator (including the confidence intervals for each experiment) can be seen in Fig.8.3. This graph again shows that the cell-rate simulator slightly overestimates the cell loss, when compared with the testbed results. The cell loss overestimation appears to be more noticeable when the number of sources increases (i.e., for increasing network loads).

Traffic Sources & System	
On state mean duration	10 ms
Off state mean duration	190 ms
peak bit rate	31.1 Mbit/s
buffer size	48 cells
link capacity	155.52 Mbit/s
number of sources	18 to 34 (increments of 4)

Table 8.4 - Traffic parameters for experiments with an increasing number of multiplexed sources (network load: 18% to 34%).

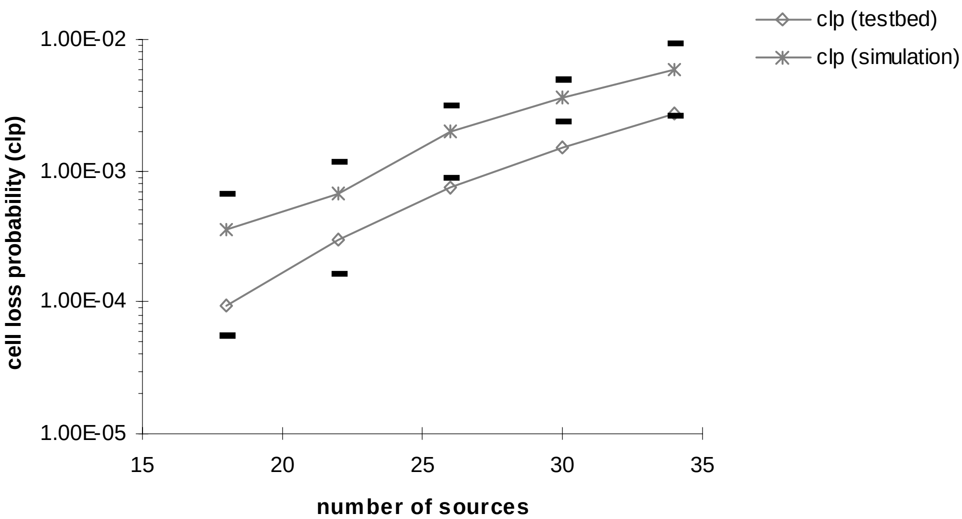


Fig.8.4 - Simulation *versus* ATM testbed: varying the number of sources multiplexed (network load: 18% to 34%).

For the second set of experiments, similar traffic sources to that of Table 8.3 are considered, except for the number of sources multiplexed and the proportion of time for which the traffic sources are active (20% in the first case and 5% in this case). Table 8.4 contains the details. It is worth noticing that this set of experiments considers very low network loads. The results obtained for the network and source configurations described in Table 8.4 are plotted in Fig.8.4. The graph (see Fig.8.4) shows that the cell-rate simulator again overestimates the cell loss, but this time to a greater extent. Why there is this increase is not clear. Finally, another set of experiments was considered for 3-state GMDP modelled traffic sources.

Traffic Sources & System		
state	mean duration	bit rate
1	70 ms	3.11 Mbit/s
2	20 ms	6.22 Mbit/s
3	10 ms	31.10 Mbit/s
buffer size	48 cells	
link capacity	155.52 Mbit/s	
number of sources	10 to 16 (increments of 2)	

Table 8.5 - Traffic parameters for experiments with 3 state GMDP traffic sources (network load: 42% to 67%).

For this set of experiments, the number of sources was varied, therefore changing the network load. The cell loss results shown by Fig.8.5 (which includes confidence intervals for each of the experiments) indicate a good accuracy of the cell-rate simulator for moderate network loads.

It should also be mentioned that the ATM testbed results relative to this set of experiments had to be extrapolated, in order for a comparison to be possible between the two different environments. More specifically, the total number of sources used in each experiment was divided into two sets of equal size, which were analysed separately by two traffic analysis tools. However, in some cases (including the one now being compared with the cell-rate simulator), it was not possible to measure the cell loss for one of the groups of sources.

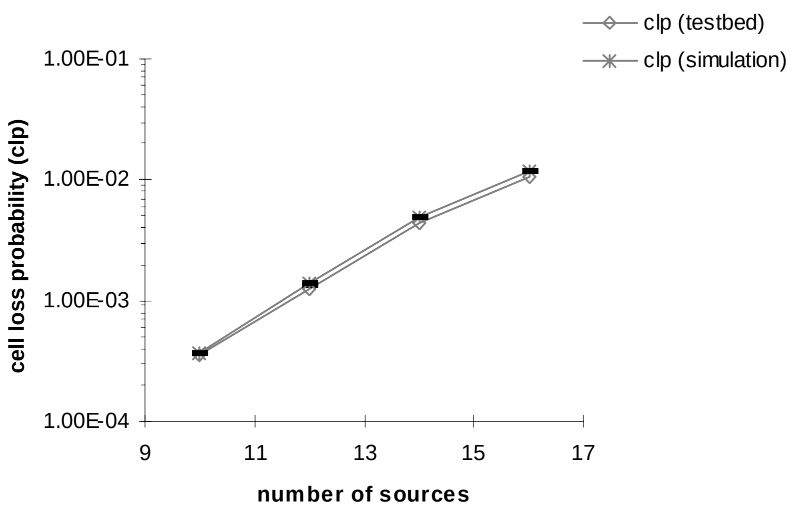


Fig.8.5 - Simulation *versus* ATM testbed: 3 state GMDP traffic sources (network load: 42% to 67%).

For these cases, it was still possible to compare the two environments. In fact, if it is assumed that all sources having the same characteristics (defined by the source type) also suffer the same cell loss ratio, then it is possible to take a certain percentage of sources from each class and only use those for the measurements. This can be seen with an example.

Consider two traffic classes, named $class_1$ and $class_2$, and for each class, N_1 and N_2 traffic sources of that class, respectively. Assume also that the cell loss ratio of each traffic class is known and denoted by clr_1 and clr_2 (where these two values may be different). Then, the global cell loss ratio is calculated as,

$$\frac{N_1 \cdot clr_1 + N_2 \cdot clr_2}{N_1 + N_2}. \tag{8.1}$$

If it is necessary to measure this overall cell loss ratio by considering say, only 50% of the total number of sources from each class, then the global cell loss ratio will be,

$$\frac{0.5 \cdot N_1 \cdot clr_1 + 0.5 \cdot N_2 \cdot clr_2}{0.5 \cdot N_1 + 0.5 \cdot N_2} = \frac{N_1 \cdot clr_1 + N_2 \cdot clr_2}{N_1 + N_2} \tag{8.2}$$

which shows that it is possible to measure the overall cell loss ratio by just taking a proportion of the total traffic for the measurement (the same reasoning will also be applied to the sets of experiments considered for Section 8.1.2). Note also that other percentages could have been chosen to show the same result, provided the percentage taken is the same for all traffic classes. Moreover, the experiments' settings (such as the link bandwidth and number of sources) must not be scaled, as the system's behaviour is not linear.

In conclusion, the results obtained for the experiments with homogeneous non-prioritised traffic appear to show that, in general, the cell-rate simulator is fairly accurate when compared with the more realistic environment of an ATM testbed.

8.1.2. Heterogeneous Scenarios

The consideration of different types of traffic in simulation experiments is advantageous, both in terms of the closeness to reality (since the main idea of broadband networks is to accommodate a wide range of traffic types) and in terms of investigating the flexibility of a given simulator. Bearing this in mind, a few sets of experiments previously realised at an ATM testbed (see [EXPL94c]) have been repeated with the cell-rate simulator LINKSIM; they consider the multiplexing of two different types of traffic sources into an ATM buffer and evaluate the total cell loss obtained. For these experiments, only On/Off modelled sources were used, with different traffic parameters. All experiments considered the same available link bandwidth (155.52 Mbit/s in this case) and a buffer size of 48 cells (the buffer size of the network switch used); also, all experiments vary the number of sources of one type, while maintaining the number of sources of the other traffic type.

***Type A* --** peak bit rate = 31.1 Mbit/s		***Type B1* --** peak bit rate = 7.78 Mbit/s	
state	duration	state	duration
On	20 ms	On	50 ms
Off	80 ms	Off	50 ms
no. of sources	4A+6B, 4A+8B, 4A+12B, 4A+16B, 4A+20B, 4A+24B		

Table 8.6 **-** Traffic parameters for experiments with increasing number of On/Off multiplexed sources (network load: 31% to 76%).

Consider the first set of experiments, for which the sources' traffic parameters are described in Table 8.6. The comparison of the cell loss results obtained in the ATM testbed and with the cell-rate simulator shows a good accuracy of the simulator over a relatively wide range of network loads, hereby translated in the number of *type B1* sources (see Fig.8.6). The plotted lines, *ETB loss* and *Sim. loss*, represent the testbed results and the simulated results, respectively. The number of sources indicated in the horizontal axis of the graph corresponds to the total number of sources. However, it should be noted that each of those values corresponds to a sum of four *type A* sources and a varying number of *type B1* sources. So, for an increasing number of *type B1* sources, the cell loss increases (although not linearly) as would be expected.

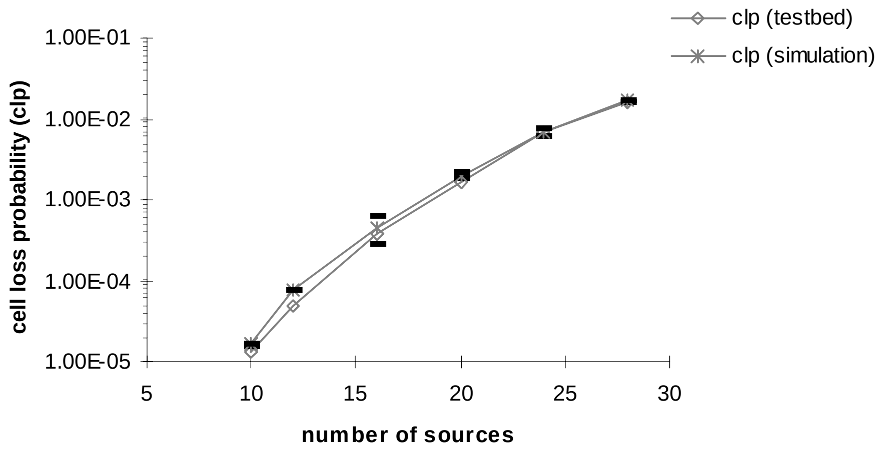


Fig.8.6 - Simulation *versus* ATM testbed: increasing the number of sources multiplexed (network load: 31% to 76%).

The next set of experiments uses a slightly different mix of traffic sources, as given by Table 8.7. Here, it can be seen that the peak bit rate of the sources was maintained, as well as the *On* and *Off* average state durations for traffic *type A*.

***Type A* -- peak bit rate = 31.1 Mbit/s**		***Type B2* -- peak bit rate = 7.78 Mbit/s**	
state	duration	state	duration
On	20 ms	On	10 ms
Off	80 ms	Off	190 ms
no. of sources	4A+60B2, 4A+80B2, 4A+100B2, 4A+140B2, 4A+160B2		

Table 8.7 - Traffic parameters for experiments with increasing number of On/Off multiplexed sources (network load: 31% to 56%).

The changes occur in the proportion of time that *type B2* traffic is now active. For this set of experiments, the cell loss overestimation that was almost non-existent in the previous example is now noticeable, although always within at most one order of magnitude of the testbed results. The confidence intervals shown also indicate that the cell-rate simulator produces more realistic results for an increasing number of *type B2* sources (i.e., increasing network loads).

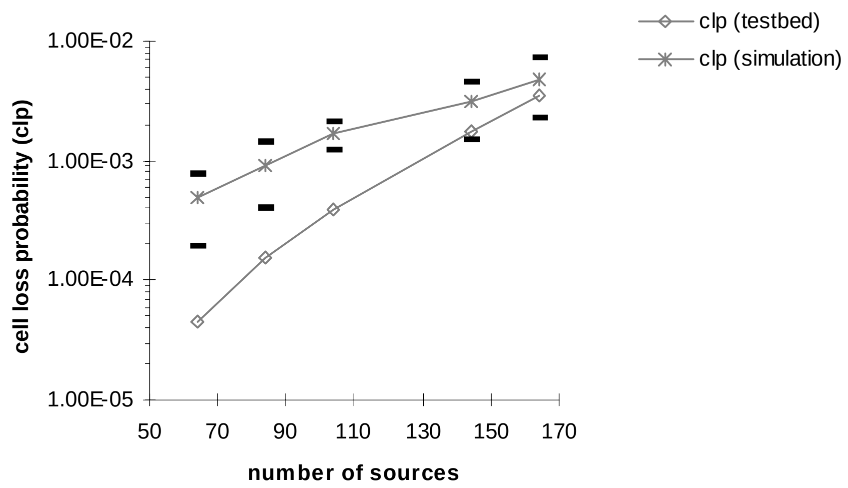


Fig.8.7 - Simulation *versus* ATM testbed: increasing the number of sources multiplexed (network load: 31% to 56%).

The next set of experiments is very similar to the previous one; the difference is that now, the fixed number of *type A* sources considered has been reduced to 2, while the number of *type B2* sources is further increased. Table 8.8 gives the combination of sources used in each experiment; the details of each traffic source type can be found in Table 8.7.

	exp#1	exp#2	exp#3	exp#4	exp#5
Number of Sources	2A + 130B2	2A + 150B2	2A + 170B2	2A + 190B2	2A + 230B2

Table 8.8 - Combination of sources (network load: 41% to 66%).

The results shown by Fig.8.8 are, as expected, analogous to those of Fig.8.7. This appears to show that no significant differences occur in the obtained overall cell loss when the number of sources of one traffic type is decreased while increasing the number of sources of another traffic type, provided the overall network load remains approximately the same.

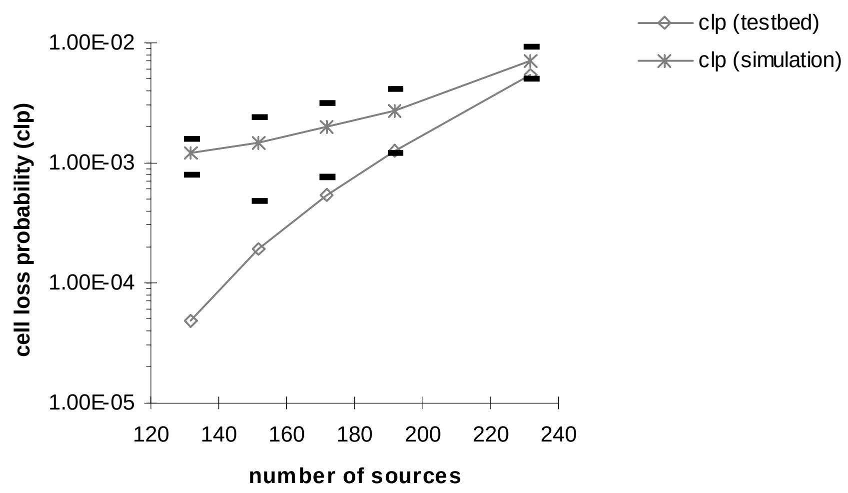


Fig.8.8 - Simulation *versus* ATM testbed: increasing the number of sources multiplexed (network load: 41% to 66%).

The last set of experiments considered in this Section is configured in Table 8.9. Again, a fixed number of *type A* sources (4, in this case) is combined with a varying number of sources of *type C*, which has a very low bit rate when compared to traffic *type A*.

Type A **--** peak bit rate = 31.1 Mbit/s		***Type C*** **--** peak bit rate = 1.94 Mbit/s	
state	duration	state	duration
On	20 ms	On	50 ms
Off	80 ms	Off	50 ms
no. of sources	4A+32C, 4A+40C, 4A+50C, 4A+60C, 4A+70C, 4A+80C, 4A+100C		

Table 8.9 - Traffic parameters for experiments with increasing number of On/Off multiplexed sources (network load: 36% to 78%).

As seen from Fig.8.9, there is a good agreement between the ATM testbed results and the cell-rate simulator's results. Notice however the slight discrepancies in the curve corresponding to the testbed, labelled as *clp (testbed)*; these give an indication that the testbed can produce results that are not completely accurate (this was already referred to in the beginning of this Chapter).

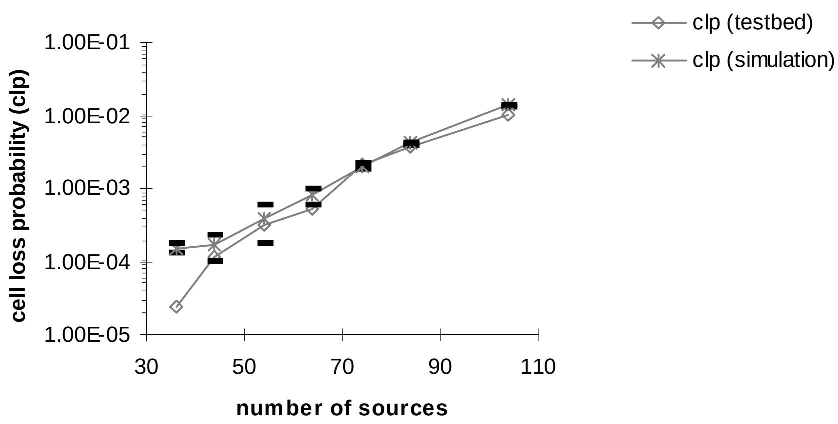


Fig.8.9 - Simulation *versus* ATM testbed: increasing the number of sources multiplexed (network load: 36% to 78%).

In view of the results obtained for the several experiments with heterogeneous traffic described in this Section, it can be said that the cell-rate simulator appears to behave fairly well in diverse circumstances, such as different bit rate sources and for a wide range of network loads. All the experiments show the slight overestimation of the cell loss when using the simulator, but this overestimation is within one order of magnitude for most cases. Also, the simulator's accuracy appears to be better for increasing network loads. This was not always verified for the homogeneous traffic scenarios considered in Section 8.1.1.

8.2. Experiments to Predict the Behaviour of Prioritised Networks

An attempt is made in this Section to evaluate whether or not the prioritised cell-rate simulator is capable of reproducing the behaviour of a network link when traffic with assigned priority levels is considered. To this end, a comparison has been made between the prioritised cell-rate simulator and the ATM testbed used in previous Sections. This was possible by altering some of the testbed features, since no priority mechanism is usually available in the testbed. The alterations involved emulating the exact version of Partial Buffer Sharing mechanism (see Section 3.8.2.2) in a piece of hardware equipment, initially designed to be a policing unit.

The main purpose of this policing unit is to control the traffic in the network and make sure that users do comply to their respective traffic contracts (see Section 3.6.2); in this respect, it can be set up to take actions on non compliant traffic (cell discarding, in this case) for different peak bit rates. Taking into account the features of the policing unit, a low level programming was performed (see [Wins95]) in order to have two distinct traffic streams multiplexed (one for low priority traffic and another one for high priority traffic) and then analysed by a priority mechanism.

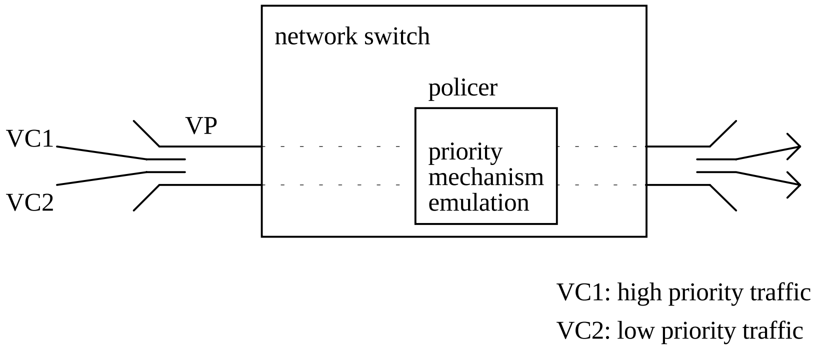


Fig.8.10 - Priority scenario in the ATM testbed.

The diagram in Fig.8.10 shows how the prioritised traffic was analysed by using VP policing; due to hardware restrictions, different VCs had to be considered to carry the low and high priority traffic streams. Therefore, only a very simple scenario was considered for which the sources' traffic parameters are given in Table 8.10.

	Low priority traffic		***High priority traffic***	
	peak bit rate = 10 Mbit/s On duration = 2.12 ms		peak bit rate = 12 Mbit/s On duration = 2.12 ms	
	Off duration		*Off duration*	
	time	cells	time	cells
exp#1	2.12 ms	50	2.12 ms to 8.48 ms *	60 to 240
exp#2	4.24 ms	100	2.12 ms to 8.48 ms *	60 to 240
exp#3	6.36 ms	150	2.12 ms to 8.48 ms *	60 to 240
exp#4	8.48 ms	200	2.12 ms to 8.48 ms *	60 to 240
exp#5	10.6 ms	250	2.12 ms to 8.48 ms *	60 to 240

Table 8.10 - Configuration of the experimental scenario for traffic with priorities (*increments of 2.12 ms).

All the experiments assumed an available link bandwidth of 10 Mbit/s (which effectively represents the policing rate) and a buffer size of 27 cells with a threshold at 17 cells, above which only high priority traffic is accepted by the queue.

The results obtained with both the exact priority mechanism emulated in the testbed and the prioritised cell-rate simulator (that considers an approximated priority mechanism) plot the cell loss for fixed *On* and *Off* state durations of the low priority traffic source and varying *Off* state durations of the high priority traffic (see Fig.8.11). A similar comparison can be found in Fig.8.12, by varying the *Off* state duration of the low priority traffic source instead. Graphs (see Fig.8.11 and Fig.8.12) are shown only for two particular experiments, since similar results were seen with the remaining experiments.

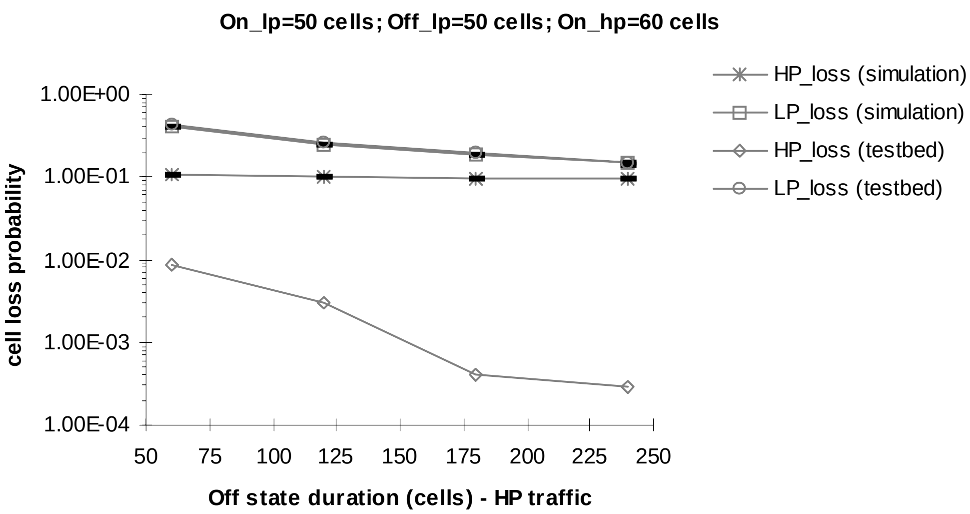


Fig.8.11 - Prioritised simulation *versus* ATM testbed: varying the *Off* state duration of the high priority traffic.

The results shown in the first graph (see Fig.8.11) (along with confidence intervals for the simulated experiments) confirm what had already been proven by using an analytical method (see Section 6.4.2), i.e., that the prioritised cell-rate simulator gives accurate measures for the low priority cell loss and it overestimates the high priority cell loss. The high priority cell loss overestimation is approximately of one order of magnitude. This has been seen for all the other experiments indicated in Table 8.10. Notice also that the high priority cell loss obtained with the prioritised simulator appears to be invariant to changes in the *Off* duration of the high priority traffic; this is partly explained by the mode of operation of the approximate priority mechanism which only loses high priority traffic when its rate combines to more than the service rate of the queue. A future study should however be conducted to investigate the true reasons for this occurrence. Similar considerations can be taken by observing the next graph (see Fig.8.12), that plots the cell loss for low and high priority traffic streams against varying *Off* state durations of the low priority traffic.

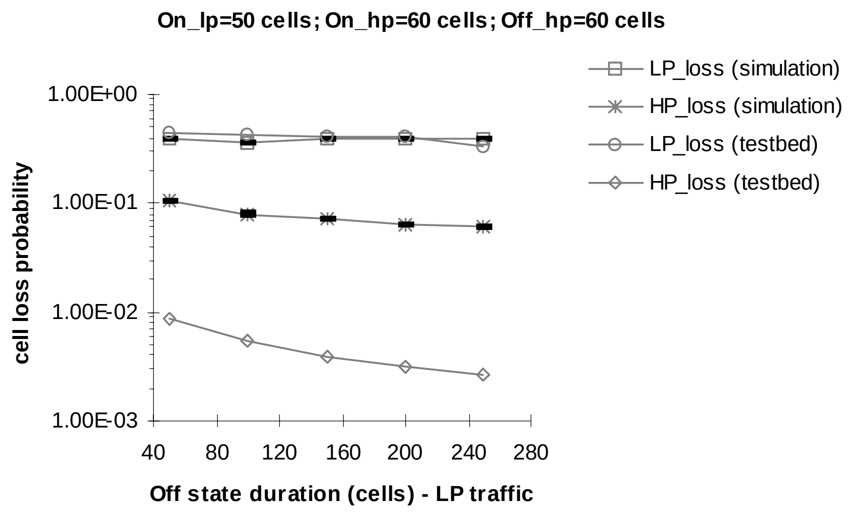


Fig.8.12 - Prioritised simulation *versus* ATM testbed: varying the *Off* state duration of the low priority traffic.

In conclusion, it can be said that the simulator accurately measures the cell loss for low priority traffic, while overestimating the high priority traffic cell loss. It is also worth noticing that the simulator had already been compared with the testbed, for the case of traffic without priorities; it was then seen that the simulator reproduced quite accurately the behaviour of the testbed. However, in view of the discrepancies in the high priority cell loss results, further investigation, comprising mainly a wider variety of traffic experiments and scenarios, should be undertaken in the future. These would be time-consuming since it would be necessary to match the testbed conditions as near as possible to those modelled by the simulator.

8.3. Conclusion

The aim of this Chapter was to be the final proof of whether the cell-rate simulator is or is not useful (and under which circumstances) as a predictor of networks where traffic with priorities is considered. Several experiments carried out in an ATM testbed and afterwards repeated with the cell-rate simulator were considered for two main situations:

1. multiplexing of traffic without priorities;
2. multiplexing of prioritised traffic.

The results obtained for the first situation indicate a good agreement between the cell-rate simulator and the testbed. Although the simulator slightly overestimates the results, this is always within one order of magnitude. Also, the simulator performs better for increasing network loads when in the case of a heterogeneous traffic scenario, while the opposite happens for homogeneous scenarios.

On the experiments carried out for traffic with priorities, two observations should be made that may explain the discrepancies verified in the high priority traffic results:

1. only one traffic scenario was considered, due to hardware restrictions;
2. not many traffic sources were considered, which implies obtaining a small statistical gain.

Although the high priority traffic results are overestimated with the cell-rate simulator, the low priority traffic cell loss results are very accurate, when compared with the results from the testbed. In view of the two observations above, it is fair to conclude that more experiments (with a wider range of low and high priority traffic sources) need to be carried out, in order to correctly evaluate the usefulness of the prioritised simulator.

It therefore appears that further work should be done in order to better establish the boundary conditions under which the simulator can be used as an accurate tool for traffic with priorities and whether the threshold approximation is valid.

9. Discussion

The assessment of the work described in this thesis encompasses two main points. The first one concerns an evaluation of the work carried out. The other point that will be analysed refers to what could be added to the research work developed in order to improve it.

Traffic control mechanisms - namely *Priority Control* and *Usage Parameter Control* (UPC) - and *cell-rate simulation* were the main areas of research addressed in this thesis. Concerning priority control, a cell-rate simulator initially developed by Pitts (see [Pitt93]) was used to implement a space (or loss) priority mechanism (see Section 5.2). Cell-rate simulation has been used as it leads to faster simulations of rare events in complex ATM networks and until now, only cell level simulation had been used to study traffic with priorities. The use of cell-rate simulation also made possible a simplification in the implementation process of the chosen priority mechanism, i.e., Partial Buffer Sharing (PBS). The simplification consisted of considering the threshold of a queue to be its full size. In fact, with an exact PBS mechanism, the system's queue is virtually partitioned into two parts: one part that admits both low and high priority traffic and the space above the threshold that only accepts high priority traffic (see Fig.3.9 and Section 3.8.2.2).

The choice and use of the PBS mechanism from the available priority mechanisms relied on previous research work by other authors (see for example [Kang93], [Meye93] and [Elwa94]). When compared with other priority mechanisms, the PBS mechanism is thought to be "... the only likely candidate for implementation ..." (see [Grav91b]), due to its compromise between simplicity and accuracy.

The implementation of the priority mechanism was followed by a validation process. This involved a comparison with simulation results by Kröner (see [Krön91] and Section 5.3.1), as well as comparisons with two analytical methods. One of the analytical methods gives an upper bound to calculate the cell loss of a system with two types of prioritised traffic (see [Krön91] and Section 5.3.2). The other analytical method considers the modelling of one On/Off source only. It represents an extension by the author, to the case of priorities of the model (an exact fluid flow analysis) developed by Schormanns (see [Scho94] and Section 6.3). The closed-form formulas for cell loss probability given by the extended model can also be used to determine the equivalent capacity required for sources that make use of space priority. The validation process (see Section 6.4) confirmed the accuracy of the priority mechanism implemented in the cell-rate simulator referred to above, especially concerning the cell loss probability obtained for low priority traffic. The cell loss results obtained for high priority traffic show that the cell-scale behaviour (not modelled by the cell-rate simulator) is an important component of queuing, particularly when the proportion of high priority traffic is low.

The implementation of a space priority mechanism in the cell-rate simulator LINKSIM increased the complexity of the simulator, which can be translated into a reduction of the processing speed when using the prioritised cell-rate simulator. This speed reduction was quantified by running traffic experiments with both versions of the cell-rate simulator (see Section 5.5). The analysis of the results obtained showed that the prioritised cell-rate simulator has a similar processing speed to that of LINKSIM for traffic without priorities, provided the sources involved do not have very high values of mean burst length (up to a few thousand cells) and in the case of moderate network loads.

UPC is a monitoring function used in ATM networks "... to monitor and control traffic, in terms of traffic offered and validity of the ATM connection at the user access network ..." (see [ITU94]). In other words, any traffic that is analysed by the UPC and is found to be violating its traffic contract established at connection set-up suffers penalties. These can be translated in the discarding or marking of cells (see Section 3.6.2.3).

The possible UPC actions on violating traffic are the connecting point between the study of priority control and policing in this thesis. In fact, the tagging of cells as the chosen UPC action on violating traffic involves handling traffic with priorities. This is because tagging cells corresponds to no more than setting high priority cells to be of low priority (note that all traffic is, by default, of high priority). Therefore, when the UPC action on violating traffic is set to be "tagging cells", it is important to investigate the effects that it has on the QoS requirements of traffic (of both high and low priority, if any). The research work reported in this thesis addressed this problem with the aim of not only identifying the impact of using either UPC action, but also suggesting when to use each of the possible UPC actions without significantly compromising any QoS requirements (see Chapter 7).

Evaluating the performance of the UPC function involved the implementation (see Section 7.1) by the author of four policing scenarios, two of which concern traffic without priorities that is either discarded or tagged when found to be non-compliant. These two policing scenarios were implemented in the prioritised cell-rate simulator referred to above and afterwards validated with an approximate fluid flow analysis developed by Yin (see [Yin91] and Section 7.2.1) for the particular case of one On/Off source being policed (see Section 7.2). The study of the performance evaluation for the two implemented policing scenarios using traffic without priorities indicates that the tagging of non-compliant cells by the UPC should be applied when in the situation of traffic with a high violation percentage (i.e., traffic whose generated input rate exceeds the corresponding declared peak bit rate). In other cases, the discarding of non-compliant cells is a better overall solution (see Section 7.3). The other two policing scenarios implemented in the prioritised cell-rate simulator concern traffic with priorities, that is monitored in two stages:

- a *first policing test* (named *Test 0*) that only takes action on any violating high priority traffic in the system;
- a *second policing test* (called *Test 0+1*) that considers the updated global traffic (after the operation of *Test 0*) and takes action on any violating traffic (be it of low or of high priority).

The policing *Test 0* can either discard or tag violating high priority traffic, while *Test 0+1* always discards any violating traffic.

The validation of the two policing scenarios for traffic with priorities involved the use of an approximate bufferless analysis for homogeneous On/Off sources, introduced in [Robe92, pp.150-152] and adapted by the author to study traffic with priorities (see Section 7.4.1). The traffic experiments carried out to evaluate the performance of these policing scenarios indicate that the scenario which tags violating high priority traffic performs better overall, especially for high mean loads at the policing *Test 0* and this is true for both low and high burstiness sources (see Section 7.5).

Again, with the implementation of the UPC function (via the four policing scenarios referred to before), extra complexity was added to the prioritised cell-rate simulator. To evaluate how much is lost in processing speed when using the policed cell-rate simulator, traffic experiments were run with the prioritised simulator and then repeated, under the same conditions, with the policed version of the simulator. The results obtained showed that the processing speeds of the two simulator's versions are of the same order of magnitude, except when the mean burst length of the sources in the system is very high. Therefore, the study of both traffic with priorities and monitoring of traffic with priorities with the enhanced versions of the cell-rate simulator does not incur in a significant processing speed reduction, provided there is a moderate load in the network and the traffic sources do not have a very high mean burst length.

Finally, this thesis investigated the degree of accuracy of the cell-rate simulator for traffic with and without priorities by comparing the simulator's results with those of an ATM testbed (see Chapter 8). Both versions of the cell-rate simulator proved to be quite accurate and thus confirm the enhanced simulator as a powerful tool in the study of ATM based networks. Some policing results of the ATM testbed were compared with a fluid flow analysis, as reported in Section 7.2.2 (see also [EXPL94d] and [Yin91]). In this thesis, two related comparisons were carried out: 1) the prioritised cell-rate simulator gives similar results (although it overestimates the high priority cell loss) to the ATM testbed; 2) the exact fluid flow analysis (described in Chapter 7) compares well with the prioritised cell-rate simulator. Therefore, it can be said that the ATM testbed and the discretised fluid flow model were indirectly compared; this represents new material, since the fluid flow model described in Chapter 7 is different from that of [Yin91].

In summary, the core research work reported in this thesis comprised the enhancement by the author of a cell-rate simulator to study two traffic control mechanisms in ATM networks: *priority control* and *policing*. Also, an exact fluid flow analysis has been extended by the author to study traffic with priorities and an upper bound analysis for homogeneous traffic sources was adapted by the author to study the policing of traffic with priorities. The new improved versions of the simulation tool proved to be fairly accurate when compared with other simulation and analytical approaches. The new facilities (i.e., a priority mechanism and a policing mechanism) implemented in the cell-rate simulator thus make it a powerful tool in the performance evaluation of ATM networks links.

Bearing in mind the brief description just given, it is obvious that some aspects of the work reported in this thesis could be improved. For example, concerning priority mechanisms, an implementation of the exact priority mechanism PBS (i.e., where the threshold of the queue is only a proportion of the queue size) in both a cell level and the cell-rate simulator would be useful to investigate the true accuracy degree of the prioritised cell-rate simulator. In fact, the cell-rate simulator does not model *cell scale queuing* (that occurs when several cells arrive more or less simultaneously, with small buffer capacities and low network utilisations). This is the reason why the cell loss results for high priority traffic do not closely match those obtained by Kröner (see [Krön91]), particularly when the proportion of high priority traffic is low.

The implementation of the exact PBS mechanism in the cell-rate simulator is possible, as noted in Section 5.2.1. However, since the analysis of the cell-rate simulator's queue is executed according to both its previous and current states (determined by the buffer capacity and the service rate), modelling the queuing above the threshold would imply the analysis of more states in the queue. This would introduce a significant extra degree of complexity in the simulator, but it would also produce more accurate results.

Concerning the priority mechanism implemented in the simulator, it is worth noting that all the studies carried out in this thesis considered only fixed queue thresholds. Other authors have considered the introduction of adaptive thresholds which "… can be optimum all the time …" (see [Roth90]) in terms of the maximum admissible load in a network, but their implementation has proved to be very costly.

The studies carried out in this thesis (for traffic with priorities and for policing) evaluated the QoS of the traffic simulated only with regard to cell loss. In Section 3.5, it was noted that the two most common (and important) measures for evaluating the QoS requirements of a given traffic source are the cell loss and the cell delay. The introduction of the latter measure in the cell-rate simulator would therefore give more insight into how (for example) traffic is affected by policing. The calculation of the cell delay distribution in the cell-rate simulator is possible, as noted by Pitts (see [Pitt93, pp.162]). However, it would only be possible to calculate those measures in situations of burst scale congestion since the simulator does not model cell scale congestion.

Part of the validation process applied to the prioritised cell-rate simulator involved using an extended exact fluid flow analysis. This analytical method, although valuable, only refers to the modelling of one On/Off traffic source. It would therefore be interesting to extend the model to the case of multiple traffic sources. This would enable studies involving more realistic mixtures of traffic. The extension of the fluid flow analysis to multiple sources could prove to be a difficult task. This is because, even without considering traffic with assigned priority levels, it would be difficult to enumerate the states of the system and quantify what happens in each of them. In other words, the consideration of multiple sources would lead to a so called *state explosion*, that would be mathematically complex to solve by using the level crossing approach (see [Scho94]).

As seen in Section 5.1.1, the cell-rate simulator describes traffic sources by using states, each with a fixed cell-rate that follows a given probability distribution. This implied the restricted type of sources used in the studies carried out, namely On/Off and GMDP modelled traffic sources (which represent variable bit rate sources). For a more realistic representation of the traffic environment in an ATM network, it would be important to have considered also constant bit rate sources. The cell-rate simulator can represent the so called *pseudo CBR sources* (see [EXPL94c]). These are described by two states that have the same characteristics in terms of their cell rates and state durations. The degree of accuracy of such a modelled CBR source is dependent on the states' chosen duration.

One important aspect of the traffic source models used in the traffic studies is the consideration of *measures of correlation*; they should be captured by the source models used and extracted from traffic measurements. This is catered for (to some extent) by the cell-rate simulation model described in Chapter 5. Directly related is the need to develop tools that consider reliable source models (i.e., models that consider the matching of traffic parameters, by taking into account statistical measurements). This is usually difficult because of the (still) lack of statistical measurements from certain traffic sources (e.g., new services such as Internet access and Video on Demand). As more ATM sources become available, the range of experiments on the testbed could be extended.

Another aspect related to the modelled and used traffic sources in this thesis is the lack of studies that take into consideration the new traffic classification only introduced very recently by influence of the ATM Forum and recognised by ITU in [ITU95b] (see Section 3.3). This classification includes two new types of traffic that relate mainly to computer data applications: the *Available Bit Rate (ABR)* and *Unspecified Bit Rate (UBR)* traffic types, apart from the already existing CBR and VBR types of traffic.

The ABR and UBR type services have been introduced with the aim of dynamically using the bandwidth not already used by the CBR and VBR traffic types. Since the two new types of traffic do not have very stringent QoS requirements, the so called *guaranteed traffic* (i.e., the CBR and VBR types of traffic) will be "served" first in the network. This is similar to the situation where traffic has assigned priority levels and where a priority mechanism decides:

1. what traffic can always be accepted by the network (i.e., the so called *high priority traffic*);
2. which traffic should be lost/delayed in a situation of congestion (i.e., the *low priority traffic*).

In other words, the high priority traffic is in this case the guaranteed traffic, while the ABR and UBR types of traffic represent low priority traffic. This has a direct bearing to the studies with prioritised traffic carried out in this thesis.

Future work could (and should, since computer data applications/traffic have an increasing importance in telecommunications networks) address the performance of ATM network links to which also ABR traffic has access. The prioritised cell-rate simulator could again be used for this study. However, the assignment of priorities would now be done only in an implicit way (i.e., all the cells of a determined type of traffic - e.g., ABR traffic - would have the same priority) as opposed to the other possibility (i.e., the assignment of explicit priorities - see Chapter 1) also considered in the priority studies presented in this thesis. Apart from this, separate buffers could also be used for different types (i.e., levels of priority) of traffic.

ABR traffic only gives indication (to the network) of its minimum traffic requirements. Therefore, the representation of this type of traffic within the simulator would imply a mapping of the existing traffic indicators (e.g., peak cell rate and duration of states) into the parameters of a typical ABR traffic source. The type and extent of the modifications to the simulator that this would involve cannot be evaluated without a careful study of the behaviour of real ABR traffic sources.

The studies that involved using the policed simulator for traffic without priorities only considered one source being policed. This lead to very restrictive conclusions both about the performance of the policing scenarios for traffic without priorities and about the best UPC actions to take on violating traffic without priorities. Therefore, traffic experiments should be designed and analysed that consider the policing of a wider range of sources.

The validation of the policing scenarios for traffic with priorities used an approximate bufferless analysis adapted by the author to the case of policing traffic with priorities (see Section 7.4.1). However, the analysis is only valid for the case of homogeneous sources and it only gives global values for the cell tagging/discarding probability at *Test 0*, for the cell discarding probability at *Test 0+1* and for the cell loss probability at the queue. It would be interesting to modify the analysis in order to obtain the low and high priority loss ratios at each point in the network link and also consider heterogeneous sources. This could be done by using Kröner's description of an upper bound analysis - equivalent to that described in Section 7.4.1 - (see [Krön91] and Section 5.3.2) for traffic with priorities, that gives an upper bound for the cell loss of a network link where no queuing is taken into account.

The several versions (with/without priorities, with policing) of the cell-rate simulator used throughout this thesis only address what happens at one particular ATM link. It would be interesting to extend the simulator to the network level. This would allow more realistic traffic studies, as well as the possibility of analysing a link inside the network. Also, it must be noted that, by having just one network link, it will not be possible to know the characteristics of the output traffic (which will almost certainly differ from the input traffic). The (very simple) switching architecture considered by the simulator is another aspect that could be improved. Indeed, it may be worth implementing (in the future) switching architectures such as *shared memory* and *space division switching*; these would extend the cell-rate simulator's range of application, making it a more useful tool (see [Kouv94a], [Kouv95a] and [Yama91b]).

Finally, one last aspect of the studies with the cell-rate simulator could be improved. It refers to the fact that no Connection Admission Control (CAC) function has been used within the simulator. In other words, it is assumed that all the traffic generated (and analysed) by the simulator has previously been accepted to the network by some CAC function. The implementation of a CAC function in the prioritised cell-rate simulator would add more complexity to the tool. It would also be necessary to choose from a variety of possible methods (i.e., *linear approach*, *two-moment allocation*, *convolution approach*, … - see Section 3.6.1) the most adequate to carry out the task of allocating network resources and deciding which traffic connections can be accepted by the network. The extent of the modifications necessary to include this facility in the prioritised cell-rate simulator cannot be evaluated without a careful study of the available CAC mechanisms.

10. Conclusions

In this thesis, an existing cell-rate simulator was extended by the author to study traffic with priorities and the policing actions on misbehaving traffic in ATM based networks. The research work was developed by taking into account what happens at the ATM layer.

A priority mechanism, the Partial Buffer Sharing (PBS), was implemented by the author in the cell-rate simulator. This represents new work by the author, since burst level simulation had not been used before to study traffic with priorities. The PBS mechanism was preferred to the Pushout mechanism (this is the priority mechanism that leads to the highest load improvements) because of its implementation simplicity. The simulation results for cell loss obtained and compared with both other simulation and analytical approaches proved the accuracy of the simulator for low priority traffic (that usually represents the bulk of the traffic) and an overestimation of the high priority traffic cell loss results. Moreover, the priority mechanism implemented in the cell-rate simulator proved to satisfy the typical properties of priority mechanisms.

One of the analytical methods used to validate the prioritised cell-rate simulator was the result of an extension by the author to the case of space priorities (for both the approximate and exact PBS priority mechanisms) to a model that represents an On/Off source feeding an ATM buffer. The comparison of the two extensions indicated that the use of the approximate PBS mechanism produces accurate low priority cell loss results and it overestimates the prediction of the high priority cell loss. This means that the high priority cell loss results given by the prioritised cell-rate simulator are conservative.

The cell-rate simulation method used by the author has some limitations, such as the low speed gain for fairly short burst lengths and the non-modelling of the cell scale queuing component. However, the method was preferred to the corresponding cell level approach because it is faster at simulating rare events (such as cell loss) in ATM based networks.

Policing was also studied by implementing a UPC function in the prioritised cell-rate simulator, using the Leaky Bucket mechanism. This represents new work by the author, since so far (and to the author's knowledge), no burst level approach had yet been applied in the study of policing traffic. The policing studies addressed traffic without priorities that is either discarded or tagged when found to be misbehaving. The policing scenarios studied for traffic without priorities were validated by using an approximate fluid flow analysis for one On/Off modelled traffic source being policed by a Leaky Bucket. Two policing scenarios for traffic with priorities were also implemented by the author that take into account the latest ITU recommendations (i.e., that tag or discard violating high priority traffic and always discard any violating global traffic). These policing scenarios were validated by using an approximate analysis for a multiplexing of homogeneous On/Off modelled traffic sources, derived by the author for the particular case of policing traffic with priorities.

The performance of the cell-rate simulator (for traffic with and without priorities) was compared by the author with that of an ATM testbed under several experimental conditions. The simulator proved to be fairly accurate, especially for the global cell loss results of traffic without priorities and the low priority cell loss results.

The enhancement of the cell-rate simulator with the insertion, both of a priority mechanism and a policing function, implied a reduction in the processing speed of the prioritised and policed cell-rate simulators, respectively. However, it was observed that the reduction in speed is almost negligible for moderate network loads and for up to reasonably high mean burst lengths of the traffic sources.

This thesis has described how a simulator that uses the cell-rate simulation method can be used to study traffic with priorities and monitor traffic (with/without priorities) in an ATM based network. In view of the results obtained with the several tool versions, it can be said that the new enhanced simulator is a useful prediction tool in the study of traffic control mechanisms and behaviour of ATM networks.

References

[Fons94a] Paula Fonseca, Jonathan M. Pitts and Laurie G. Cuthbert, *Implementing a Space Priority Mechanism in a Cell-Rate ATM Network Simulator*, 11th UK Teletraffic Symposium, March 1994, pp.13B/1-13B/7

[Fons94b] Paula Fonseca, Jonathan M. Pitts and Laurie G. Cuthbert, *Validation of a Cell-Rate Model for Partial Buffer Sharing in an ATM Network Simulator*, Second Workshop on Performance Modelling and Evaluation of ATM Networks, IFIP TC6, July 1994, pp.53/1-53/9

[Fons95a] Paula Fonseca, Laurie G. Cuthbert and Jonathan M. Pitts, *Study of a Space Priority Mechanism for ATM Networks Using Both Analytical and Simulation Techniques*, 12th UK Teletraffic Symposium, March 1995, pp. 3/1-3/8

[Fons95b] Paula Fonseca, Jonathan M. Pitts and Laurie G. Cuthbert, *Exact Fluid-Flow Analysis of a Single On/Off Source Feeding an ATM Buffer With Space Priority*, IEE Electronics Letters, 22nd June 1995, vol.31, no.13, pp.1028-1029

[Alon89] Pedro L. Chas Alonso, Frco. J. Herrera Gálvez and Alejandro Alonso Hernán, *On the Use of Priorities in ATM Switching Structures*, Proceedings of the IEEE International Conference on Communications (ICC), 1989, pp.718-721

[Alta91] M. Villén-Altamirano and José Villén-Altamirano, *RESTART: A Method for Accelerating Rare Event Simulations*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.71-76

[Alta93] M. Villén-Altamirano, *Activities on Traffic Engineering for B-ISDN in CCITT, SG II*

[Ahma90] H. R. Ahmadi, R. Guérin and K. Sohraby, *Analysis of Leaky Bucket Access Control Mechanisms with Batch Arrival Process*, Proceedings of the IEEE GLOBECOM'90, 1990, pp.344-349

[Anic82] D. Anick, D. Mitra and M. M. Sondhi, *Stochastic Theory of a Data-Handling System with Multiple Sources*, The Bell System Technical Journal, vol.61, no.8, October 1982, pp.1871-1894

[Anti95] M. Antico, F. Bernabei and L. Gratta, *Traffic Control Mechanisms to Support ABR*

Services in ATM Networks, Proceedings of ATM Hot Topcis on Traffic and Performance: from RACE to ACTS, Milano, June 14th-15th, 1995, paper no.3

[ATMF92] ATM Forum: *Leaky Bucket Based UPC Algorithms*, Doc. No.92-256, November 1992

[ATMF94] ATM Forum: *Performance Analysis of GCRA for CBR Sources*, Doc. No.94-0182, March 1994

[Bae92] Jaime Jungok Bae, Tatsuya Suda and Rahul Simha, *Analysis of Individual Packet Loss in a Finite Buffer Queue with Heterogeneous Markvo Modulated Arrival Processes: A Study of Traffic Burstiness and Priority Packet Discarding*, Proceedings of the IEEE INFOCOM'92, 1992, pp.219-230

[Balt95] E. Baltatzis, K. Gratland, G. Rigolio and E. Wallmeier, *Comparison of Fast Resource Management and Available Bit Rate Techniques for Efficient Use of ATM Bandwidth*, Proceedings of ATM Hot Topcis on Traffic and Performance: from RACE to ACTS, Milano, June 14th-15th, 1995, paper no.2

[Bono95] Flavio Bonomi, *The ABR Service for ATM Networks*, Proceedings of ATM Hot Topcis on Traffic and Performance: from RACE to ACTS, Milano, June 14th-15th, 1995, paper no.1

[Boud91] Jean-Yves Le Boudec, *An Efficient Solution Method for Markov Models of ATM Links with Loss Priorities*, IEEE Journal on Selected Areas in Communications, vol.9, no.3, April 1991, pp.408-417

[Brie91] U. Briem, T. H. Theimer and H. Kröner, *A General Discrete-Time Queuing Model: Analysis and Applications*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.13-19

[Butt91] M. Buttó, E. Cavallero and A. Tonietti, *Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks*, IEEE Journal on Selected Areas in Communications, vol.9, no.3, April 1991, pp.335-342

[Cast91] P. Castelli, E. Cavallero and A. Tonietti, *Policing and Call Admission Problems in ATM Networks*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.847-852

[Chao91] H. J. Chao, *Design of Leaky Bucket Access Control Schemes in ATM Networks*, Proceedings of the IEEE International Conference on Communications (ICC), 1991, pp.180-187

[Çinl75] Erhan Çinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Inc., 1975

[Cosm90] J. Cosmas and A. Odinma-Okafor, *Variable Rate Video Signals Characterisation using a GMDP Model*, 7th IEE UK Teletraffic Symposium, April 1990, pp.5/1-5/7

[Cuth93] L. G. Cuthbert and J.-C. Sapanel, *ATM, The Broadband Telecommunications Solution*, IEE Telecommunications Series 29, 1993

[Dagi93] A. F. Dagiukas and M. Ghanbari, *Priority Queuing Disciplines in ATM Switches Carrying Two Layer Video Traffic*, IEE 10th UK Teletraffic Symposium on Performance Engineering in Telecommunications Networks, BT Laboratories, Martlesham Heath, 14th-16th April 1993, pp.4/1-4/6

[Dron91] L. G. Dron, G. Ramamurthy and B. Sengupta, *Delay Analysis of an ATM Switch for Continuous Bit Rate Traffic*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.33-38

[Eckb92] A. E. Eckberg, *B-ISDN/ATM Traffic and Congestion Control*, IEEE Network, September 1992, pp.28-37

[Elwa92] Anwar I. Elwalid and Debasis Mitra, *Fluid Models for the Analysis and Design of Statistical Multiplexing with Loss Priorities on Multiple Classes of Bursty Traffic*, Proceedings of the IEEE INFOCOM'92, 1992, pp.415-425

[Elwa94] Anwar I. Elwalid and Debasis Mitra, *Statistical Multiplexing with Loss Priorities in Rate-Based Congestion Control of High-Speed Networks*, IEEE Transactions on Communications, vol.42, no.11, November 1994, pp.2989-3002

[Esak90] Hiroshi Esaki, Kazuaki Iwamura and Toshikazu Kodama, *A Simple and Effective Admission Control Method for an ATM Network*, Proceedings of the IEEE GLOBECOM’90, 1990, pp.28-33

[EXPL94a] RACE R2061 EXPLOIT, Deliverable 18, *Results of Experiments on Traffic Control using Test Equipment*, June 1994

[EXPL94b] RACE R2061 EXPLOIT, Deliverable 19, *Results of Experiments on Resource Management using Test Equipment*, June 1994

[EXPL94c] RACE R2061 EXPLOIT, Document R2061_UST_31_006.02_CD_CC, *Summary of Experiment Results on Multiplexing in Weeks 40 and 44*, November 1994

[EXPL94d] RACE R2061 EXPLOIT, Deliverable 28, *Results of Experiments on Traffic Control using Real Applications*, December 1994

[Fend91] K. W. Fendick, D. Mitra, I. Mitrani, M. A. Rodrigues, J. B. Seery and A. Weiss, *An Approach to High-performance, High-speed Data Networks*, IEEE Communications Magazine, October 1991, pp.74-82

[Fisa91] N. Fisal and R. L. Brewster, *Congestion and Flow Control in Private Wide Area ATM Network - A simulation Study*, IEE 8th UK Teletraffic Symposium, April 1991, pp.16/1-16/4

[Fisc94] Wolfgang Fischer, Eugen Wallmeier, Thomas Worster, Simon P. Davies and Andrew Hayter, *Data Communications Using ATM: Architectures, Protocols and Resource Management*, IEEE Communications Magazine, August 1994, pp.24-33

[Fros94] Victor S. Frost and Benjamin Melamed, *Traffic Modelling for Telecommunications Networks*, IEEE Communications Magazine, March 1994, pp.70-81

[Gall89] G. Gallassi, G. Rigolio, P. Vaccari and L. Verri, *Resource Allocation in ATM Networks*, Report from REVOLVE R1028, 1989, pp.1-12

[Gall90] G. Gallassi, G. Rigolio and L. Fratta, *Bandwidth Assignment in Prioritised ATM Networks*, Proceedings of the IEEE GLOBECOM, 1990, pp.852-856

[Garc91] J. Garcia and O. Casals, *Statistical Multiplexing Gain Using Space Priority Mechanisms*, Proceedings of the IEEE GLOBECOM’91, Phoenix, Arizona, vol.2, December 1991, pp.929-933

[Garc92] J. Garcia and O. Casals, *Stochastic Models of Space Priority Mechanisms with Markovian Arrival Processes*, Annals of Operations Research, vol.35, 1992, pp.271-295

[Gilb91] Henry Gilbert, Osama Aboul-Magd and Van Phung, *Developing a Cohesive Traffic Management Strategy for ATM Networks*, IEEE Communications Magazine, October 1991, pp.36-45

[Grav91a] Annie Gravey and Gérard Hebuterne, *Mixing Time and Loss Priorities in a Single Server Queue*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.147-152

[Grav91b] Annie Gravey, Pierre Boyer and Gérard Hebuterne, *Tagging versus Strict Rate*

Enforcement in ATM Networks, Proceedings of the IEEE GLOBECOM'91, 1991, pp.271-275

[Guér91] R. Guérin, H. Ahmadi and M. Naghshineh, *Equivalent Capacity and its Application to Badnwidth Allocation in High-Speed Networks*, IEEE Journal on Selected Areas in Communications, vol.9, no.7, September 1991, pp.968-981

[Guer91] A. Guerrero and F. Lozano, *Queuing Models Applying to a Class of ATM Multiplexers*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.999-1004

[Guil92] Fabrice Guillemin and Alain Dupuis, *A Basic Requirement for the Policing Function in ATM Networks*, Computer Networks and ISDN Systems, no.24, 1992, pp.311-320

[Habi91] Ibrahim W. Habib and Tarek N. Saadawi, *Controlling Flow and Avoiding Congestion in Broadband Networks*, IEEE Communications Magazine, October 1991, pp.46-53

[Habi92] Ibrahim W. Habib and Tarek N. Saadawi, *Multimedia Traffic Characteristics in Broadband Networks*, IEEE Communications Magazine, vol.30, no.7, July 1992, pp.48-54

[Hébu90] Gérard Hébuterne and Annie Gravey, *A Space Priority Queuing Mechanism for Multiplexing ATM Channels*, Computer Networks and ISDN Systems, no.20, 1990, pp.37-43

[Heff86] Harry Heffes and David M. Lucantoni, *A Markov Modulated Characterisation of Packetised Voice and Data Traffic and Related Statistical Multiplexer Performance,* IEEE Journal on Selected Areas in Communications, vol.4, no.6, September 1986, pp.856-868

[Hong91] Duke Hong and Tatsuya Suda, *Congestion Control and Prevention in ATM Networks*, IEEE Network Magazine, July 1991, pp.10-16

[Hong92] S. Hong, H. G. Perros and H. Yamashita, *A Discrete-Time Queuing Model of the Shared Buffer Switch with Bursty Arrivals*, Research Report, Computer Science Department, North Carolina State University, 1992

[Huan94] T.-Y. Huang and J.-L. C. Wu, *Performance Analysis of ATM Switches using Priority Schemes*, IEE Proceedings on Communications, vol.141, no.4, August 1994, pp.248-254

[Hui88] J. Y. Hui, *Resource Allocation for Broadband Networks*, IEEE Journal on Selected

Areas in Communications, no.6, 1988, pp.1598-1608

[IBC93] IBC Common Functional Specification, *Usage Parameter Control/Network Parameter Control*, Specification RACE D711, Issue D, December 1993, pp.1-9

[ITU88a] *General Aspects of Quality of Service and Network Performance in Digital Networks, including ISDNs*, ITU Recommendation I.350, 1988

[ITU88b] *Network Performance Objectives for Connection Processing Delays in an ISDN,* ITU Recommendation I.352, Vol. III, Fascicle III.8, Blue Book, 1988

[ITU89] *ITU COM XVIII-R 4-E*, Report of the BBTG Meeting, January/February 1989

[ITU90a] *Broadband Aspects of ISDN-BISDN ATM Functional Characteristics, ATM Layer Specification for B-ISDN and B-ISDN ATM Adaptation Layer Functional Description/Specification*, ITU Draft Recommendations I.150 and I.361-I.363, Geneva, Switzerland, February 1990

[ITU90b] *B-ISDN Service Aspects*, ITU Recommendation I.211, 1990

[ITU90c] *Broadband Aspects of ISDN-BISDN*, ITU Recommendation I.121, 1990

[ITU91] *Vocabulary of Terms for Broadband Aspects of ISDN*, ITU Recommendation I.113, 1991

[ITU92] *ITU IVS Baseline Document*, SG XVIII/8, Geneva, June 1992

[ITU93] *B-ISDN ATM Layer Cell Transfer Performance*, ITU Draft Recommendation I.356, March 1993

[ITU94] *Traffic Control and Congestion Control in B-ISDN*, ITU Recommendation I.371, Frozen Issue - Geneva, March 1994

[ITU95a] *Methods for Traffic Control in B-ISDN*, ITU Draft new Recommendation I.37x, June 1995

[ITU95b] *Traffic Control and Congestion Control in B-ISDN,* ITU Recommendation I.371, Frozen Issue - Geneva, July 1995

[Izma93] Rauf Izmailov and Ender Ayanoglu, *Priority Statistical Multiplexing of Mixed VBR Video and CBR Traffic in B-ISDN/ATM with a Threshold Algorithm*, Proceedings of the IEEE INFOCOM'93, 1993, pp. 910-918

[Kang93] Chung G. Kang and Harry H. Tan, *Queuing Analysis of Explicit Priority Assignment Partial Buffer Sharing Schemes for ATM Networks*, Proceedings of the IEEE INFOCOM'93, 1993, pp.810-819

[Kim92] Young Han Kim, Byung Chul Shin and Chong Kwan Un, *Performance Analysis of Leaky-Bucket Bandwidth Enforcement Strategy for Bursty Traffics in an ATM Network*, Computer Networks and ISDN Systems, vol.25, no.3, 1992, pp.295-303

[Kouv94a] D. D. Kouvatsos and S. G. Denazis, *A Universal Building Block for the Approximate Analysis of a Shared Buffer ATM Switch Architecture*, Annals of Operational Research, no.44, 1994, pp.241-278

[Kouv94b] D. D. Kouvatsos and S. G. Denazis, *ME-Based Approximations for General Discrete-Time Queuing Models*, Performance Evaluation, Special Issue on Discrete-Time Models and Analysis Methods of Performance Evaluation, vol.21, 1994, pp.81-109

[Kouv95a] D. D. Kouvatsos and J. Wilkinson, *A Product-Form Approximation for Discrete-Time Arbitrary Networks of ATM Switch Architectures*, Performance Modelling and Evaluation of ATM Networks, IFIP Publication, Chapman and Hall, London, vol.1, 1995, pp.365-383

[Kouv95b] D. D. Kouvatsos and R. Fretwell, *Closed Form Performance Distributions of a Discrete Time* $GI^{G}/D/1/N$ *Queue with Correlated Traffic*, Data Communications and their Performance, IFIP Publication, Edited by S. Fdida and R. O. Onvural, Chapman and Hall, London, 1995, pp.142-163

[Krön90] Hans Kröner, *Comparative Performance Study of Space Priority Mechanisms for ATM Networks*, Proceedings of the IEEE INFOCOM'90, 1990, pp. 1136-1143

[Krön91] Hans Kröner, Gérard Hébuterne, Pierre Boyer and Annie Gravey, *Priority Management in ATM Switching Nodes*, IEEE Journal on Selected Areas in Communications, vol.9, no.3, April 1991, pp.418-427

[Kühn95] Paul J. Kühn, *Reminder on Queueing Theory for ATM Networks*, Proceedings of the First ATM Traffic Expert Symposium, Basel, Switzerland, April 1995, paper no.4

[Law91] Averill M. Law and W. David Kelton, *Simulation Modeling and Analysis, McGraw-Hill,* Second Edition, 1991

[Lee92] Jun Won Lee and Byeong Gi Lee, *Performance Analysis of ATM Cell Multiplexer with MMPP Input*, IEICE Trans. Commun. vol.E75-B, no.8, August 1992, pp.709-714

[Leme94] Marc Lemercier and Guy Pujolle, *A Performance Study of a New Congestion Management Scheme in ATM Broadband Networks: The Multiple Push-Out*, Proceddings of the International Teletraffic Congress ITC-14, June 1994, pp.1261-1270

[Liao94] KE-Qiang Liao, *Queuing Analysis of Partial Buffer Sharing with Markov Modulated Poisson Inputs*, Proceedings of the International Teletraffic Congress ITC-14, June 1994, pp.55-64

[Lin91a] Arthur Y.-M. Lin and John A. Silvester, *Priority Queuing Strategies and Buffer Allocation Protocols for Traffic Control at an ATM Integrated Broadband Switching System*, IEEE Journal on Selected Areas in Communications, vol.9, no.9, December 1991, pp.1524-1535

[Lin91b] T. Lin and L. Kleinrock, *Performance Analysis of Finite-Buffered Multistage Interconnection Networks with a General Traffic Pattern*, Proceedings of the ACM SIG - METRICS'91, 1991, pp.68-89

[Magl88] Basil Maglaris, Dimitris Anastassiou, Prodip Sen, Gunnar Karlsson and John D. Robbins, *Performance Models of Statistical Multiplexing in Packet Video Communications*, IEEE Transactions on Communications, vol.36, no.7, July 1988, pp.834-844

[Mant91] Sam Manthorpe, John A. Schormans, Jonathan M. Pitts and Eric M. Scharf, *A Simulation Study of Buffer Occupancy in the ATM Access Network: Are Renewal Assumptions Justified?*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.801-805

[Mark95] Brian L. Mark and G. Ramamurthy, *UPC Based Traffic Descriptors for ATM: How to Determine, Interpret and Use them*, Proceedings of the First ATM Traffic Expert Symposium, Basel, Switzerland, April 1995, paper no.9

[Mase91] Kenichi Mase and Shigeo Shioda, *Real-Time Network Management for ATM Networks*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.133-140

[Meye93] J. F. Meyer, S. Montagna and R. Paglino, *Dimensioning of an ATM Switch with shared Buffer and Threshold Policy*, Computer Networks and ISDN Systems, no.26, 1993, pp.95-108

[Mitr87] I. Mitrani, *Modelling of Computer and Communication Systems*, Cambridge Computer Science Texts, Cambridge University Press, 1987

[Mitr94] N. M. Mitrou, *Traffic Control & Congestion Control in ATM Networks: A Unified View*, Proceedings of the Second Workshop on Performance Modelling and Evaluation of ATM Networks, IFIP Workshop TC6, University of Bradford, 4th-7th July 1994, Tutorial paper no.5

[Mitr95] N. M. Mitrou, K. P. Kontovasilis and E. N. Protonotarius, *ATM Traffic Engineering for ABR Service Provisioning*, Proceedings of the First International ATM Traffic Expert Symposium, Basel, 20th-21st April 995, paper no.11

[Miya91] Yasuhiro Miyao, *Bandwidth Allocation in ATM Networks that Guarantee Multiple QoS Requirements*, Proceedings of the IEEE INFOCOM'93, 1993, pp.1398-1403

[Mura91] Tutomu Murase, Hiroshi Suzuki, Shohei Sato and Takao Takeuchi, *A Call Admission Control Scheme for ATM Networks Using a Simple Quality Estimate*, IEEE Journal on Selected Areas in Communications, vol.9, no.9, December 1991, pp.1461-1470

[Nayl67] T. H. Naylor and J. M. Finger, *Verification of Computer Simulation Models*, Management Science, no.14, 1967, pp.92-101

[Newm94] Peter Newman, *Traffic Management for ATM Local Area Networks*, IEEE Communications Magazine, August 1994, pp.44-50

[Okad91] Tadanobu Okada, Hirokazu Ohnishi and Naotaka Morita, *Traffic Control in Asynchronous Transfer Mode*, IEEE Communications Magazine, September 1991, pp.58-62

[Onvu94] Raif O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House, 1994

[Pan91] Huanxu Pan, Hiroyuki Okazaki and Issei Kino, *Analysis of a Gradual Input Model for Bursty Traffic in ATM*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.795-800

[Petr91] David W. Petr and Victor S. Frost, *Nested Threshold Cell Discarding for ATM Overload Control: Optimisation under Cell Loss Constraints*, Proceedings of the IEEE INFOCOM'91, 1991, pp. 1403-1412

[Pitt90] Jonathan M. Pitts and Zhili Sun, *Burst Level Teletraffic Modelling and Simulation of*

Broadband Multiservice Networks, 7th UK Teletraffic Symposium, April 1990, pp.7/1-7/5

[Pitt91a] Jonathan M. Pitts, Zhili Sun, John P. Cosmas and Eric M. Scharf, *Burst-level Teletraffic Modelling: Applications in Broadband Network Studies*, 3rd IEE Conference on Telecommunications, March 1991, pp.348-352

[Pitt91b] Jonathan Pitts, Zhili Sun and Eric M. Scharf, *A Comparison of Burst-level and Cell-level Approaches to the Simulation of ATM Networks*, Proceedings of the International Teletraffic Congress ITC-13, June 1991, pp.257-262

[Pitt91c] Jonathan M. Pitts and Laurie G. Cuthbert, *Traffic Mixes on a Broadband ATM Link*, Electronics Letters, vol.27, no.15, July 1991, pp.1350-1351

[Pitt91d] Jonathan M. Pitts and Laurie G. Cuthbert, *Multi-Service Bandwidth Allocation in ATM*, IEE 8th UK Teletraffic Symposium, April 1991, pp.17/1-17/4

[Pitt93] Jonathan M. Pitts, *Cell-Rate Simulation Modelling of Asynchronous Transfer Mode Telecommunications Networks*, PhD Thesis, July 1993

[Port94] Thomas F. La Porta, Malathi Veeraraghavan, Ender Ayanoglu, Mark Karol and Richard D. Gitlin, *B-ISDN: A Technological Discontinuity*, IEEE Communications Magazine, October 1994, pp.84-97

[Pryc91] Martin de Prycker, *Asynchronous Transfer Mode - Solution for Broadband ISDN*, Ellis Horwood, 1991

[RACE91] RACE R1022 Technology for ATD, Deliverable D22, *Final Traffic Models and Applications*, R1022 Project Office, Alcatel Bell, 1991

[RACE92] RACE R1022 Technology for ATD, Deliverable D126, *Final Recommendations on Connection Admission Control, Usage Monitoring and Validation of Control Schemes*, R1022 Project Office, Alcatel Bell, 1992

[Rama91] V. Ramaswami, M. Rumsewicz, W. Willinger and T. Eliazov, *Comparison of Some Traffic Models for ATM Performance Studies*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.7-12

[Rama94] G. Ramamurthy, R. S. Dighe and D. Raychaudhuri, *An UPC-Based Traffic Control Framework for ATM Networks*, Proceedings of the IEEE GLOBECOM'94, 1994, pp.600-605

[Rath91] Erwin P. Rathgeb, *Modelling and Performance Comparison of Policing Mechanisms for ATM Networks*, IEEE Journal on Selected Areas in Communications, vol.9, no.3, April 1991, pp. 325-334

[Ren94] Jing-Fei Ren, Jon W. Mark and Johnny W. Wong, *A Dynamic Priority Queuing Approach to Traffic Regulation and Scheduling in B-ISDN*, Proceedings of the IEEE INFOCOM’94, 1994, pp.612-618

[Robe91a] James W. Roberts and Annie Gravey, *Recent Results on B-ISDN/ATM Traffic Modelling and Performance Analysis - A Review of ITC13 Papers*, Proceedings of the IEEE GLOBECOM'91, 1991, pp.37.3.1-37.3.6

[Robe91b] James W. Roberts and Jorma T. Virtamo, *The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer*, IEEE Transactions on Communications, vol.39, no.2, February 1991, pp.298-303

[Robe92] Edited by J. W. Roberts, *Information Technologies and Sciences - COST 224 - Performance Evaluation and Design of Multiservice Networks*, 1992

[Roth90] Karl Rothermel, *Priority Mechanisms in ATM Networks*, Proceedings of the IEEE Global Telecommunications Conference & Exhibition - IEEE GLOBECOM'90, 1990, pp. 847-851

[Rose 92] Jonathan Rosenberg, Robert E. Krout, Louis Gomez and C. Alan Buzzard, *Multimedia Communications for Users*, IEEE Communications Magazine, May 1992, pp.20-36

[Sait94] Hiroshi Saito, *Teletraffic Technologies in ATM Networks*, Artech House, 1994

[Scho91] J. A. Schormanns, J. Pitts and E. Scharf, *Time Priorities in ATM Switches*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.527-532

[Scho92] J. A. Schormanns, *Prioritised Geo/D/1 Telecommunications Switch Model*, IEE Electronics Letters, vol.28, no.6, March 1992, pp.594-598

[Scho94] J. A. Schormanns, J. M. Pitts and L.G. Cuthbert, *Exact Fluid-Flow Analysis of Single ON/OFF Source Feeding and ATM Buffer*, IEE Electronics Letters, vol.30, no.4, pp.1116-1117

[Schw87] Mischa Schwartz, *Telecommunication Networks: Protocols, Modelling and Analysis*, Addison-Wesley, 1987

[Schw95] Mischa Schwartz, *Lecture Series on Traffic Management for Broadband ATM-Based Integrated Networks*, Department of Electrical and Electronic Engineering, University College of London, 24th January 1995 - 28th March 1995

[Skli93] Andreas Skliros, *Dimensioning the CDV Tolerance Parameter (τ) for ATM Multiplexers*, IEE 10th UK Teletraffic Symposium on Performance Engineering in Telecommunications Networks, BT Laboratories, Martlesham Heath, 14th-16th April 1993, pp.5/1-5/6

[Srir86] K. Sriram and W. Witt, *Characterising Superposition Arrival Processes in Packet Multiplexers for Voice and Data*, IEEE Journal on Selected Areas in Communications, vol.4, no.6, September 1986, pp.833-846

[Stav92] Ioannis Stavrakakis, *Traffic Modelling for Packet Communication Networks*, International Journal of Digital and Analog Communication Systems, vol.5, 1992, pp.157-165

[Sun92] Zhili Sun, Jonathan M. Pitts and Laurie G. Cuthbert, *MACROSIM - A Burst Level Simulation tool for ATM Networks*, QMW, 1992

[Sun93] Zhili Sun, John Cosmas and Laurie G. Cuthbert, *Simulation Studies of Multiplexing and Demultiplexing Performance in ATM Switch Fabrics*, 10th UK Teletraffic Symposium, 14th-16th April 1993, pp.21/1-21/5

[Suth92] Joe Sutherland and Larry Litteral, *Residential Video Services*, IEEE Communications Magazine, July 1992, pp.36-41

[Syka91a] Efstathios D. Sykas, Konstantinos M. Vlakos and Emmanuel N. Protonotarios, *Mathematical Tools for Analysis of ATM Systems*, Proceedings of the International Teletraffic Congress ITC-13, 1991, pp.781-786

[Syka91b] Efstathios D. Sykas, Konstantinos M. Vlakos and Nikolaus G. Anerousis, *Performance Evaluation of Statistical Multiplexing Schemes in ATM Networks*, Computer Communications, vol.14, no.5, June 1991, pp.273-286

[Wins95] Steven B. Winstanley (QMW) programmed the policing unit in the EXPLOIT ATM testbed in order to emulate the Partial Buffer Sharing priority mechanism.

[Wu93] Guo-Liang Wu and Jon W. Mark, *Discrete Time Analysis of Leaky-Bucket Congestion Control*, Computer Networks and ISDN Systems, no.26, 1993, pp.79-94

[Yama91a] Hiroshi Yamada and Shuichi Sumita, *A Traffic Measurement Method and its Application for Cell Loss Probability Estimation in ATM Networks*, IEEE Journal on Selected Areas in Communications, vol.9, no.3, April 1991, pp.315-323

[Yama91b] H. Yamashita, H. G. Perros and S. Hong, *Performance Modelling of Shared Buffer ATM Switch Architecture*, Proceedings of the International Teletraffic Congress ITC-13, vol. on Teletraffic and Datatraffic in a Period of Change, Edited by Jensen and Iversen, North-Holland, 1991

[Yama92] Naoaki Yamanaka, Youichi Sato and Ken-Ichi Sato, *Traffic Shaping for VBR Traffic in ATM Networks*, IEICE Transactions on Communications, vol. E75-B, no.10, October 1992, pp.1105-1108

[Yama95] Naoaki Yamanaka, Youichi Sato and Ken-Ichi Sato, *Performance Limitation of the Leaky Bucket Algorithm for ATM Networks*, IEEE Transactions on Communications, vol.43, no.8, August 1995, pp.2298-2300

[Yazi92] Setiadi Yazid and H. T. Mouftah, *Congestion Control Methods for BISDN*, IEEE Communications Magazine, July 1992, pp.42-47

[Yin91] Nanying Yin and Michael G. Hluchyj, *Analysis of the Leaky Bucket Algorithm for On/Off Data Sources*, Proceedings of the IEEE GLOBECOM'91, 1991, pp.254-260

[Zeng94] Jianyu Zeng and Lazaros F. Merakos, *Analysis of a Priority Leaky Bucket Regulator for Virtual Path Traffic in ATM Networks*, Proceedings of the IEEE GLOBECOM'94, 1994, pp.606-611