Simulation of virtual connections in ATM networks

ter Horst, Paul

Award date:
1993

Eindhoven University of Technology
Department of Electrical Engineering
Digital Information Systems Group

Simulation of Virtual Connections in ATM networks

Paul ter Horst

Graduation report

Graduation period: September 1992 - July 1993

Supervisor: Prof. ir. F. van den Dool
Coaches: Ir. M.J.M. van Weert
Ir. B.J. van Rijnsoever

Abstract

In an ATM network, information is transferred in packets of a short fixed length named cells. Due to the asynchronous way of multiplexing , queuing of cells at switching points in the network is needed. There is a maximum of cells that can be queued. If more cells need to be queued than possible, cells are lost.

Besides cell loss, cells are delayed due to the queuing. The variation of cell delay(delay jitter) is also a performance measure. The jitter is important when a time relation between cells is required, since this time relation is disturbed by the queuing delay.

Simulation is needed to validate assumptions made by analytic modeling of a virtual connection. The analytic modeling is done to study the performance of a virtual connection in an ATM network. The values of the performance measures are that extreme that conventional simulation will last too long.

The main object of this graduation report is to study and implement a simulation technique to estimate, with relative short simulation runs, the probability of performance measures. The small probabilities of the performance measures can be expressed as probability of rare events.

Restart is a method for accelerating simulations by estimating rare events. Restart stands for repetitive simulation trials after reaching a threshold. To implement Restart a simulation environment is created. This simulation environment named Simmust, is developed using an object oriented approach. With Simmust it is possible to simulate a diversity of models of virtual connections.

Using Simmust as basis Restart is implemented. Restart is applied on one particular model. On this model Restart is used to determine the probability of cell loss. Finally some experiments are done to see in what way the acceleration of the method can be improved.

Contents

Contents 1

Abbreviations: 4

Chapter 2
Asynchronous Transfer Mode 7
2.1 Introduction 7
2.2 The transfer mode ATM 8
2.2.1 ATM as transfer technique 8
2.2.2 Statistical multiplexing 8
2.2.3 ATM connections 9
2.3 ATM protocol reference model(PRM) 10
2.3.1 Physical Layer 11
2.3.2 ATM layer 12
2.3.3 AAL layer 12
2.4 ATM switching 13
2.4.1 Header translation and routing 14
2.4.2 Queuing Strategies 15
2.5 Topics on ATM 17

Chapter 3
Performance of a Virtual Connection 19
3.1 Introduction 19
3.2 Performance measures 20
3.2.1 Cell Loss 20
3.2.2 Cell delay 21
3.3 Parameters of influence 22
3.4 The use of simulation in performance study. 24
3.5 Performance model 24
3.5.1 Traffic models 24
3.5.2 Virtual Connection model 27
3.6 Topics 28

Chapter 4
Simulation techniques 31
4.1 Introduction 31
4.2 Simulation 32
4.3 Bottleneck of simulation 33
4.4 Accelerated simulation methods 33
4.5 Extrapolative Methods 33
4.6 Variance reduction techniques 34
4.6.1 Importance sampling 35
4.6.2 Restart 37
4.7 Overview of the methods 37

Chapter 5
Restart and the implementation of the simulation software 39
5.1 Introduction 39
5.2 Restart 40
5.2.1 Working of Restart 40
5.2.2 The assumptions and variance evaluation of Restart 42
5.2.3 Restart with Hysteresis 44
5.2.4 Gain of the method 46
5.3 Choice of the simulation language 47
5.3.1. The methods of simulation languages 47
5.3.2. Possibilities of languages: 48
5.3.3 Choice of the language 49
5.4 Simulation of a Virtual connection using Restart 50
5.4.1 SIMMUST 51
5.4.2 Restart 54
5.5 Restart on a specific VC model 56
5.6 Topics 56

Chapter 6
Analysis of simulation results obtained with Restart 59
6.1 Introduction 59
6.2 The gain of using Restart 60
6.2.1 The Gain with optimal parameters 60
6.2.2 The Gain with non-optimal parameters 63
6.2.3 The three formulas of the gain 64
6.3 Calculation of the confidence interval 64
6.4 Model to which Restart is applied 65
6.5 Simulation results 66
6.5.1. Effect of a smaller probability of event A 66
6.5.2 Increasing of the hysteresis effect 67
6.5.3 Different choices of event B 68
6.6 Review of the Results 70
6.7 References 70

Chapter 7
Conclusions and recommendations 71

Appendices

A Notations Restart 73

Appendix B
SIMMUST 75
B1 Introduction 76
B2. Constant 76
B3. Modelobj 77
B4. Timing of the processes 79
B5. Perfobj 80
B5.1 Introduction 80
B5.2 Cell 80
B5.3 Cell loss 80
B5.4 Cell Delay 80
B6. Procedure Initialize 82
B7. Input specification: 82
B7.1 specification of input file 84
B8. Output specification 86
B9. Example 87

Appendix C
Restart 93
C1 Introduction 94
C2. Overview of the software 94
C3. Algorithm restart with hysteresis: 96
C3.1 Save and restore 97
C4. Input output specification 99
C4.1 specification of input file 99
C4.2 Output specification 101

Appendix D
Model of a VC and the results of Restart 103
D.1 Model on which Restart is applied 104
D.2 The four simulated cases 105
D.3 Results 107

4

Abbreviations:

AAL	ATM Adaption Layer
B-ISDN	Broadband ISDN
CLP	Cell Loss Priority
CBR	Constant BitRate
CLR	Cell Loss Rate
CLS	Connectionless server
CRC	Cyclic Redundacy Check
CS	Convergence Sublayer
FIFO	First In First Out
GFC	General Flow Control
HEC	Header Error Control
HOL	Head Of Line
HT	Header Translation
IBP	Interrupted Bernoulli Process
ISDN	Integrated Service Digital Network
LDM	Label Division Multiplexing
MID	Multiplex IDentification
MMBP	Modulated Markov Bernoulli Process
OOP	Object Oriented Programming
OSI	Open Systems Interconnection
pdf	probability density function
PDF	Probability Distribution Function
PL	Physical Layer
PLOAM	Physical Layer Operating And Maintenance
PM	Physical Medium
PRM	Protocol Reference Model
PT	Payload Type
RES	Reserved
SAR	Segmentation And Reassemble
SDU	Service Data Unit
SDH	Synchrone Digital Hierarchy
SN	Sequence Number
SNP	Sequence Number Protection
SP	Switching Point
TC	Transmission convergence
UNI	User Network Interface
VC	Virtual Connection
VCC	Virtual Channel Connection
VP	Virtual Path
VPC	Virtual Path Connection
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
VBR	Variable BitRate

Chapter 1
Introduction

This report is the result of the work done during my graduation period in the digital information systems group of the department of Electrical Engineering at the Eindhoven University of Technology.

One of the areas of research in this group is Broadband Integrated Service Digital Networks(B-ISDN). B-ISDN is a future telecommunication network. A network consists of nodes and links. Links are the media on which the information is transferred from one node to another. The nodes are switching points at which the information is switched from an incoming link to an outgoing link. Tomorrow's telecommunication networks must deal with a rapidly increasing number of services. What used to be done by separate networks, like a telephone and a cable TV network, will be integrated in one overall network. For this purpose B-ISDN is developed.

The technique used to transfer information and all aspects involving the switching and multiplexing of signals are included in the term transfer mode. The asynchronous transfer mode(ATM) is the most promising transfer mode for B-ISDN. ATM is a connection oriented transfer mode by which information is transferred in packets of a short fixed size. These packets are named cells. A virtual connection is a concept to describe the unidirectional transport of cells between one end(source) and another end(destination) in an ATM based network. The connection is named virtual because the links used in an ATM based network are shared by other virtual connections. Each cell has a label. This label is used to multiplex several virtual connections on one link. Some aspects of ATM which are important to this project are reviewed in chapter 2.

One of the research projects in the digital information systems group involves the study of the performance of a virtual connection in an ATM based network. What is meant with performance and which performance measures are significant with respect to a virtual connection in an ATM based network is discussed in chapter 3.

One way to evaluate the performance is analytic modeling. Within the digital information systems group an approximate model for the end to end performance of a connection in an ATM network is being developed. Like any other model such a model is based on assumptions. The assumptions are necessary to make the analysis less complex. It is desirable to validate the assumptions made. Which is the bases of existence of this project.

This project involves a performance evaluation with simulation. In figure 1.1 the schematic place of the simulation study is illustrated. From the connection in an ATM based network an approximate performance model is defined. The assumptions made by analytic modeling can be validated by simulating a connection with a model where these assumptions are not made.

The bottleneck with simulation is the following: Performance requirements are that extreme that simulation runs to evaluate these performance measures are exceedingly long. This project involves the study of methods to cope with this problem. This bottleneck with

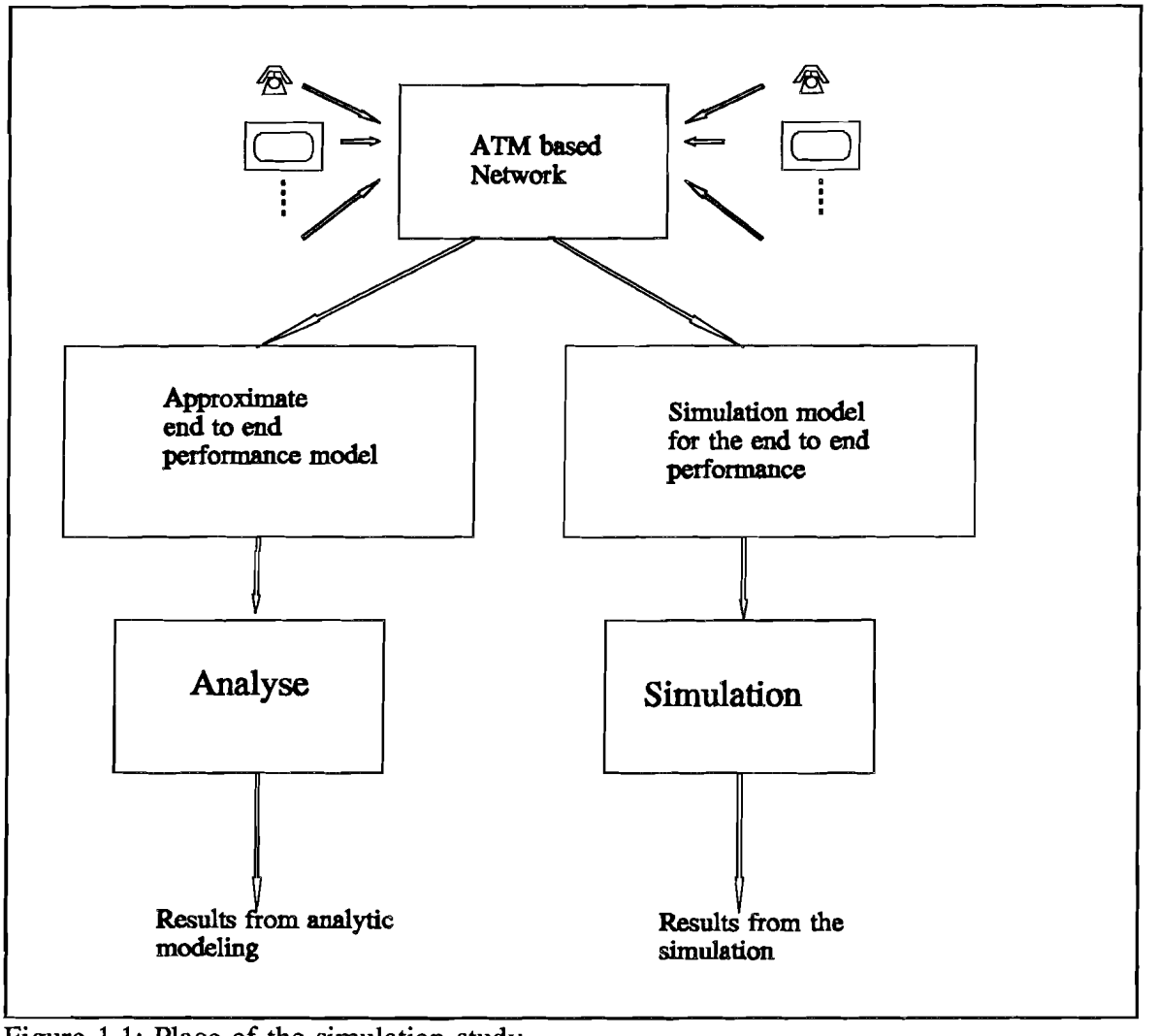


Figure 1.1: Place of the simulation study.

simulation, and the methods to cope with it will be discussed in chapter 4.

One method is applied and implemented. The working and the aspects of implementation will be explained in chapter 5. With the developed software, simulation runs are made. Chapter 6 reviews the results and the analysis of the results. Finally chapter 7 contains the conclusions and recommendations.

At this place I would like to thank the digital information systems group for the support given. Especially I would like to thank professor F. van den Dool who made this project possible, and ir M.J.M. van Weert and ir B. J. van Rijnsoever for their useful discussions and their continuous support. Furthermore I would like to thank the people at the faculty of Industrial Engineering who made it possible to use their simulation software tools.

Chapter 2
Asynchronous Transfer Mode

2.1 Introduction

The need for high bitrates in networks and the use of the optical fibre as transmission medium has led to the development of Broadband ISDN(B-ISDN). Because of the need for communication services of all kind, B-ISDN has become important. ATM is the most promising transfer technique for B-ISDN. The object of this chapter is to give the reader sufficient knowledge over the working of the ATM with respect to the performance of a connection in such a network.

The organization of this chapter is as follows. The next paragraph explains the basics of ATM and the types of connections. Paragraph 2.3 gives information about the protocol reference model of ATM. An network consists of several nodes at which information is switched into different directions. Paragraph 2.4 examines the switching function within ATM. The last paragraph reviews the important topics of this chapter.

2.2 The transfer mode ATM

2.2.1 ATM as transfer technique

A transfer mode is a technique used to transfer information including all aspects involving the switching and multiplexing of signals. This set of agreements determines in what way the information should be provided to the network, plus the condition of the delivered information at the destination.

Within B-ISDN two transfer modes can be used: STM(Synchronous transfer mode) and ATM(Asynchronous transfer mode). The terms synchronous and asynchronous tells us something about the way signals are multiplexed. With the synchronous mode several signals are multiplexed in a fixed way. On a time scale every signal has a deterministic position. This in contrast with the asynchronous mode by which signals are multiplexed on a random basis.

Few books are written over ATM at the moment, one book that covers ATM in detail is [Pry91]. Some overview articles are [Bou92] and [Jun91]. ATM is an asynchronous transfer mode, so the signals are multiplexed asynchronously. ATM is a combination of packet switching and circuit switching. Information is transferred in packets of a short fixed length named cells. A cell consists of an information field and an header.

Information field:

This field is for user information and the size is 48 octets.

Header:

The header is a label to the information field and contains network and routing information.

Because these cells are generated on demand, any bitrate up to a maximum can be achieved. The mode is connection oriented which means that for every transmission a connection has to be set up first. The error control is not performed on a link-by-link basis, like conventional packet switched modes, but is left to an end-to-end error control. This means that no effort is spend on error control in the nodes. Therefore switching speed is improved. The end-to-end error control is possible due to the fact that optical fibre links are assumed. The optical fibre has the characteristic of very low error probability in comparison with copper wired links.

2.2.2 Statistical multiplexing

Statistical multiplexing is not the same as asynchronous multiplexing. Statistical multiplexing is a technique easily applicable on an ATM environment with bursty sources. The technique uses the burstiness of offered traffic by multiplexing more traffic onto a link than the sum of the separate source peak rates would allow.

The object of statistical multiplexing is to gain bandwidth. If this gain is achieved by ATM connections is still under study. One problem with statistical multiplexing is that congestion might occur. Especially when a number of bursts take place at the same time. The disadvantage of the application of statistical multiplexing is that a complex congestion control is needed.

2.2.3 ATM connections

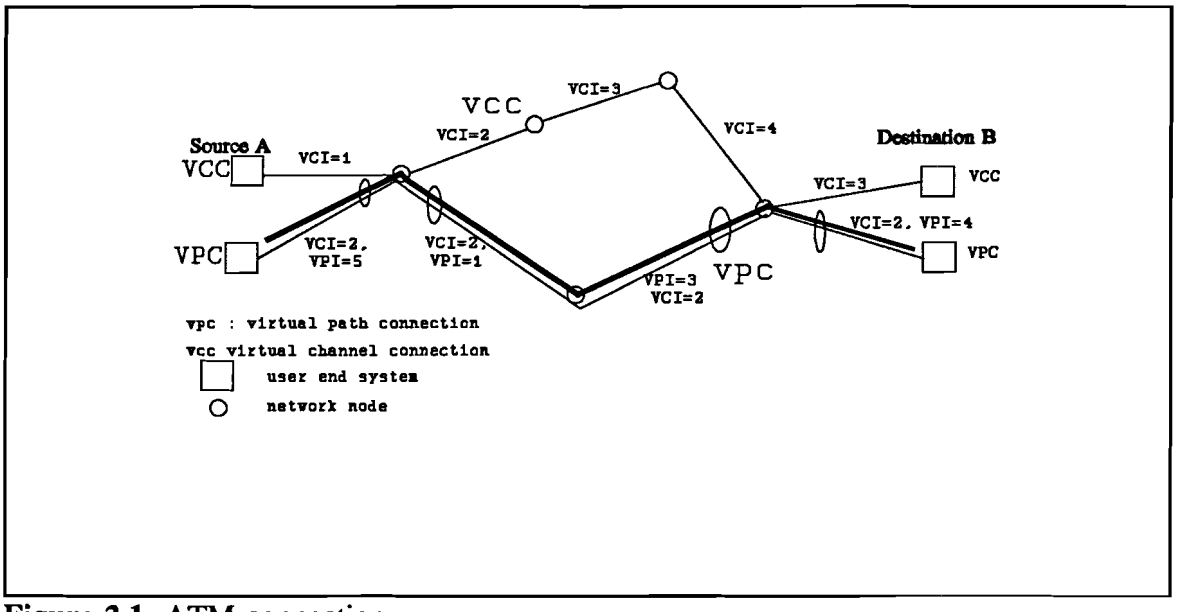


Figure 2.1: ATM connection.

ATM is a connection oriented transfer mode. In the following will be explained what sort of connections can be built. The example connection of figure 2.1 will be used. In this figure a connection is required between source A and destination B.
There are two levels of connections:

- Virtual Channel Connection(VCC)
- Virtual Path Connection(VPC)

Virtual Channel Connection(VCC)

A VCC is a concatenation of virtual channels. A virtual channel is a link between two switching points(SP). The CCITT defines a virtual channel as follows:

'A concept used to describe **unidirectional** transport of **ATM cells** associated by a common **unique identifier** value'.

Let us highlight some keywords in this definition:

Unidirectional	The channel is unidirectional, data flows from source SP to destination SP.
ATM cells	The unidirectional flows consist of a set of ATM cells.
Unique identifier	The cells belonging to the same unidirectional data flow have the same unique identifier within the link. This identifier is named the virtual channel identifier(VCI).

Due to the uniqueness of the VCI within the link, several data flows can be multiplexed. The multiplex technique used therefore is named Label Division Multiplexing(LDM). Each cell is labelled by its VCI. The identification at the destination is done using that label. In figure 2.1 an example of a VCC is depicted. The VCC is a concatenation of the virtual channels with the VCI: 1,2,3,4 and 3. The VCI is only unique within a link. The value of the VCI is replaced in every node.

Virtual Path Connections(VPC)

It is no surprise that VPC is a concatenation of virtual paths. VPC is like VCC an end to end connection. The CCITT defines the VP as follows:

'A concept used to describe transport of cells belonging to virtual channels that are associated by a common identifier value.'

A VP consists of a bundle of virtual channels between two SPs. The virtual channels belonging to that VP are associated by the same VPI. A VPC is a concatenation of VPs. Since in a SP only the VPI changes, the VCIs shown uniqueness within a VPC. In figure 2.1 a VPC is depicted(thick line). From this bundle of virtual channels only one is depicted with VCI=2. The VCI is unique within the VPC. Only the VPI is replaced in every node. Virtual paths are used to switch bundles of virtual channels.

2.3 ATM protocol reference model(PRM)

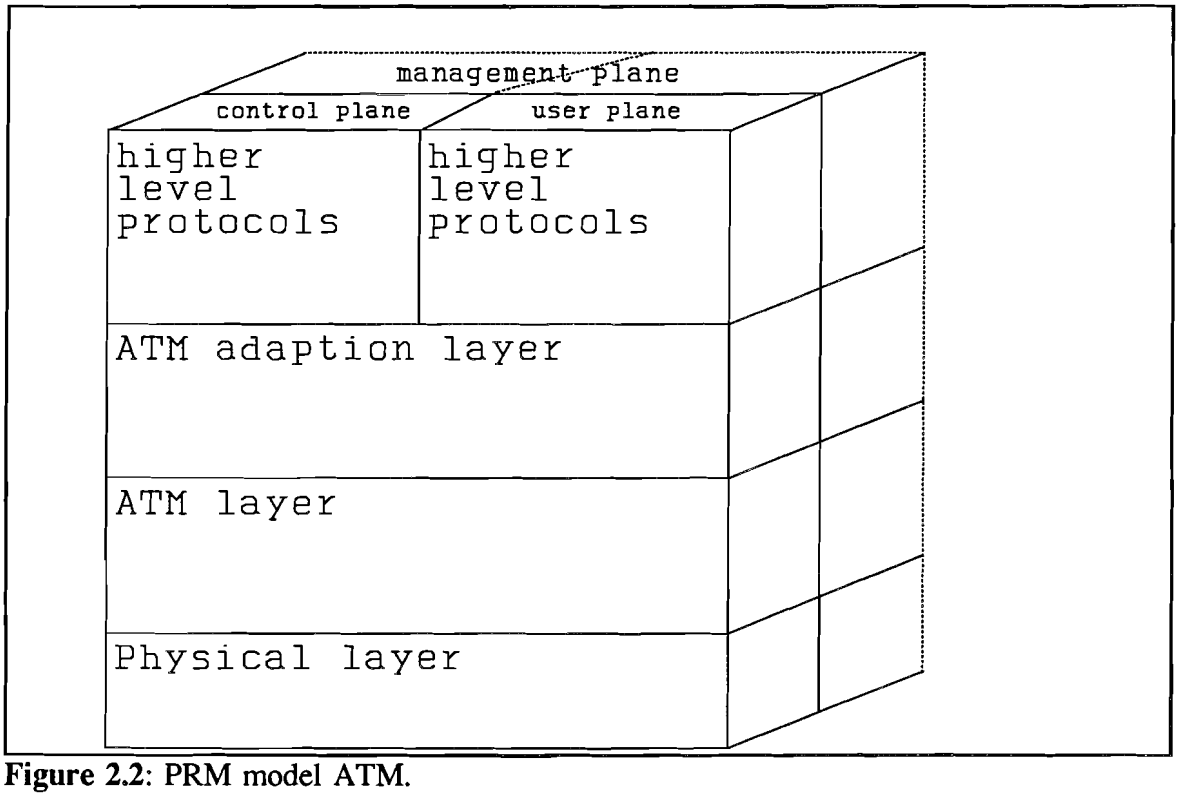


Figure 2.2: PRM model ATM.

In ATM besides horizontal layering a concept of separated planes is used for the segration of user, control and management functions. The ATM PRM is shown in figure 2.2. The PRM contains three planes:

- **User plane** to transport user information.
- **Control plane** for signalling functions.
- **Management plane** to maintain the network and to perform operational functions.

For each plane a layered approach is used. Next the three lower layers are discussed using figure 2.2.

The three lower levels are:

- Physical layer(PL).
- ATM layer
- ATM adaption layer(AAL).

In figure 2.3 the three lower layers are envisioned. Within the layers, functions performed by it have been depicted.

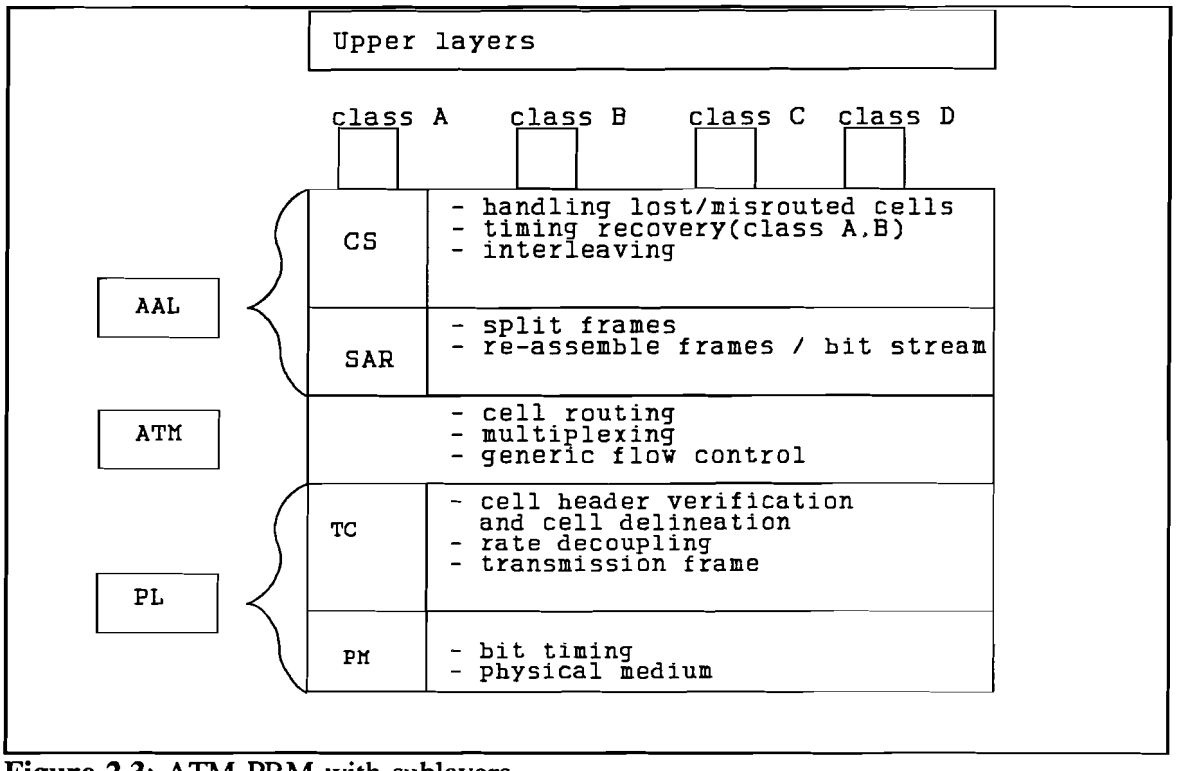


Figure 2.3: ATM PRM with sublayers.

2.3.1 Physical Layer

The function of the PL is two fold:

- Transport of valid cells.
- Delivery of timing information for upper layer services.

The last service is needed since the time relation, between input and output, is disturbed by the network. When timing information is needed, this information has to be provided by the lowest level.

The PL consists of two sublayers: the transmission convergence and the physical medium. The TC layer accept cells from the ATM layer and packs them into the formats for transmission. This is named transmission frame adaption. On the receiving side the TC extracts cells from the PM layer and verifies cell headers. Valid cells are passed to the ATM layer. The physical medium provides bit transmission and physical access to the transmission medium.

2.3.2 ATM layer

The ATM layer is independent of the PL and equal to all services which make use of it. The basic functions of the ATM layer are:

- The multiplexing and demultiplexing of cells of different connections onto a single cell stream.
- The translations of the identifiers at the ATM switches and cross connects.
- Extraction of the payload of the cells received from the PL. And header generation for prefixing on the payload.
- At the UNI a flow control mechanism may be implemented which makes use of the GFC bits.

2.3.3 AAL layer

The AAL consists of two sublayers. The SAR layer transforms data units so that they fit into cells(segmentation), and conversely reconstruct data units from the payload of the cells. The convergence sublayer performs additional functions.

ATM is very flexible, due to this flexibility a variety of services can be offered. To prevent chaos in services offered, classification of services is defined by the CCITT. The services are classified into four classes according to the following three basic parameters:

- The timing relation between source and destinations. This timing relation can be required or not required.
- The bitrate which can be a constant bit rate(CBR) or a variable bitrate(VBR).
- Connection mode. The mode can be connection oriented or connection-less

With these parameters the services can be classified into four classes listed below

Class	Timing	Bitrate	Connection mode
class A	required	constant	connection oriented
class B	required	variable	connection oriented
class C	not required	variable	connection oriented
class D	not required	variable	connection-less

These classes represent different types of services:

Class A	Circuit emulation
Class B	Variable bit rate service with time synchronisation between sender and receiver. Examples are video and high quality audio.
Class C	Connection oriented data service.
Class D	Connection-less data service.

Class A: Circuit emulation

The services are connection oriented. Before information is transferred(at a CBR) a connection set up must take place. In addition a timing relation has to be transferred.
The SAR layer adds a sequence number for detecting cell losses. For error protection on that sequence number 4 bits sequence number protection(SNP) are added. An undesired effect is delay variation. This can be handled with playout buffers. For providing the timing relation an end to end synchronisation is needed. This is done by allowing the receiver to recover the clock used by the sender to transmit the signal. Three solutions to this problem are envisioned in [Bou92].

Class B: VBR service with timing relation.

The difference with class A is that the bitrate is variable. The functions of the AAL are the same except that handling the synchronisation is more complex.

Class C: Connection oriented data services.

In this class of services a connection is also set up before the data transfer phase starts. There are two characteristics that qualify the services in this class:

a Non assured, assured

b Message, streaming mode

ad a: In the assured mode the delivery of every service data unit(SDU) at the destination without errors is guaranteed. Flow control is provided between the endpoints. In the non assured mode some SDU may be lost or delivered incorrect.

ad b: The message mode accepts only a fixed size of SDUs, in contrast with the streaming mode by which SDUs of variable size are accepted.

Class D: Connectionless data service

It may seem a bit strange for a connection oriented transfer mode to have a class of services that are connection-less. For connection-less data services predefined ATM connections are used over which connection-less frames are sent.

Two methods of connectionless data service are described in [Bou92]:

a. Broadcast connection.

This method makes use of a shared broadcast virtual connection(VCC or VPC). Every destination reads all segments and picks the ones which are destined for. The drawback of this method is that a complex operation of the AAL is needed and no advantage is taken of the switching at ATM level.

b. Connectionless servers.

This method uses a connection less server(CLS) which routes the frame to its destination or to another CLS. The disadvantage is that a CLS is needed.

2.4 ATM switching

In this section aspects of switching in ATM networks will be discussed. An ATM switch performs three basic functions:

- header translation
- routing
- queuing

Paragraph 2.4.1 explains the header translation function and the routing aspect. Queuing in ATM switches is unavoidable. The reason for queuing and queuing disciplines will be discussed in 2.4.2.

2.4.1 Header translation and routing

Header translation

ATM switching is performed in two levels. On a virtual channel link level and on a virtual path basis. With a virtual channel level the VCI from each cell is identified. The second level is on a virtual path basis, now only the VPI is identified. Both VC switching and VP switching are based on the same principle which is depicted in figure 2.4.

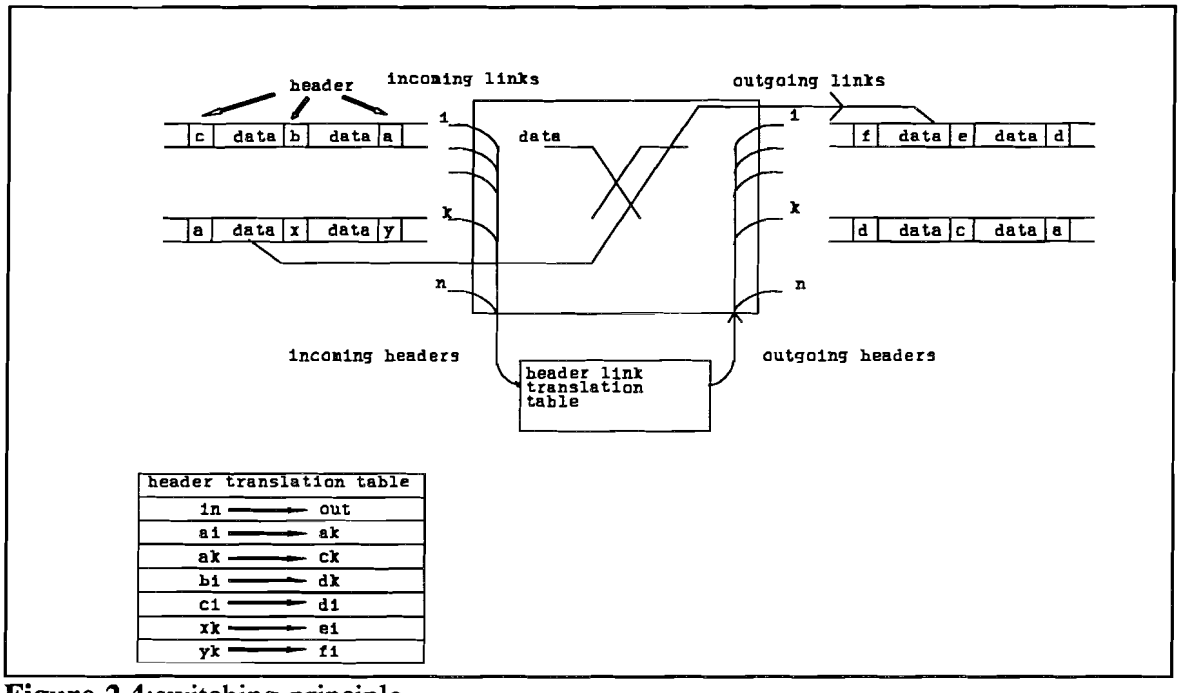


Figure 2.4:switching principle.

The header of each incoming link is translated according to the translation table. This translation table determines from the incoming cell and the link, the new header and to which link the cell must be switched. Only the header is not enough since it is only unique within the link. The cell, including the old header, is switched according to the translation table. The old header is replaced by the new one and the cell is transmitted on the outgoing link. In figure 2.4 an example is depicted. The cell with header x on incoming link k, must be switched onto outgoing link 1 with header e.

Routing methods

With routing is meant the way information is routed internally from the inlet to the outlet[Pry91]. There are two important questions that arise with this aspect of switching. First the place, where routing information is present. This can be in the network with routing tables or at each cell by placing a routing tag at the front of each cell. The first network based method is more flexible. Especially point to multipoint connection can easily be realized with this method. The disadvantage is the overhead involved and the fact that routing information has to be available on several places in the network. This requires extra management.

Secondly the routing information must exist for the duration of the connection or only for one cell(connection-less). If chosen for connectionless the cells can be routed on a random basis through the ASF. In this way the burstiness of cells will be distributed over several paths. There is one problem that must be taken care of: The cell integrity should be recovered before the cells are transmitted.

The decisions based on these two questions divide the routing methods in four types as depicted below.

	Routing tag	Routing table
Connection mode	type I	type III
Connectionless	type II	type IV

The routing methods II and III are most natural. When the choice is made for connection mode, it is more convenient to choose for routing tables. If chosen for connectionless routing, then routing tags are more natural. Because routing tables should be implemented on all possible internal switching points.

2.4.2 Queuing Strategies

In ATM non deterministic multiplexing is applied. The arrivals of cells are unscheduled. The situation where more then one cell arrive in the same timeslot for one output is common. Therefore buffers are necessary for queuing the cells. The queuing is done in the switch. Several queuing strategies can be applied. They differ in the place where cells are buffered and the sharing of buffers. First some strategies are discussed. At the end an overview is given.

There are four strategies for buffering cells:

1. Input queuing
2. Output queuing
3. Central queuing
4. Distributed queuing.

Input queuing

In this strategy the buffers are placed at the incoming links. Separate buffers are provided for each incoming link. This is illustrated in figure 2.5. Suppose FIFO buffering is applied and k cells arrives in one cell cycle for one specific output. Assume also that the queues are empty. Then one cell is transferred and k-1 cells are put into the queue. The switching speed might be the same as the arrival rate at the incoming links. When another cell arrive at a link where cells are in the queue it's placed at the end of the queue, regardless if it could be switched to an unoccupied output. This type of blocking is named head of line blocking(HOL). This results in a limited throughput [Kar88].

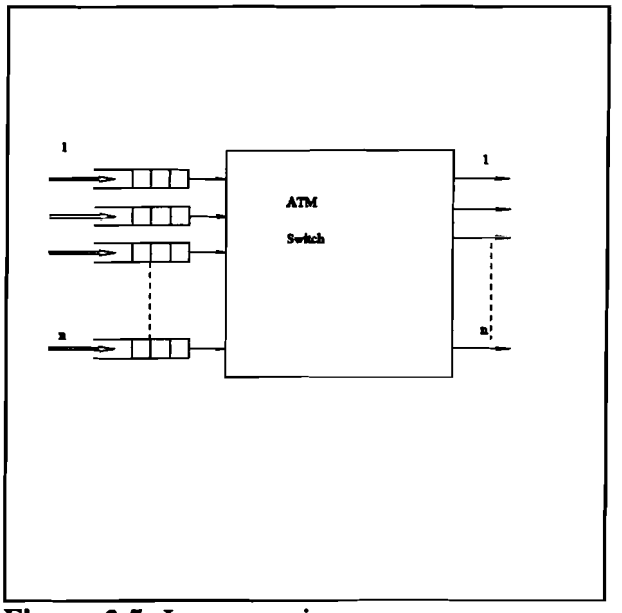


Figure 2.5: Input queuing.

Output queuing

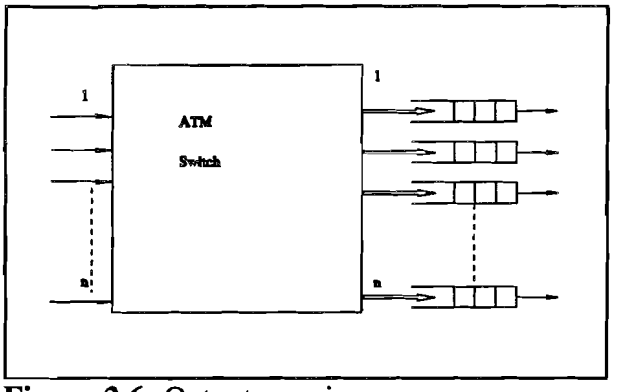


Figure 2.6: Output queuing.

Output queuing is illustrated in figure 2.6. With output queuing all buffering is done at the outgoing links. Each outgoing link has a FIFO buffer. Since no buffers at the input are provided and there might arrive at all N inputs cells for one output, the switch fabric has to operate N times faster than with input queuing. In one cell cycle all incoming links are served. If k cells are switched to the same output then one cell can be transmitted and the other k-1 cells are put into the queue. Because arriving cells do not interfere with cells going to different outputs there is no HOL blocking. Due to this the throughput with output queuing is for large N optimal [Mor87]. The disadvantage is the speed up of the switch.

Central queuing

This is also referred to as completely shared buffering as depicted in figure 2.7. This strategy makes no use of separate buffering, all queued cells are in the same buffer space. The buffer capacity is reduced compared to separate buffering. The allocation of queues can be organized in such a way that the behaviour is similar to output queuing. Because of the shared buffering a complex mechanism is needed for administration and control.

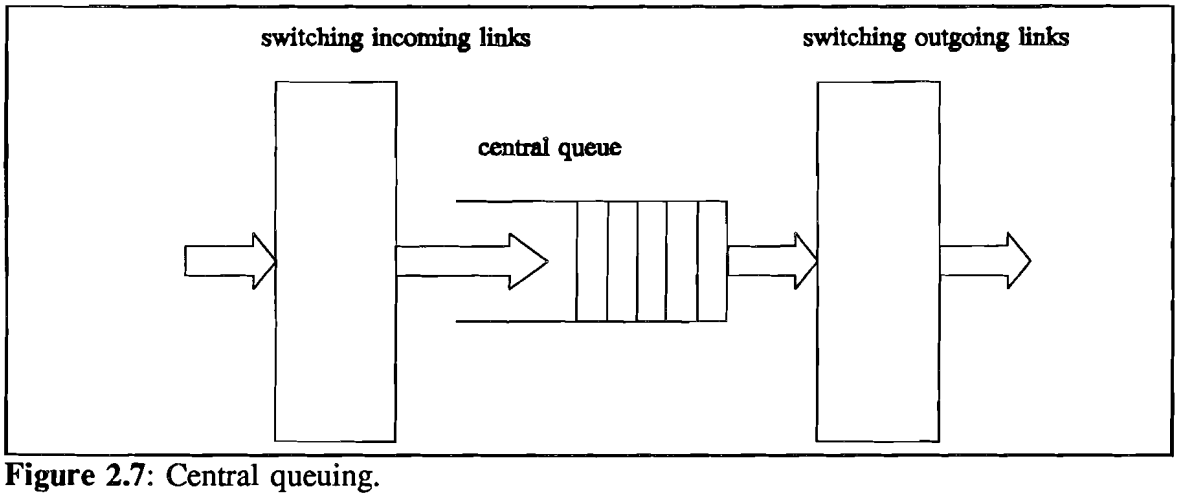


Figure 2.7: Central queuing.

Distributed queuing

In this method the queue is distributed over the switch architecture. The switching elements of the fabric possess parts of the queue. When the buffers at a specific switching element is full this element sends a stop sending signal to the preceding switching element. This technique of flow control is named back pressure technique.

2.5 Topics on ATM

ATM is the most promising transfer mode for B-ISDN. It is a connection oriented transfer mode which means that before information is send the connection has to be build up first. Two levels of connection are defined in ATM:

Virtual channel connection
Virtual Path connection.

Cells are associated to a virtual channel by an unique identifier in the header. Due to the uniqueness of the identifier, cells belonging to different virtual channels can be multiplexed onto a link.

ATM is very flexible, due to this flexibility a variety of services can be offered. These services are classified according to the timing relation, bitrate and type of connection. The switching in ATM based networks involves three functions:

- Header translation
- Routing
- Queuing

Queuing in ATM switches is unavoidable. To perform queuing several strategies can be used. An overview is given below:

Input queuing:
- Simple queuing structure.
- Throughput limited.

Output queuing:
- High switching speed needed.
- Optimal throughput/delay performance.

Central queuing:
- Achieves optimal throughput/delay performance.
- Small total bufferspace for large N(incoming links).
- Complex control mechanism needed.

Distributed queuing:
- Flow control needed.
- No speed up of switching speed needed.

References

[Bou92]; Le Boudec J. Y., "The asynchronous Transfer Mode", Computer networks and ISDN systems, 24(1992), pp 279-309.

[Pry91] de Prycker M.,"Asynchronous Transfer Mode, Solution for broadband ISDN", Ellis Horwood, New York 1991.

[Kar88] Karol M.J. et al, " Queuing in High Performance Packet switching networks", IEEE journal of selected areas of communications, Vol 6 No 9, December 88, pp 1587-1597.

[Jun91] Jungok J. et al, "Survey of traffic control schemes and protocols in ATM networks", Proceedings of the IEEE, Vol 79, No 2, 1991, pp 170-189.

[Mor87] Morgan P et al, "Input versus output queuing on a space division packet switch", IEEE transactions of communications, Vol 37, No 12, December 87 pp 1347-1356.

Chapter 3
Performance of a Virtual Connection

3.1 Introduction

To study the performance of a virtual connection, performance has to be defined first. The important question hereby is: What parameters of a virtual connection give information about the degree of quality? These parameters are named performance measures and discussed in paragraph 3.2. Another goal of a performance evaluation is to study the effect of various parameters and their interaction with regard to the performance measures(3.3). The evaluation techniques and the role of simulation is discussed in 3.4. The modeling needed to evaluate the performance is explained in 3.5.

3.2 Performance measures

Performance is an important part of the three-way relationship in teletraffic between:
- Traffic
- Capacity
- Performance.

There are two kinds of traffic on a virtual connection. One is the traffic from source to destination. Traffic sources of the second type generated by the other sources of connections on the same links. The last type of traffic is referred to as background traffic.

Capacity of a virtual connection has two aspects. The first aspect is the capacity of the link on which the cells are multiplexed. This link capacity is shared with other connections. Secondly the switches must have buffers for queuing cells. The size of these buffers is the second aspect of capacity.

Performance is the degree in which the connection satisfies the need and requirements of the users. The performance measures should provide information how the connection performs. The value of the performance measures should represent the degree in which the connection fulfils the requirements.

The following two paragraphs discusses cell loss and cell delay as possible performance measures.

3.2.1 Cell Loss

A connection is said to perform good if it matches our expectations. One expectation is that if a cell is sent it reaches the destination. The probability that a cell is lost provides information about the performance of the connection. Before discussing measures of cell loss, the definition and causes are discussed. When is a cell considered to be lost? A cell that never reaches its destination is lost. Also lost are cells that did reach the destination but of which the information can not be recovered. A requirement for a cell to be lost is that it was sent. A second requirement is that the information transferred by the cell can not be used by the destination.

Cell loss can have several causes. The most significant causes are listed below.
- Defects in hard & software.
- Misrouting.
- Buffer overflow.
- Transmission error

In this project only cell loss caused by bufferoverflow is envisioned. Other types of cell loss are out of the scope of this project.

It is not sufficient to know the value of the cell loss probability. Important is also the cell loss distribution. Three components are examined:
1. The distribution of cell losses in time.
2. The distribution of cell losses over the sequence.
3. The distribution of cell losses over the different nodes in the network.

These three distributions answers the questions when, which and where. These answers provide information on the question why.

ad 1. Distribution in time

Not only the frequency in which cell loss occurs gives information about the performance, also the distribution in time provides information. This distribution should give insight in **when** cell losses occur in relation with other cell losses. The distribution tells us something about the burstiness of cell losses. Two situations are depicted in figure 3.1. Both situations have 5 cell losses. The difference is when they occur.

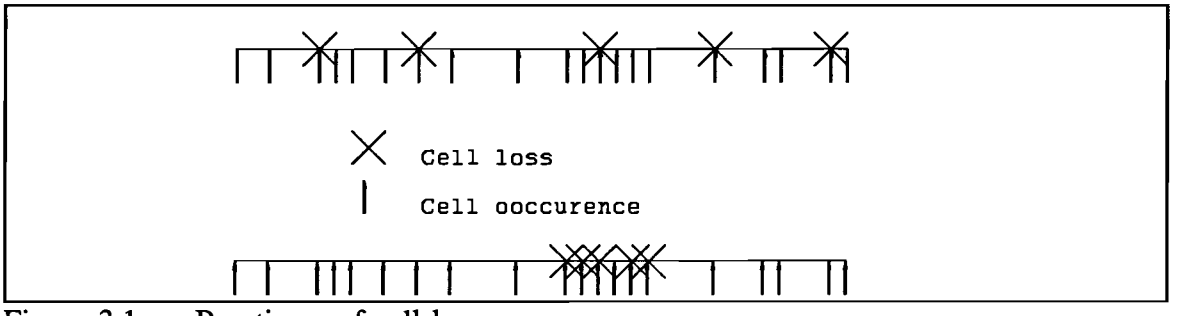


Figure 3.1 Burstiness of cell losses.

ad 2. Distribution over the sequence of cells

This distribution is closely related with the time distribution. The cells send from source to destination are send in a sequence. Cell loss occurrences in relation to the other cells of the virtual connection give information about **which** cell is lost. If a burst of cells occur, as in figure 3.1, the distribution over the sequence will tell if subsequent cells are lost or if cells between a burst of errors are correctly transmitted. This information is not provided by the distribution of time.

ad 3. Distribution over the nodes

This distribution answers the question **where** the cells are lost. The place of cell loss occurrences in a connection give information over the performance of the nodes. A second aspect is that depending on the traffic mix, correlation might occur between cell losses at successive nodes. The distribution of the cell losses over the nodes in combination with the time of occurrences should give insight if there is such a correlation.

3.2.2 Cell delay

From a connection the delay provides also information about the degree of quality. A definition for cell delay is: The delay time during the transfer from point A to point B over a virtual connection.

The cell delay consists of four components:

1. Processing time in switching facilities.
2. Propagation time in links.
3. Transmission time.
4. Queuing delays in switching facilities.

The first three components are fixed and easy to predict. The sum of these three delays are longer than the delays caused by queues. The variance of the delay is caused by the queuing delay. The variance of the delay is also referred to as delay jitter. The importance of cell delay as a performance measure depends on the type of service. Two classes of services must be envisioned:

a. Time relation not required.
b. Time relation required.

ad a. Time relation not required

These services belong to classes C and D (paragraph 2.3.3). Hereby is delay less important as performance measure since no time relation is required. There is a delay threshold at which higher layer protocols assume a cell to be lost. The probability of exceeding this delay, P[delay > c], gives a component of the cell loss probability. But the queuing delay is significant smaller than the other components of delay. Therefore such a situation is not likely to occur.

ad b. Timing relation required.

The variance of the delay results in the disturbance of the time relation between cells, since the delay varies for the different cells of the VC. At the destination the time relation between cells must be recovered. This is done by applying a playout buffer at the end of the connection[Bou92]. If the delay is within the specifications, the playout buffer is able to produce a periodic stream. But the queuing delay can cause variation of the delay of cells that can not be recovered. Two faults might occur. The first is buffer overflow. If cells arrive faster than the outgoing rate, the playout buffer becomes full. If more cells should be queued as necessary, buffer overflow occurs and cells are lost. The second one is referred to as buffer underflow or starvation. In this situation no cells are available, caused by delay, when the playout buffer should sent a cell.

The variance of the delay is a performance measure of a virtual connection if services with a timing relation are provided. The queuing delay is a cause of buffer over,- and underflow at the end of the connection. Due to this cause the required timing relation can not always be restored. The degree in which the timing relation can be restored is a performance aspect. This can be measured by the probability of overflow and starvation.

Another performance measure of the delay jitter is the distribution of the interval lengths is a performance measure of queuing delay. This distribution should be measured at different places in the connection to provide information on the question where.

ATM connections operates in bursty traffic environments. Delays of a bursty traffic source require special treatment. The transfer delay is compensated by a playout buffer. The timing relation is restored at the end. This timing relation is only valid within a burst. Therefore variance of the delay relative to the arrival of the burst becomes important.

3.3 Parameters of influence

Important to the study of performance are the parameters that influence the performance. These parameters can be classified into two categories:

a. Architecture of the virtual connection
b. Traffic

ad a. Architecture

Parameters of influence of the architecture are:

- number of nodes
- cell loss priority mechanism
- queuing strategy
- buffer capacity

When focusing on buffer saturation and queuing delay, the nodes are the places where these events take place. Therefore the more nodes are in the connection the more the performance will decrease.

The cell loss priority mechanism has not been standardized yet. The mechanism has one bit available in the header of each cell. One possible implementation is to set this field to zero for the cells which need quaranteed delivery and to set it to one for the cells which are droppable. When congestion occurs, cells whose priority field is set, are dropped first [Jun91]. It is obvious that this affects the cell loss probability and the delay. The degree in which it improves the performance is for further study.

The values of the performance measures depend on which queuing strategy is used. Paragraph 2.4.2 classifies the queuing strategies. For the way this influences the performance, I refer for more reading to [Kar87], [Hlu88] and [Mor87].

The buffer capacity in the switches affect the performance measures directly. Several studies are done to evaluate this relation. For more reading on this subject I refer to [Rob91], [Mur90], [Hlu88] and [Ohb91].

ad b. Traffic

As said the traffic consists of background traffic(crossing and interfering) and traffic on the virtual connection. The total of the traffic sources is named the traffic mix. Different traffic mixes will lead to differences in performance.

Other parameters of traffic that affect the performance are:

- Traffic load
- Burst length
- Burst distribution
- Correlation

Traffic load is the intensity of the traffic on a link. This is mostly expressed relative to the link capacity. In [Rob91] the effect of the load on buffer saturation in an ATM multiplexer is depicted. If the load becomes close to the link capacity the performance decreases.

Bursts are periods in which the load is high. The conversely periods are called silent or low periods. An example of bursty traffic is that of a voice source. When people are talking to each other they don't talk continuously. Speech is highly bursty. Within a call are silent periods when a person is breathing, listening or thinking, and periods when the person actually produces sound. During talk spurts cells are generated. No cells are generated during silent periods.

The performance in relation to burstiness is one of the topics of the performance evaluation study. The ATM environment is usable for statistical multiplexers. If and how this influence the performance of a virtual connection is a object of the performance study of a virtual connection. The parameters of traffic that give insight in the burstiness are:

Burst length distribution

Burst distribution

Cell distribution in a burst.

Correlation gives an indication of the dependency between events or stochastic variables. The generation of cells in an ATM environment is mostly serial correlated. The generated cells are said to be serial correlated if there exists a dependency relation between the probability of cell generation in successive slots. The degree of correlation influences the performance measures and can not be ignored[Ram88].

3.4 The use of simulation in performance study.

To study the performance, the values of the performance measures have to be determined. In general there are three techniques for evaluating the performance:

1. Measuring.
2. Analytic Modeling
3. Simulation.

The first one is possible only when the object already exists. This is not practical with a virtual connection. The other two performance evaluation techniques determine the measures from a model of the object. This model is named the performance model.

Analytic modeling provides more insight in the effect of various parameters with respect to the performance measures than simulation. A drawback of analytic modeling is the complexity of the models. To simplify the model, assumptions have got to be made. To validate these assumptions simulation can be used.

Not only the assumptions but also the results should be validated. This validation is important with both simulation and analytic modeling. Validation is not practical with real measurements. Therefore the simulation besides analytic modeling might be used to validate the results of each other.

Simulation and analytic modeling can be used sequential. This results in a hybrid evaluation technique. Simulation can be imbedded in those parts of the performance study where analytic modeling becomes impractible. Conversely analytic modeling could determine parameters for simulation to reduce the computertime for simulation.

3.5 Performance model

The performance model is used for determining the values of the performance measures. Every model is a simplification of the real world. This simplification is based on assumptions. The values determined from a model are estimators of the real parameters.

Not only a model is needed of the virtual connection but the traffic on the connection must be modeled too.

3.5.1 Traffic models

B-ISDN integrates different services. Therefore different types of traffic sources like data,- voice,- and video sources must be dealt with. Traffic sources are modeled by stochastic processes. To model traffic sources in an ATM environment a slotted structure is assumed. This means that the time is divided into slots of equal length. This paragraph treats the most commonly used stochastic processes by modeling traffic sources.

Bernoulli Process[Jai91]

Generation of data can be modeled with a Bernoulli process. The interarrival time between generation of data is geometrical distributed. The data can be one cell or a number of cells(Batch). The batches are used to model bulk data transmission.

Deterministic Process

This process generates cells with constant interval times.

Batch Process with binomial distributed batchsize.

This process models generation of batches independent from slot to slot. A batch is zero or more cells generated in one timeslot. The number of cells in a batch is binomial distributed.

ON OFF Process

A ON OFF process is used by modeling the behaviour of a voice source. With this modeling two aspects must be considered:

- Cell generation process
- Characterization of the ON/OFF periods.

The ON state is when cells are generated, conversely no cells are generated in the OFF state. A graphic representation of an ON OFF process is depicted in figure 3.2.

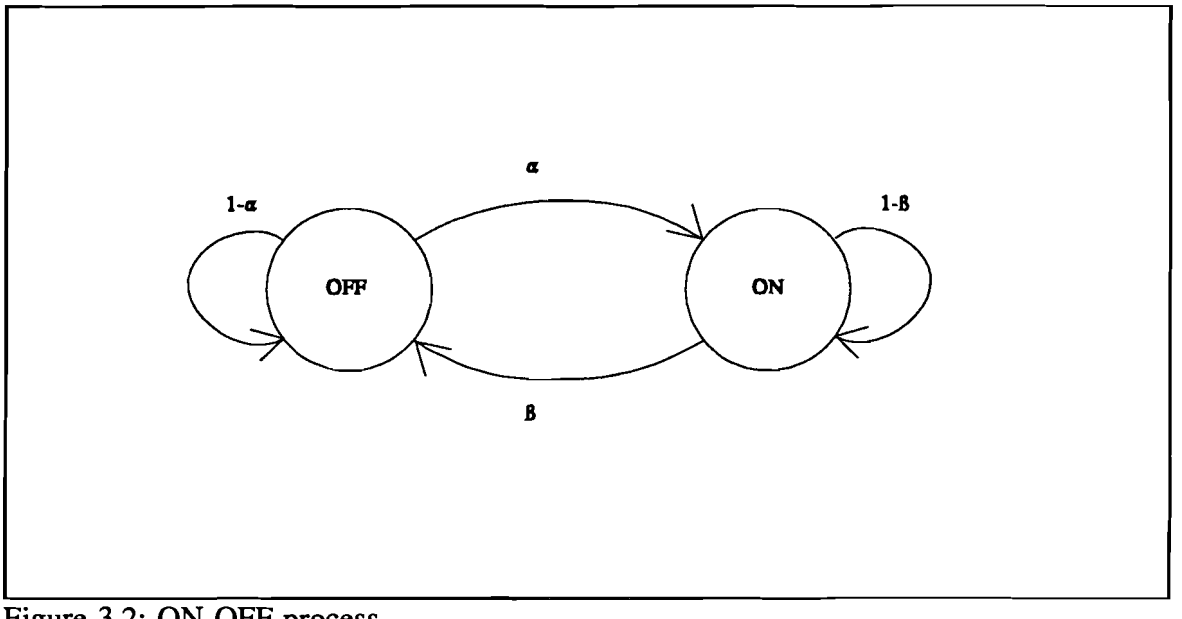


Figure 3.2: ON OFF process.

The transition from the ON to the OFF state occurs with probability ß. The conversely transition from OFF to ON with probability α. The ON and OFF periods are geometrical distributed with mean respectively 1/ß and 1/α. Cells are generated in the ON state only, by a deterministic process[Jun91].

Interrupted Bernoulli Process(IBP)

This process is the same as an ON-OFF source except that in the ON state cells are generated according to a bernoulli process.

Two State Markov Modulated Bernoulli Process(TwoStateMMBP)

To model a superposition of voice or data sources a TwostateMMBP is commonly used. It is a double stochastic batch process where the rate of cells is determined by the state of a two state Markov Chain. Below the state diagram is depicted.

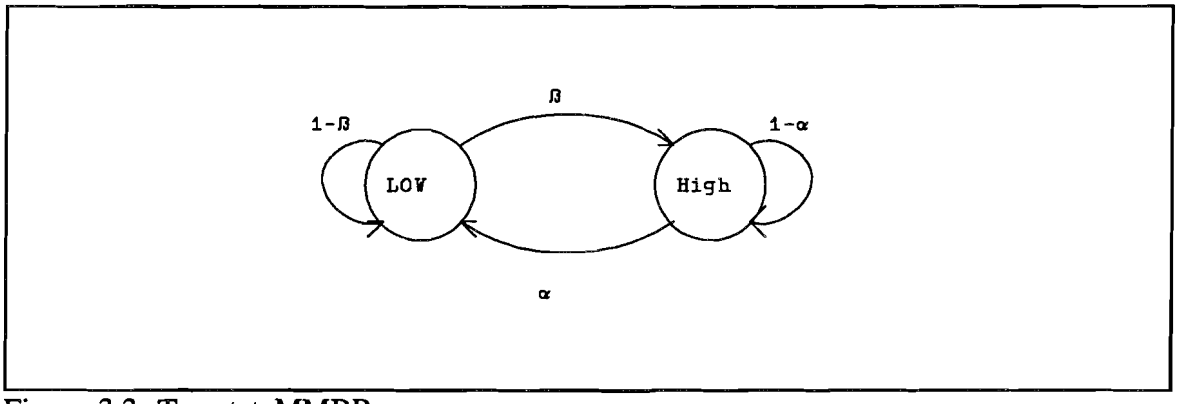


Figure 3.3: TwostateMMBP.

A ON OFF source can be seen as a special case of a TwostateMMBP, as that in the low state no cells are generated and in the High state cells are generated according to a deterministic process.

Analog to the ON OFF process, the IBP is a special case of a TwoStateMMBP. In the High state cells are generated according to a bernoulli process, in the low state no cells are generated at all.

In the figure 3.4 below all sources are envisioned by their behaviour. A arrow represents a cell occurrence. The differences in the length of the arrows represent the difference in number of cells generated per slot.

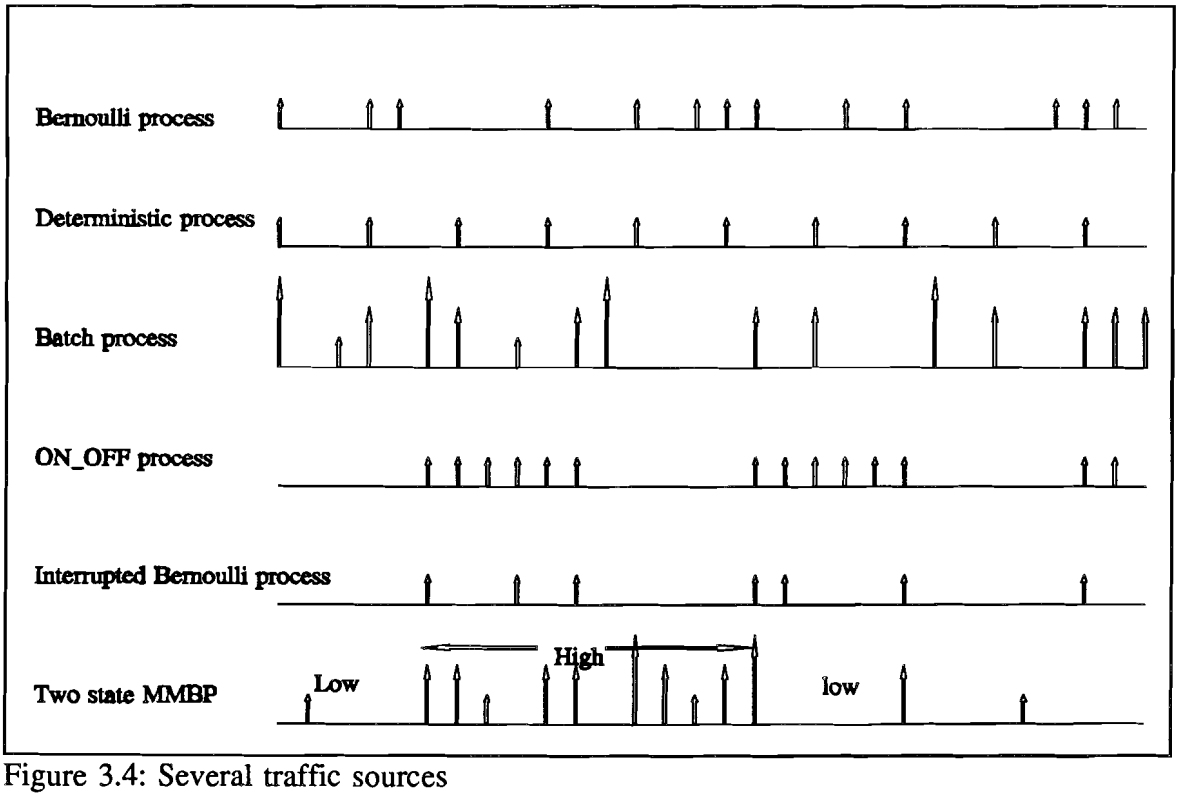


Figure 3.4: Several traffic sources

Several models for traffic can be found in the literature. [Jun91] gives an overview of traffic models. [Grü91] describes an ARMA model for video codecs. The Markov Modulated approach of describing data models is discussed in[Hef86] and used in [Lia86]. [Dia86] describes models for voice sources. [Ram88] includes several models for traffic.

3.5.2 Virtual Connection model

A virtual connection is an unidirectional connection on which several traffic flows are transferred. The traffic from the VC-source to the destination is named VC-traffic. Other traffic is indicated as background traffic. The connection consists of nodes which represent switches. Leaving out details involving modeling of traffic and nodes, the VC can be modeled as below

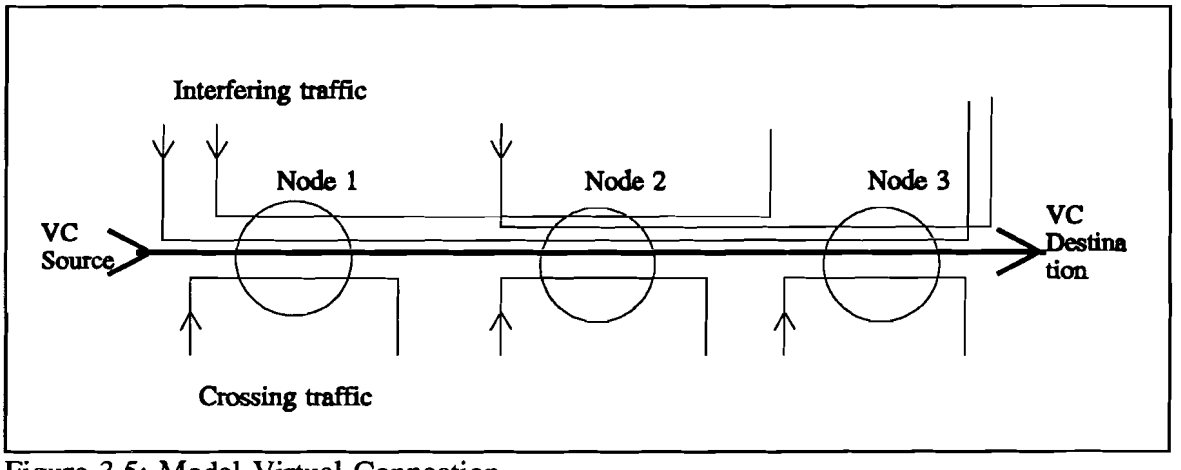


Figure 3.5: Model Virtual Connection

In figure 3.5 the difference between crossing traffic and joining traffic is shown. The routing in this model is fixed. The model in figure 3.5 shows only three nodes. This number is an example. Both performance measures cell loss and delay, focus on loss and delay caused by queues. This queuing aspect of the nodes should therefore be modeled.

A node is mostly modeled by a FIFO queue and a deterministic server as depicted below.

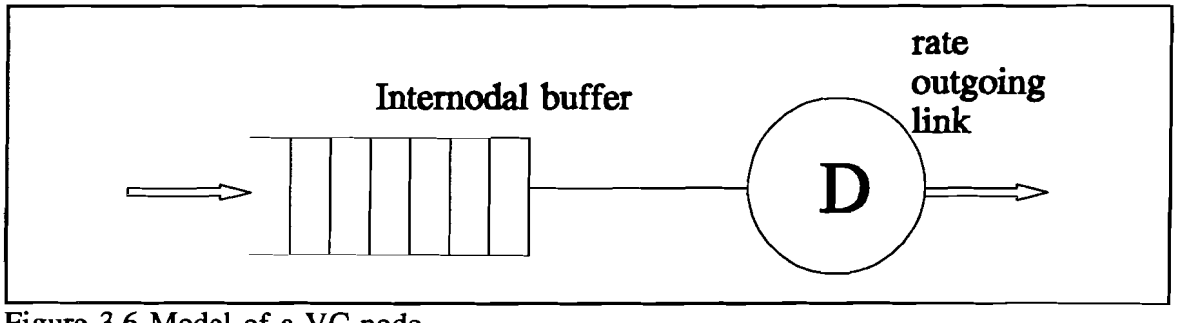


Figure 3.6 Model of a VC node

The rate at which cells are serviced is equal to the rate of the outgoing link. The maximum queue length coincide with the internodal buffersize[Yin90] and [Dia86].

The previous model does not take priorities into account. The choice of a model involving priority schemes depends naturally on the priority scheme used. One possibility is the model depicted below.

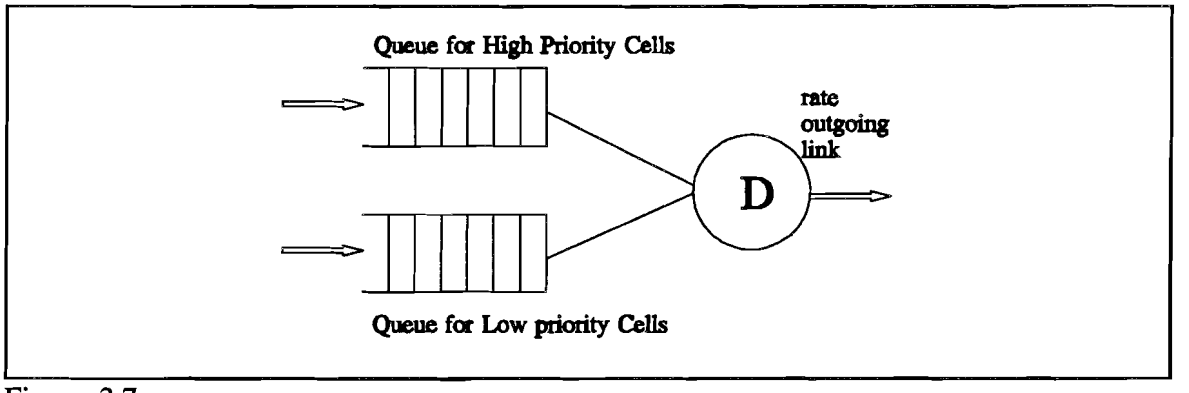


Figure 3.7

Here a node is modeled by a deterministic server with two queues. One for cells with priority and one for cells without. The server first serves cells with priority. When this queue is empty it serves the queue with cells without priority.

In this project the use of priority based schemes is left out. And the nodes are modeled by a FIFO queue with a deterministic server.

3.6 Topics

Performance is the degree in which the VC satisfies the requirements of the user. The probability of cell loss is a performance measure. With cell loss not only the probability is important but also the distribution in the time, sequence and the place. The variance of the cell delay is a performance measure when a service is offered where a time relation between cells is required. The variance of the delay disturbs the time relation between cells. The probability that this time relation cannot be recovered is a performance measure.

The object of simulation in this project is to validate the assumptions made by analytic modeling. Simulation can also used in an hybrid manner together with analytic modeling.

To simulate a VC not only the VC has to be modeled but also the traffic sources. The traffic sources are modeled by stochastic processes. The switches in the VC are modeled by queuing models. The routing of the cells is fixed.

References

[Bou92]: Le Boudec J. Y., "The asynchronous Transfer Mode", *Computer networks and ISDN systems*, 24(1992), p 279-309.

[Dia86]: Daigle J.N., et al, "Models for Analysis of Packet Voice Communication Systems", *IEEE journal on selected areas in communications*, Vol. SAC-4, No. 6, September 1986, p 847-67.

[Grü91]: Grünenfelder R., et al, "Measurement and arma model of video codecs in an ATM environment", TeleTraffic and DataTraffic in a period of change, ITC-13, Elsevier Science Publishers B.V. (North Holland) 1991, p 981-5.

[Hef86]: Heffes H. et al, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance", *IEEE Journal on selected areas in communications*, Vol. SAC-4, No. 6, September 1986, p 856-67.

[Hlu88]: Hluchyj M.G. et al, " Queuing in High Performance Packet switching networks", *IEEE journal of selected areas of communications*, Vol. 6 No. 9, December 88, p 1587-97.

[Jai91]: Jain R.,"The art of computer system performance analysis", John Wesley(1991).

[Jun91] Jungok J. et al, "Survey of traffic control schemes and protocols in ATM networks", *Proceedings of the IEEE*, Vol. 79, No. 2, 1991, p 170-89.

[Kar87]: Karol M. J., et al, "Input Versus Output Queueing on a Space-Division Packet Switch", *IEEE transactions on communications*, Vol. COM-35, No. 12, December 1987, p 1347-56.

[Lia89]: Liao K., et al, "A discrete time single server queue with a two level modulated input and its applications", *IEEE proceedings*, Vol 26.1.1,1989, p 913-8.

[Mor87]: Morgan P et al, "Input versus output queueing on a space division packet switch", *IEEE transactions of communications*, Vol. 37, No. 12, Decmeber 87 p 1347-56.

[Mur80]: Murato M., "Analysis of a discrete time single server qwueuewith bursty inputs for traffic control in ATM networks", *IEEE journal of selected areas of communications*, Vol. 18, No. 13, April 1990, p 496-76.

[Ohb91]: Ohba Y. et al, "Analysis of interdeparture proccesses for bursty traffic in ATM networks", *IEEE journal of selected areas of communications*, Vol. 9, No. 3, April 1991, p 469-76.

[Pry91]: de Prycker M.,"Asynchronous Transfer Mode, Solution for broadband ISDN", Ellis Horwood, New York 1991.

[Ram88]: Ramaswami V., "Traffic performance modeling for packet communication whence, where and whither", Third Australian Teletraffic Seminar, November 1988.

[Rob91]: Roberts J.W. et al, "Recent results on B-ISDN/ATM traffic modelling and performance evaluation", Globecom 1991, p 37.3.1.6-11.

[Yin90]: Yin N., "Voice packet loss destination versus innodal links", *IEEE proceedings*, 1990, p 956-62.

Chapter 4
Simulation techniques

4.1 Introduction

To perform simulation a lot of aspects should be considered. Books that give an overview of these aspects are [Jai91] and [Kle74]. Simulation allows a detailed model to be evaluated. The choice of a detailed model might lead to complex simulation software. The level of detail with simulation is an important aspect which should be considered first. Paragraph 4.2 discusses an appropriate level of detail for simulating a virtual connection. The types of simulation and the terminology used is also reviewed in this paragraph.

Simulation will be used to determine performance measures with a certain confidence interval. Straight simulation of these performance measures of a virtual connection will lead to exceedingly long simulation runs. This problem will be explained with an example in paragraph 4.3.

Techniques to cope with this problem are referred to in the literature as variance reduction techniques and accelerated or efficient simulation methods. Some methods described in the literature are investigated. The techniques studied will be discussed in paragraph 4.4, 4.5 and 4.6. The motivation and reasons for choosing the method applied will be reviewed in the last paragraph. The method that is applied will be explained in more detail in the next chapter.

4.2 Simulation

Level of detail

The choice of an inappropriate level of detail is in [Jai91] given as a common mistake. The possibilities of simulation tempt to a very detailed simulation model, not looking at the purpose of simulation. This results mostly in complex and non traceable simulation software. In paragraph 3.5 a model to determine the performance measures is discussed. A number of aspects are discussed, but not considered as involving the level of detail. Let us review the model where the point of view is now the level of detail.

The traffic is modeled by stochastic processes. The processes generates cells. Several processes should be possible. So the level of detail concerning the traffic sources is that they are modeled by stochastic processes on cell level.

A node is represented by a FIFO queue with a deterministic server. But other queuing models should be possible. The level of detail here is that the nodes are modeled by queuing models.

Another aspect of the level of detail is the output of the simulation. This output will be the performance measures. Which are discussed in paragraph 3.2. When the implementation phase is reached these aspects of the level of detail will be transformed into input output specifications of the simulation software.

Types of simulation

Simulation can be carried out in several ways. First of all a distinction can be made between simulating with software and simulation using hardware. The last type is also referred to as emulation and is above price. Hence this type is not discussed here. There is an article written in which a hardware simulator for an ATM environment is used[Yok91].

Another distinction between simulation language is if the system state is continuous or discrete. Both continuous and discrete simulation languages are reviewed in most books about simulation also in [Jai91] and [Kle74]. The performance model in this project is a discrete state model. Therefore the type of simulation usable is discrete event simulation. This is the opposite of continuous event simulation in which the state of the system takes continuous values. The term discrete does not refer to the time variable. This can be both continuous and discrete. But in the performance model of the VC a time slotted structure is assumed. The type of simulation becomes therefore discrete-event discrete-time simulation.

Two terms are used frequently: State and event. The state of a system is represented by a collection of state variables. In the performance model a state variable is for example the occupance of a queue. An event is defined as a change in the value of one or more system state variables at a point of time. If a cell is put to the tail of a queue, a change in the system state happens. This is an example of an event.

4.3 Bottleneck of simulation

It has been stated before that it takes exceedingly long simulation runs, with straight simulation, to determine the performance measures. How long can best be illustrated with an example. The values in this example are extracted from the article[Vil91].

Suppose we want to evaluate a cell loss probability in the order of 1E-9. So on the average 1 in the 1E9 cells are lost. These low probabilities are common in an ATM environment. Suppose further that 100 cell loss occurrences are needed for the desired confidence interval. Than the number of cells to be simulated is in the order of 1E11. Assuming that 0.1mS computertime is needed for each simulated cell, the time to simulate becomes in the order of 1E7 seconds which corresponds 115 days. It is clear that such a long simulation run is impracticable.

One could wonder if the assumption of 0.1mS for each simulated cell is a bit pessimistic. This value is from [Vil91] and comes from a simulation of one node. With simulation of more nodes, more effort is spend for each simulated cell. It will appear that the 0.1mS is rather an optimistic than a pessimistic assumption.

4.4 Accelerated simulation methods

The previous example stresses the need to accelerate simulation or to make the simulation more effective. [Fro88] and [Cos224] give an overview of efficient techniques of simulation of computer communication networks. Three categories are made in the [Cos224]:

1. Hybrid techniques
2. Extrapolative methods
3. Variance reduction techniques

Hybrid Techniques

Hybrid techniques combine analytic modeling with simulation. These techniques do not accelerate the simulation. Instead they try to prevent the need for simulation by combining it with analytic modeling. This is conversely the object of this project to use simulation to validate analytic modeling.

The other two techniques will be discussed in the two following paragraphs.

4.5 Extrapolative Methods

Extrapolative techniques are applied to the results obtained from simulation. One extrapolative technique using the extreme value theory is applied to an ATM environment. [Dij91] handles the problem of determining the buffersize in an ATM switch for a required error probability, using a quantile estimator. The method makes use of the delay distribution which is extrapolated. The estimator of the buffersize is derived from the extrapolated delay distribution. To explain the assumptions made, some definitions are reviewed below:

P_n: The required error probability.
F: Unknown delay distribution function.
U: The inverse function of 1/(1-F)
X_{Pn}: Variable to be estimated, representing the buffersize needed for the required error probability Pn.
$X_1, X_2, .., X_n$: n observations from the unknown distribution F.

The problem to be solved is to find an estimator such that:

$$1-F(x_{P_n})=P_n \Rightarrow X_{P_n}=U(\frac{1}{P_n}) \quad (1)$$

For the exact working and application I refer to [Dij91] and [Cas88]. In [Dij91] the following assumptions are made to apply this extrapolative technique.

Assumptions

1: $X_1, X_2, ..., X_n$ are independent and identical distributed(F) observations.

2: The buffersize can be derived from the queue length distribution of a corresponding system with infinite queues.

3: The service times are assumed to be deterministic.

4: The arrival process must be continuous, so that the unknown distribution is three times differentiable. This assumption is necessary to apply the quantile estimator.

The method is applied on two queue models the M/D/1 and a two state MMPP/D/1 model. These queue models are simulated. The n samples from the delay distribution are used to extrapolate this distribution from which the needed buffersize is estimated. For estimating the buffersize of the queueing models with a required probability of 1E-9 and a 95% confidence interval, 1E5 samples were needed.

Application to simulation of a virtual connection

The application to a virtual connection causes two problems. One is the assumption of the i.i.d of the observations. This excludes serial correlation of the samples. This is also the case in simulating one node. The second problem concerns the delay distribution. The derivation of the performance measures from the delay distribution is far more complex for a connection than for one node.

One could think that the first assumption is necessary to apply extreme value theory. This is not true. In [Cas88] formula are derived for order statistics of dependent sequences. The joint distribution of the dependent sequences is needed therefore.
So if estimators based on extreme value theory are used, research should be done on the joint distribution of the observations in a virtual connection.

4.6 Variance reduction techniques

Simulation will be used to estimate performance measures. These estimated measures have a certain variance. The variance is used to express the statistical confidence of the estimated measures. If the variance decreases the results become more accurate. One way to reduce the variance is to extend the simulation run. Conversely the time to simulate for estimating performance measures with a certain confidence can be reduced by using methods which reduce the variance. These methods are named variance reduction techniques.

Almost every book about simulation covers the following reduction techniques:

- Antithetic sampling[Kob78].
- Control variate[Kob78] and [Cos224].

These techniques reduce the variance using statistical characterizations but the variance reduction achieved is too small.

There are two techniques left that are described in the literature that have been investigated:

Importance sampling
Restart

4.6.1 Importance sampling

Importance sampling is applied in various ways in different areas of the technique. The working and some variants are reviewed below.

Principle and working

Importance sampling(IS) is a variation reduction technique(VRT). Sampling is the process of gathering samples of a unknown distribution. The idea behind IS is to disturb the sampling process completely by replacing the original process by another one. The distortion is corrected by some kind of weighting of the observations from the new sampling process.

The IS concept can be applied to estimate tail probabilities. The technique gives good results in estimating error probabilities in communicating channels. In [Wes90] an overview of importance sampling applied to tail probabilities is given.

Suppose the following tail probability has to be estimated.

$$P_0 = \int_T^{\infty} f(x)dx \tag{2}$$

To estimate this probability the following estimator is possible:
assumption 1: X_1, X_2 .. X_N are mutually independent random variables distributed with density function f(x).
h(x) is an indicator function which equals one when X contributes to the tail probability.

$$\hat{P}_0 = \frac{1}{N}\sum_{i=1}^{N} h(X_i) \tag{3}$$

With the inequality of Chebychev can be shown that:

$$P(|\hat{P}_0 - P_0| > \epsilon) \leq \frac{1}{N\epsilon^2} P_0(1-P_0) \tag{4}$$

The sampling process is now disturbed by sampling with a different density function of the input variables. Suppose $f^*(x)$ is the disturbed density function. Than the estimator of the disturbed process becomes[Wes90]:

assumption 2: The random variables X_i^* are mutual independent and distributed with density $f^*(x)$.

$$\bar{P}_0^* = \frac{1}{N}\sum_{i=1}^{N} \frac{h(X_i^*)f(X_i^*)}{f^*(X_i^*)} \tag{5}$$

IS results in a variance reduction when f^* is chosen such that:

$$var\left(\frac{h(X_i^*)f(X_i^*)}{f^*(X_i)}\right) < var(h(X_i)) \tag{6}$$

This can be achieved with several choices. Variance reduction can be achieved by sampling from a density function which consists only of the parts that contributes to the tail probability. Further improvements made be more sophisticated choices of f^* can be found in [Wes90].

Two problems arise when applying the IS concept on a simulation of a VC. First the assumptions made concerning the independence of the samples can not be assumed in a network of queues. The second problem is that the density function is not known a priori.

Importance sampling using likelihood ratio

To apply the concept of IS the density function or at least the ratio f/f^* should be known a priori. This information is not available for simulation. Several articles [Par89], [Fra91], [Fra91-2] and [Fre89] suggest that this problem can be overcome using the large deviation theory. Instead of the ratio of the density functions, likelihood ratios are used. A likelihood ratio gives the likelihood of occurrence of a sample in the undisturbed situation. In [Wal88] at chapter 10.4 an introduction to this theory is given.

In this chapter they use the likelihood ratio in combination with IS to estimate the time until the population in a network reaches a given large value. The likelihood ratio is determined analytically in the articles. To determine this likelihood, assumptions are made.

This is exactly the reason why IS with likelihood ratio can not be applied in this project. The necessity of making these assumptions is the reason for simulation in this project. Hence the method as proposed in [Wal88], [Fra91,92] and [Fre88] can not be applied.

4.6.2 Restart

The concept is simple. Given the rare event A of which the probability must be estimated. A set of events C is defined such that.

$$C \supset A \wedge 1 > P\{C\} > P\{A\}$$

According to the rule of Bayes:

$$P\{A\} = P\{A|C\}P\{C\}$$

Since P{A|C} is estimated from a small portion of the simulation and P{C} from the whole simulation. The probability P{C} will normally be estimated with better confidence than P{A|C}.

By repetitive simulating A|C, P{A|C} and therefore P{A} will be estimated with much better confidence.

Restart can be used to estimate the probability of event A. Where event A can be defined as the system state(S) exceeds some level L. Examples of possible parameters S are[Vil91]: Length of a queue, the maximum or any other function of several queues, or the waiting time of the oldest call. The principle of Restart is to spend more effort in simulating the system when the system exceeds some threshold T (T<L).

The small probabilities in the performance measures can be used can be estimated using Restart. The Restart method does not assume independency of samples. And it does not need complex analysis, in which assumptions have to be made.

4.7 Overview of the methods

All techniques, except Restart, discussed sofar can not be applied directly. Emulators are above price. Hybrid techniques do not prevent the need for accelerated simulation in this project. Extrapolating using extreme value theory as applied in [Dij91] assumes independent observations from the delay distribution, and requires analyses to derive the performance measures from the extrapolated delay distribution.

Importance sampling needs a priori information of the density function or at least about the ratio of the density function of two situations. The use of likelihood in combination with importance sampling requires complex analyses or assumptions which is the basic reason for simulation in this project.

The method Restart can be used to estimate low probabilities of performance measures. An important advantage of Restart above Importance Sampling and Extreme value theory is that no analysis are involved by estimating the required probability. The assumptions made sofar are not made with Restart. This method has been applied to an performance study in an ATM environment for a statistical multiplexer. But as far as the literature has been investigated not on a virtual connection.

After a thorough study on the working and assumptions, this method is considered as the most promising one for this project. The detailed working and the implementation of this method is discussed in the next chapter.

References

[Cas88]: Castillo, E. "Extreme value theory in engineering", Bookpubl. by Academic press inc., ISBN 0-12-163475-2, CXN88CAS(Besliskunde).

[Cos224]: Roberts, J.W.(editor), Performance evaluation and design of multiservice networks", Cost224 final report, October 1991, Chapter 10, p 197-214.

[Cot83]: Cottrel, M. et al, "Large deviations and rare events in the study of stochastic algorithms", *IEEE Transactions on automatic control*, Vol. AC-28, No. 9, September 1983, p 907-920.

[Dij91]: Dijk, V. et al, "Extrapolating ATM simulation results using extreme value theory", proc ITC-13, 1991, p 97-104.

[Fra91]: Frater, M. et al, "Fast estimation of the statistics of rare events in jackson networks", proc. ITC-13, 1991, p 695-700.

[Fra91-2] Frater, M. et al, "Optimally effecient estimation of the statistics of rare events in queueing networks", *IEEE Transactions on automatic control*, Vol. 36, No. 12, December 1991, p 1395-1405.

[Fre89]: Fresnedo, R.D. "Quick simulation of rare events in networks", Proc. of the 1989 winter simulation conference, 1989, p 514-23.

[Fro88]: Frost, V. "Efficient techniques for the simulation of computer communications networks", *IEEE Journal on selected areas on communications*, Vol. 6, No. 1, January 1988, p 146-156.

[Fro91]: Frost, V. et al, "A technique for extrapolating the end to end performance of HDLC links for a range of lost packet rates", *IEEE transactions on communications*, Vol. 38, No. 4, April 1990, p 461-6.

[Jai91]: Jain R.,"The art of computer system performance analysis", John Wesley(1991).

[Kle74]: Kleijnen, Jack P.C. "Statistical techniques in simulation part I&II", Book Publ. by Marcel Dekker, Inc. New York 1974.

[Kur88]: Kurose, J. et al, "Computer aided modeling, analysis, and design of communication network", *IEEE Journal on selected areas in communications* Vol. 6, No. 1, January 1988, p 130-45.

[Vil91]: Villen-Altamirano, M. et al, "Restart a method for accelerating rare event simulations", proc. ITC-13, 1991, p 71-77.

[Par89]: Parekh, S. et al, "A quick simulation method for excessive backlogs in networks of queues", *IEEE Transactions on automatic control*, vol. 34, No. 1, January 1989, p 54-66.

[Wal88]: Walrand J., "Large deviation", book "An introduction to queueing networks", Prentice Hall, Englewood Cliffs, New Jersev 07632, 1988, Ch. 10.4, p 332-45.

[Wes91] Wessel, a.e. et al, "Importance sampling via a simulacrum", *Journal of the Franklin Institute*, Vol. 327, Iss. 5, 1990, p 771-83.

[Yok91]: Yokoi, T. et al, "ATM network performance evaluation using parallel and distributed simulation technique", proc. ITC-13, 1991, p 815-19.

Chapter 5
Restart and the implementation of the simulation software

5.1 Introduction

Restart stands for REpetitive Simulation Trials After Reaching Threshold. This method for accelerating simulations is used to estimate the probability of rare events. Restart is introduced by Manual and José Villén Altamirano in 1991 at ITC-13(International teletraffic conference) [Vil91].

In paragraph 4.6.2 the principle has been described. Paragraph 5.2 discusses some aspects of Restart. In [Vil91b] an improvement of the method is described. This improvement is achieved with an hysteresis effect. The working of this hysteresis effect is also explained paragraph 5.2. Two articles are written about Restart [Vil91] and [Vil91b]. These two articles are the basis of the simulation method. The notation used is reviewed in appendix A.

Before implementing this method a choice of the language and programming environment has to be made. The possibilities and the motivation for the choice is discussed in paragraph 5.3.

The simulation software to simulate a virtual connection is developed in three stages. The three stages are explained in paragraph 5.4. Also an overview of the implemented software is given in this paragraph. The three successive paragraphs treat the implementation of these stages. The last paragraph reviews the topics of this chapter. The structure and the description of the implemented software involves a number of implementation aspects. The details of the implementation can be found in appendix B and C.

5.2 Restart

The concept is explained in 4.6.2. Paragraph 5.2.1 treats the working of Restart. To evaluate the variance of the estimated probability assumptions have to be made. These assumptions and the evaluation of the variance is explained in 5.2.2. Paragraph 5.2.3. treats the improvement of the method by introducing an hysteresis effect. The definition of the gain and the formula of it is reviewed in paragraph 5.2.4.

5.2.1 Working of Restart

The method is not restricted to discrete time discrete state models. But for clarity reasons Restart is explained assuming a discrete state discrete time model. The time axis is divided into slots of equal lengths, named timeslots.

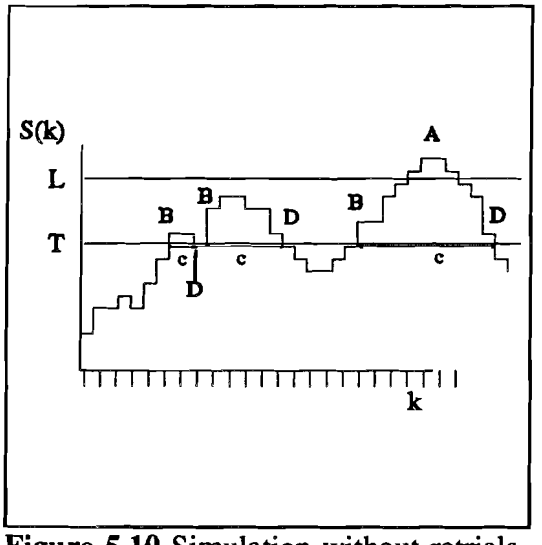


Figure 5.10 Simulation without retrials.

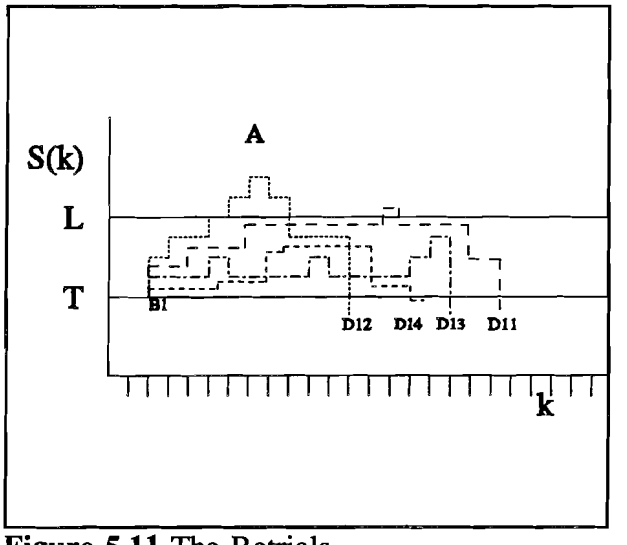


Figure 5.11 The Retrials.

The working of Restart is best explained with figures 5.10 and 5.11. In these figures the system state against the time is depicted. Examples of possible parameters S are[Vil91]: Length of a queue, the maximum or any other function of several queues, or the waiting time of the oldest call.

In these graphs the following variables are used:

S(k):	System state.
T:	System state when the system enters region C.
L:	System state where event A occurs.
B:	The event C occurred in the timeslot having occurred NOT(C) in the previous one.
D:	The event NOT(C) occurred in the timeslot, having C occurred in the previous one.

An overview of the notation used can also be found in appendix A.

Restart is used to estimate the probability of an event A. Event A is where the system state exceeds a level L. The concept of Restart is to spend more effort in simulating the system when it is above some threshold T. The entering of a region where S(k)>T is called an event B.

When event B occurs the system state is saved. Region [B,D) is repeated a fixed number of times. The repeated simulations of interval [B,D) are called retrials. R is the number of retrials for each event B. The starting system state of each retrial is the same state belonging to event B for all R retrials. The ending event D may be different for each retrial. But the retrial will end after the timeslot in which the system state becomes below T.

After each retrial the system state belonging to event B is restored. Except for the R-th retrial. After the R-th retrial the system continues in the normal way until the next event B.

Before discussing the way the probability is estimated, the notation used is reviewed below:

P: P{A}
P_1: P{C}
P_2: P{A|C}
P'_1: P{B}
P'_2: expected value of the number of timeslots in interval [B,D) where event A occurs;

According to the rule of Bayes, the probability of event A: P can be calculated by:

$P= P_1*P_2$

This can be rewritten to

$P=P_1'*P_2'$

Here P_1' is the probability of event B and P_2' the expectation of the number of events A conditioned on event B.

These P_1' and P_2' are estimated from the simulation. The algorithm which presents the working of Restart is presented by a Nassi Schneiderman diagram in figure 5.12.

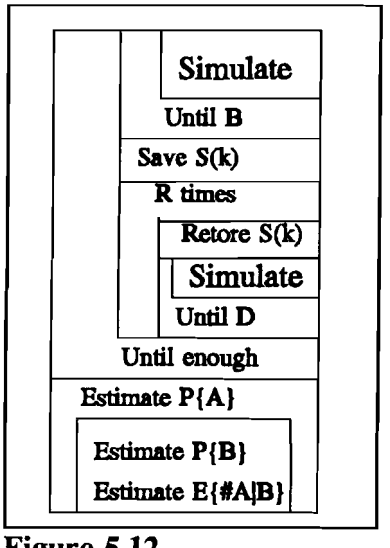


Figure 5.12

5.2.2 The assumptions and variance evaluation of Restart

Before discussing the assumptions some definitions will be reviewed. These definitions coincide with definitions given in [Vil91], and are also stated in appendix A.

- P: P{A}
- P_1: P{C}
- P_2: P{A|C}
- P'_1: P{B}
- P'_2: expected value of the number of timeslots in interval [B,D) where event A occurs;
- a: expected value of the number of events C in interval [B,D);
- N: Number of timeslots simulated without counting those of the retrials;
- N_1: Number of events B which occur in the simulation without counting the retrials.
- N_2: Number of events which occur in the whole simulation.
- Y_i: A random variable which indicates the number of events A occurred in the i-th interval [B,D)

The following relations between the variables are given:

$P = P_1P_2 = P'_1P'_2$

$P_1 = a\ P'_1$

$P_2 = P'_2/a$

The variance of $\hat{p}$ is used to express the confidence of the estimated probability.

With the analyses of the variance, the following assumptions are made in [Vil91]:

1. P'_1 and P'_2 are independent random variables i.e. The expectation of the number of events A conditioned on event B, is independent of the probability of B

2. The variables Y_i are considered mutually independent. So the number of events A is independent of the interval [B,D).

The exact analysis can be found in [Vil91], here only the results are stated.
In a simulation without Restart the following estimators are used:

$$\hat{P}'_1=\frac{N_1}{N}\quad \hat{P}'_2=\frac{N_2}{N_1}\quad \hat{P}=\frac{N_2}{N}$$

$$\textit{thus}\ \hat{P}=\hat{P}'_1\times\hat{P}'_2$$

In [Vil91] the following formula for the variance is derived:

$$V(\hat{P})=\frac{P}{N}(K_1P_2\frac{K_2}{NP}+K_1P_2+K_2)\approx\frac{P}{N}K_2$$

Here in is:

K_1: K_1 can be defined as the average the number of events C which occur in or around a given event B, minus the number of them which would occur if there were independence between near regions C.

K_2: K_2 can be defined as the average the number of events A which occur around a given event A minus the number of them which would occur if there were independence between near instants.

In a situation with Restart, the probability P{A} is estimated with the following estimators:

$$\hat{P}'_1=\frac{N_1}{N}\quad \hat{P}'_2=\frac{N_2}{RN_1}\quad \hat{P}=\frac{N_2}{RN}$$

$$thus\ \hat{P}=\hat{P}'_1\times\hat{P}'_2$$

With Restart the estimator of P_2' differs from direct simulation. Before giving the variance of P, the evaluation of var(P_2') is reviewed.

In [Vil91] two definitions are given:

- Y_{ij} is the random variable which indicates the number of events A during the j-th retrial made starting from B_i i.e. from the i-th event B.
- Y_i': the conditional expectation of Y_{ij} to B_i.

P_2' is sampled in two stages:

First a sampling of N_1 primary units is made corresponding to the events B occurring in the simulation. In the second stage R secondary units are sampled for each primary unit. Both the primary and the secondary units are assumed mutually independent. In [Vil91] it is stated that $V(Y_{ij}) = V_1+V_2$. Where V_1 and V_2, being respectively the variance between primary units and expectation of the variance between secondary units. within the same primary unit.

With these definitions a factor b is defined as follows:

$$b=\frac{V_1}{P_2\times V(Y_{ij})}$$

This factor represents the degree of influence of the variance $V_1=V(Y_1')$. In the case where the definition of event B is such that past and present system state variables which can influence have the same value for all events B_i, b will be zero.

With the evaluation of var(P_2') the following formula for the variance is derived:

$$V(\hat{P})=\frac{P}{N}[K_1P_2+(K_2+K_1P_2\frac{K_2}{2NP})\times(\frac{1}{R}+bP_2)]\approx\frac{P}{N}[K_1P_2+K_2(\frac{1}{R}+bP_2)]$$

The values of R and the definition of the events A, B, C and D are dependent of the model to be simulated. How these events are defined for a specific model will be explained in chapter 6.

5.2.3 Restart with Hysteresis

In [Vil91b] an improvement of the method is proposed by introducing an hysteresis effect. Restart repeats some regions a number of times when the system state exceeds some threshold. Therefore it is necessary to restore the system state at the beginning of each retrial. The restoration of a system may reduce the efficiency of the method. Hysteresis is introduced to cope with this problem.

Working

Instead of defining the region [B,D), to be repeatedly simulated, with one threshold T, Restart with hysteresis defines the region with two thresholds T and T' as depicted in figure 5.4.

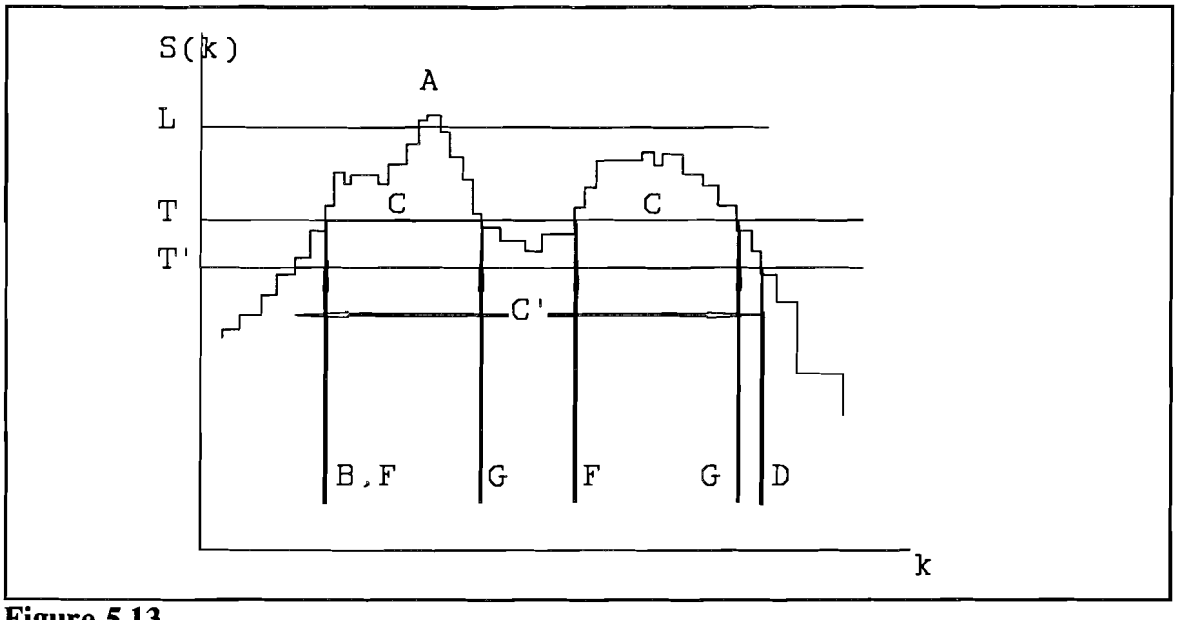


Figure 5.13

The working of the method remains the same except that the ending point of each retrial is at a lower level(T') than each starting point(T). In figure 5.4 the normal simulation is shown. In figure 5.5 the simulation with Restart is illustrated.

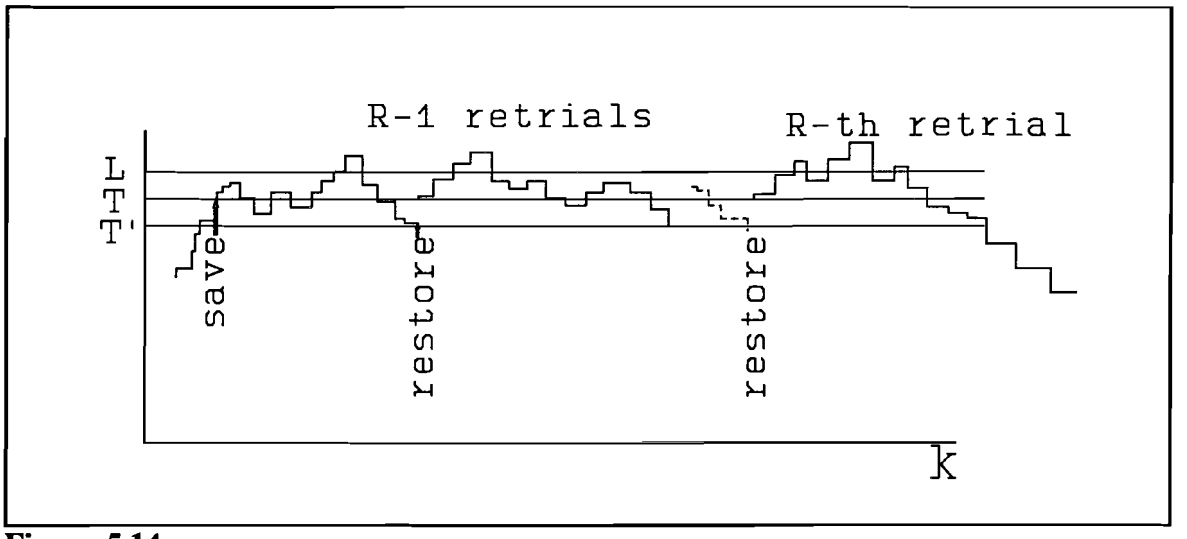


Figure 5.14

The analysis remains almost the same. The difference is that not all timeslots in interval [B,D) have to belong to region C. The parameter 'a', the number of events C in interval [B,D), used to be equal to the interval length [B,D). With hysteresis it can be less since the system state S may lower T for a while during the retrial.

Restart with hysteresis does effect the following factors of Restart:

- K1
- b
- Costs of the simulation.

Factor K1

The use of hysteresis does affect the value of K1. K1 can be defined as the average number of events C which occur in or around a given event B, minus the number of them which would occur if there were independence between near regions C. Hysteresis reduces the factor K1[Vil91b]. This reduction of K1 results in an extra variance reduction.

Factor b

This factor represents the degree of influence of the variance $V_1=V(Y_1')$. The factor b will also be reduced by the hysteresis effect. Which results in an extra variance reduction[Vil91b].

Costs of the simulation

The hysteresis effect was introduced to cope with the restoration costs. Hysteresis tries to reduce the effort spend on saving and restoring the system state by reducing the number of times that a system have to be restored. This reduction in costs can be expressed in the number of timeslots to be simulated. Let dN be the number of timeslots that are gained by using hysteresis. Using the notation as given in appendix A it can easily be shown that:

$$dN=(m-1)N_1*r*(R-1)-N_1*(R-1)*(l-a) \qquad (15)$$

The costs are reduced when dN>0. This can be rewritten to the following condition:

$$(m-1)r>(l-a)$$

In [Vil91b] a procedure to find an optimum for the threshold T' is proposed.
The procedure is based on the following. The costs of the simulation can be expressed by the formula:

$$Costs=N+Nl(l+r)(R-1)\approx N(1+\frac{Pl}{a}(l+r)(R-1)) \tag{17}$$

With a= mμ the factor (l+r)/m should be minimized.
When T'=T then l=a and m=1. When T' is decreased the number of intervals [F,G) will increase, as will l but m is the basis of the ratio and has more effect. So at first the ratio (l+r)/μ will decrease. When T' is decreased further the number of intervals [F,G) will stabilize while l shall increase. This result in a increase of the ratio. Between those situations an optimum can be found.

5.2.4 Gain of the method

The main point is the improvement of Restart. Thereby not only the variance reduction but also the costs should be taken into account. [Vil91] measures the costs as the number of instants to be simulated. The product of the costs with the variance reflects the performance of the simulation method. [Vil91] calculate the costs by:

$$C=N+N_1(l+r)(R-1)\approx N(1+P_1yR) \tag{18}$$

y here is the factor (l+r)/a.
The optimum values for P_1, R and y which minimize C*V(P) are derived in [Vil91b] to be:

$$y_o\approx\sqrt{1+\frac{r}{\mu}}\;;\; P_{1o}=\sqrt{(\frac{K_1}{K_2}+b)\frac{1}{y_o}P}\;;\; R_o=\frac{1}{\sqrt{(\frac{K_1}{K_2}+b)y_oP}} \tag{19}$$

With this optimal values the gain G is calculated as the ratio of the costs times variance of the simulation without and with Restart. In [Vil91b] the following formula for G is derived:

$$G=\frac{1}{4\sqrt{(\frac{K_1}{K_2}+b)y_oP}} \tag{20}$$

5.3 Choice of the simulation language

The choice of a language is an important one. There are three choices: Simulation language, General Purpose language or an extension of a simulation language.

5.3.1. The methods of simulation languages

The simulation languages are written especially for that purpose. The methods used can be different. Since a discrete state model has to be simulated, a simulation language which handles discrete state models has to be chosen. The simulated clock can be incremented with constant amounts of time or the clock can be set to the next event. Within simulation languages three approaches are used:

1. Event oriented.
2. Activity oriented.
3. Process oriented.

First let me review some definitions:

event:	change in the value of one or more system variables at a point in time.
activity:	State of an object over an interval.
interval:	A duration between two successive instants.
instant:	A value of the system time at which the value of at least one attribute of an object can be altered.
process:	The succession of states of an object over a span.
span:	Contiguous succession of one or more intervals.
state of an object:	The enumeration of all attribute values of that object at a particular instant.

ad 1 Event oriented.
In this approach the programmer defines events. When this event occurs, routines are invoked.
ad 2 Activity oriented.
In this approach the programmer defines activities which are started when some conditions are satisfied. Usually this type of simulation uses a clock which is incremented with constant time units.
ad 3 Process oriented.
The programmer defines the processes(entities transactions components) which use the resources. Each process can be in three states: active, scheduled or passive. A process is active when the process is currently being processed by the simulator. Scheduled means that a process is waiting for an interval or simulated time to pass. A process is in the passive state when it is waiting for an event to occur.
In figure 5.15 an example is depicted of a M/M/1 queue with two arrivals.

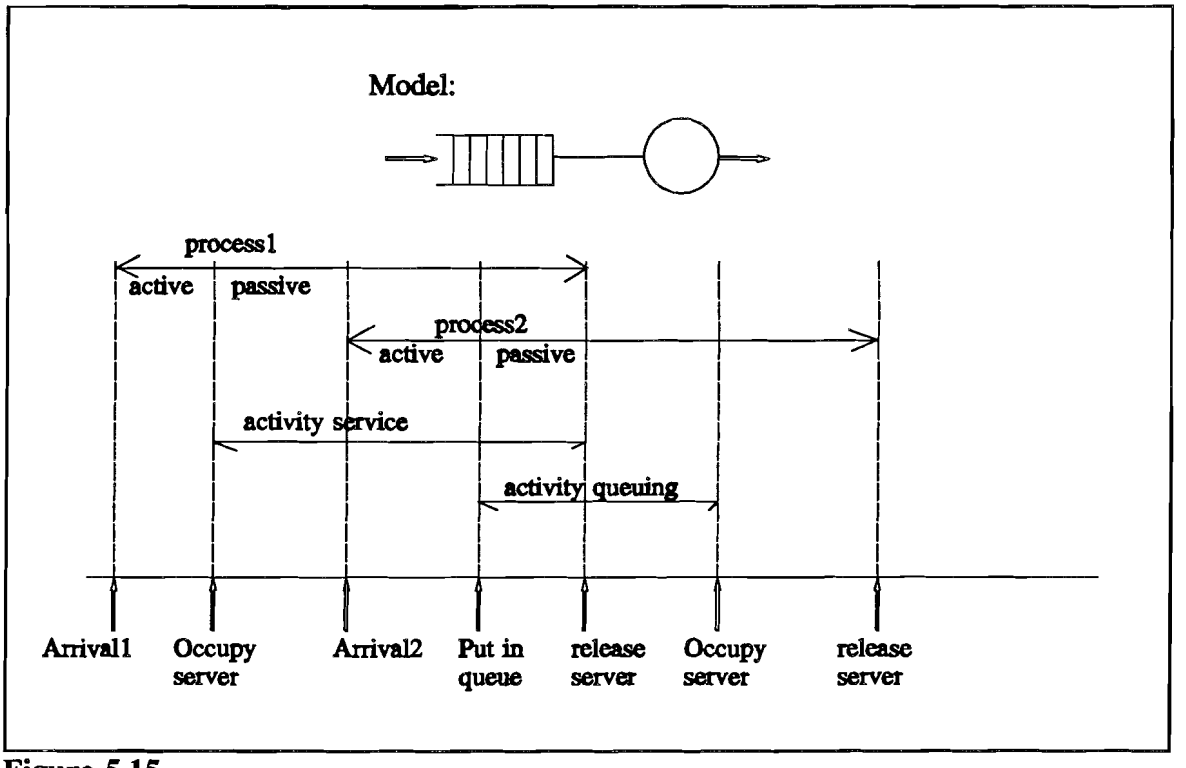


Figure 5.15

5.3.2. Possibilities of languages:

CSIM

Csim is an extension of the language C. Csim has a process oriented approach for discrete simulation. Csim is developed by Herb Schwetman of MCC(Micro electronics and computer technology corporation) in Austin(USA). Schwetman wrote two articles on Csim:

[Sch86]: Csim a C-based process oriented simulation language.

[Sch88]: Using Csim to model complex systems.

The advantage of Csim is that the C-programming environment is available to the programmer. A disadvantage is that there is no good mechanism to check if a Csim program is correct.

The program Csim runs on a UNIX operating system on several computer systems including DEC-VAX and the SUN III workstation. Csim is also referenced by [Sco91] by which it is used.

Simula

Simula is a process oriented general purpose language. It was used by the faculty of Industrial Engineering to build simulators. The faculty has turned to another simulation software named Must which is a library for Turbo Pascal. The reason for this is that the environment of Pascal is than available to the programmer.

GPSS

GPSS stands for General Purpose Simulation System. It is a discrete simulation language with a process oriented approach. It is possible to implement the model, on which Restart is applied in GPSS. The difficulty will be saving and restoring the system state at some moment. Therefore programming should be done in Fortran. A drawback of GPSS is that it is an interpreter rather than a compiler.

Must

Must is a Micro simulationtool. Must is an extension of Turbo Pascal. It provides tools for discrete simulation on a event oriented approach. Must is a trademark of a dutch company. This software package is available via the faculty of Business. It runs on a:
IBM PC or PS/2 system,
640kB
Expanded memory(optional)
Mouse(optional)
Turbo Pascal 5.0 or 6.0
Msdos ver 3.x, 4.x or 5.x(recommended).

Must contains a set of function and procedures to handle queues, statistical reports etc. There is no function available to save or restore a system state. But it should be possible to program one. Must uses an event oriented approach but a process oriented approach can easily be programmed with this tool. The disadvantage is that only compile units are available.

Simpack

Simpack is a library of simulation tools written in C. It is developed by Paul Fishwick at the University of Florida. Little documentation is available. One technical report giving general information about Simpack and an example. Furthermore a file with general information about the functions. There is no reference manual available. I contacted Mr Fishwick and he was writing a book but that is published in November 93. The approach used is event oriented.

Simpack++

Simpack++ is a library of simulation tools written in C++. The simpack is translated to C++ where each object is a class.

5.3.3 Choice of the language

Csim is not chosen because it was not available on a short notice. The lack of experience with the languages Simula and Fortran made the choice of Simula and GPSS unattractive. Simpack or Simpack++ has been considered as a possibility because it has the advantage of the availability of both UNIX and TURBO-C as programming environment. The lack of documentation of Simpack(++) and the availability of documentation with Must has led to choice of Turbo Pascal as programming environment with Must as simulation tool.

5.4 Simulation of a Virtual connection using Restart

When Restart is referred in the rest of this chapter, the hysteresis effect is assumed to be included. Restart is a variation reduction technique to estimate the probabilities of rare events. Most of the performance measures can be considered as rare events. Restart can be used to estimate cell loss probability and excessive delay.

Other performance measures involving probability density functions like delay distribution and the interval arrival time distribution can not be considered as an event.

To fulfil the object of measuring both probability of events and probability density functions, the following traject in developing the software has been followed. First a general simulation environment must be created in which straight simulation is possible to determine the distributions that can not be determined with Restart. This simulation environment will be named SIMMUST, which is a combination of the words simulation and Must.

SIMMUST is the basis of the second stage. The second stage involves the implementation of Restart leaving out details of the definition of the events. The implementation of Restart on a specific model will be the last stage at the traject of developing the simulation software.

The simulation software is with these three stages divided in three versions. A scheme of these versions is depicted in figure 5.16.

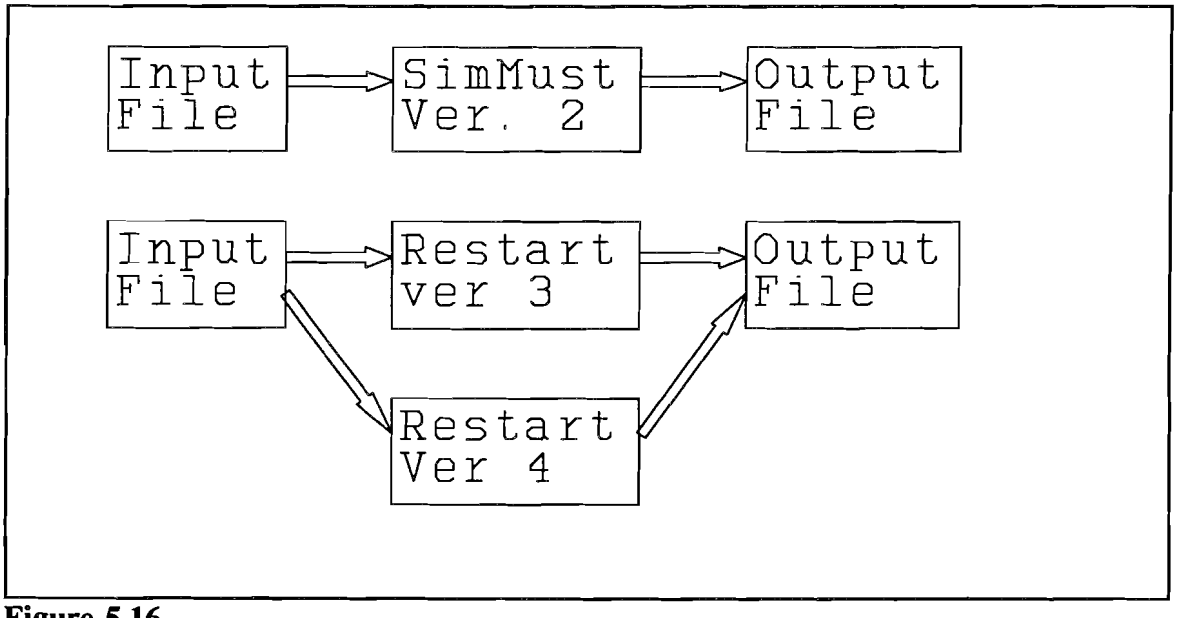


Figure 5.16

5.4.1 SIMMUST

SIMMUST means SIMulation with MUST. It is a simulation environment to simulate models of Virtual Connections. The environment is programmed in Turbo Pascal using the simulation tool Must with an object oriented approach.

Simulation is an technique wherefore an object oriented approach is very applicable and usable. The use of object oriented programming in simulation has the following advantages above classical program techniques:

- Simulation models can be assembled from objects. This allows a diversity of models.
- The simulation model can easily be extended with other objects.
- The implementation of Restart independent from the simulation model is less complex.

Turbo Pascal 6.0 supports an object oriented approach. Must is not designed to be used in an object oriented programming(OOP) environment but it can easily be used in one. For an Introduction on OOP I refer to [Smi91] and the Turbo Pascal reference Manual.

Objects

To implement the simulation software with OOP, objects have to be identified first. With the identification of the objects a distinction can be made between the objects of the model and the performance measures.

Model

A model of a VC consists of the following components:

- Source
- Node
- Cell

There are several types of sources that can be simulated. The following sources are implemented:

Deterministic source
Batch source
TwostateMMBP
IBP
ON OFF source.

All these sources can be used as sources of the virtual connection or as sources simulating background traffic. The sources are explained in paragraph 3.5.

There is only one type of node implemented. This type is named 'Server' and represents a FIFO queue with a deterministic server. The number of nodes used in the simulation model is variable, but a maximum of eight is assumed.

Three types of cells are implemented:

Cell	Represents a cell in the simulation model. This cell is used for background traffic.
CellVC	Cell of the Virtual Connection, this component represent a cell belonging to the VC traffic.
CellBB	Cell Batch Burst, this component represents also a cell of the VC traffic. But with the additional characteristic that it represents a cell in a batch or a burst of cells.

The CellBB is a type of cell generated by a source which generates bursts or batches with cells. This distinction is made to keep track of relative delays within a batch or burst. The importance of this in performance measures is explained in paragraph 3.2.2.

For every type of component an object and a process is defined. An object consists of a data field and methods. The datafield describes the attributes of the component. The methods are functions and procedures implemented in pascal which operates on the attributes. The process describes the behaviour of the component in the simulation model.

For a source this means that the process describes when cells are generated and how many. In fact the process of a source is the implementation of the stochastic process by which the source is modeled. The parameters needed are provided by the attributes and the methods.

The process of a node describes the queuing strategy and the time that the cell is serviced. The process here is the implementation of the queuing model. The process of a cell represents the behaviour of the cell in the simulation model. In fact this is mainly the routing of the cell from node to node.

Every component has three states which coincide with the states described in 5.3.1. ad3.

Current(active)
Passive
Scheduled

When a process is activated it becomes current and the process of the component is invoked. If a component is passive, it waits until another process activates this component. The process of the reactivated component continues where it was stopped. When a component is scheduled it becomes current at a known time in the future.

Example

Let us review the terms and definitions used with a example of a source of the type: Deterministic source. The object of this source consists of methods and attributes. The attributes are for example:

Deterministic interval
Number of cells to be generated.

The methods are for example:

get_interval_time;	Which returns the deterministic interval between generated cells.
get_nr_cells;	Returns the number of cells to be generated
decrement;	Decrements the number of cells to be generated with one.

The process describing the behaviour could be one illustrated in the diagram depicted in figure 5.17. This process is for a VC source.

The source generates cells as long as necessary. When a cell is generated it schedules

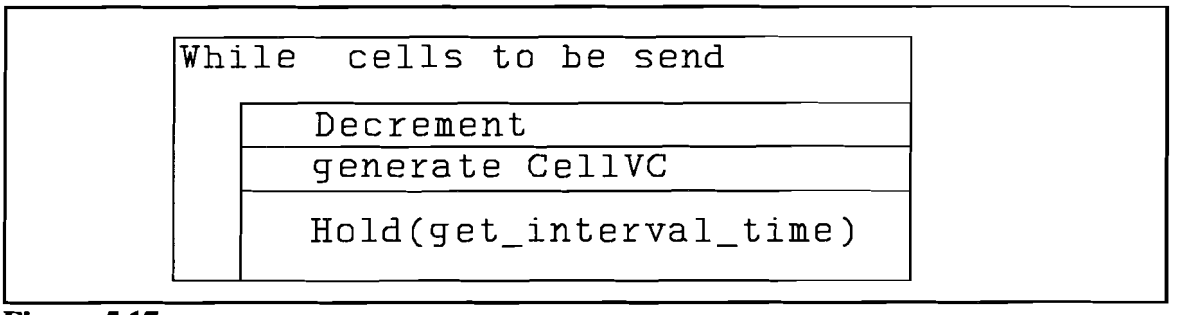


Figure 5.17

itself after a deterministic interval.

For each component an object and a process is implemented in pascal using tools from Must. The implementation is gathered in a Turbo Pascal Unit. The description and some implementation aspects can be found in appendix B.

Performance Measures

The performance measures are the results of the simulations. They are gathered in one object named Perfobject. The attributes of the object are the performance measures and the additional variables. The methods are functions and procedures which calculate and keep track of these measurements during and afterwards the simulation.

All declarations involving this object are gathered in a TurboPascal unit named Perfobject. A description of this unit can be found in appendix B.

MainProcess

The mainprocess is invoked as SIMMUST is executed. Like any other process this process has also 3 states: current, passive and scheduled. The first task of the mainprocess is to initialize the simulation model. This is done with an initialization procedure, which is described in appendix B. The input needed herefore is read from an input file. The input file is an ASCII file in which the parameters of the model are specified. What parameters and the format of this input file is specified in appendix B.

After the model is initialized. The simulation is started by activating the servers and the sources. All results from the simulation are gathered by the performance object.

The simulation can be stopped normally in two ways:

1. When the source of the virtual connection has simulated all cells. Than the simulation run is finished.
2. By the user activating the Trap procedure.

The trap procedure is activated by pressing the space bar. This trap facility is implemented in Pascal and has the ability of stopping the simulation.

The program SIMMUST has three output files:

Results
HistFile
ArrFile

Results contains the performance measures concerning cell losses. HistFile contains the histograms from the simulation. The Arrfile is optional and contains information about the relative delays when batches or bursts are involved. The specification of the outputs can be found in appendix B.

5.4.2 Restart

Restart is implemented with SIMMUST as basis. One requirement of the implementation of Restart was that the algorithm should be independent of the VC model used. Restart concentrates on estimating the probability of an event A. Other performance measures like the distribution of the delay and inter arrival times are left out of this implementation. This means that the simulation environment can be simplified by leaving out all methods involving histograms used to present the distributions.

Analog to SIMMUST the input parameters are read from an input file. Since no parameters are needed for the histograms, the input file is slightly different from the one for SIMMUST. Extra information needed is the number of retrials. The exact format of the input file can be found in appendix C.

Besides the probability of event A, the variance of this probability is needed to express the accuracy of the estimation. How this accuracy is calculated from the variance will be explained in chapter 6 with the analysis of the simulation results.

This variance must be estimated from the simulation. The formula for the variance of the estimated probability is:[Vil91]

$$\hat{V}(\hat{P}) = Vp1 * Vp2 + Vp1 * (\frac{N_2}{RN_1})^2 + Vp2 * (\frac{N_1}{N})^2 \tag{21}$$

Vp1 is estimated by a regenerative method. This can be applied when the system is regenerative. A system is said to be regenerative if there exists a particular system state called a regenerative state such that whenever the system returns to that state the history of past states of the system have no influence on the future of the system. In that case the intervals between the regeneration points are independent.

Therefore regeneration points have to be defined. Event V is defined to be come active after the last time slot of a regenerative interval. A regenerative interval is said to end when the system returns in the regenerative state. During the simulation the system is scanned for events V. When event V occurs the length of the interval, not counting the retrials, and the number of events B in that interval is saved.

The cycle in which the simulation stops is completed. This results in a better confidence interval[Rip88], since simulation is more likely to stop in a cycle in which events B take place.

Calculation of Vp1

The difference between the expected and the observed cycle sums is[Jai91]:

$$w_i = y_i - n_i P_1' \tag{22}$$

y_i: The number of events B in cycle i
n_i: The length of cycle i
Vp1 can be calculated by:

$$vp1 = \hat{V}(\hat{P}_1') = \frac{1}{m(m-1)(\bar{n})^2} \sum_{i=1}^{m} w_i^2 = \frac{1}{N^2} \frac{m}{m-1} \sum_{i=1}^{m} w_i^2 \tag{23}$$

with m is the number of regeneration intervals.

Calculation of Vp2

Vp2 is calculated with the formula:

$$V(\hat{P}_2')=E(\frac{1}{N_1})(V_1+\frac{V_2}{R})=\frac{V_1}{P_1'N}+\frac{V_2}{P_1'NR}$$

V1 and V2 are estimated using the independent replication method [Jai91]. With this method the overall mean is the same as the mean of the replications.

Estimation of V1

The definition of V1 is[Vil91]:

$$V_1=E[\overline{Y}_i-P_2']^2 \tag{25}$$

$\overline{Y}_i$: *Conditional expectation of the number of events A conditioned on System state* B_i

V1 is estimated from N1 independent replications by:

$$\hat{V}_1=\frac{1}{N_1-1}\sum_{i=1}^{N_1}(\overline{Y}_i-\hat{P}_2')^2=\frac{1}{N_1-1}\sum_{i=1}^{N_1}(\overline{Y_i^2})-\frac{N_1}{N_1-1}\hat{P}_2' \tag{27}$$

Estimation of V2

By the definition in [Vil91] $V_2= E[V_{2i}]$. V_{2i} is defined by:

$$V_{2i}=E(Y_{ij}-\overline{Y}_i)^2 \tag{28}$$

V_{2i} is estimated from R independent replications by:

$$\hat{V}_{2i}=\frac{1}{R-1}\sum_{j=1}^{R}(y_{ij}-\overline{y}_i)^2=\frac{1}{R-1}\sum_{j=1}^{R}(y_{ij})^2-\frac{R}{R-1}\overline{y}_i^2 \tag{29}$$

V2 is estimated by:

$$\hat{V}_2=\frac{1}{N_1}\sum_{i=1}^{N_1}\hat{V}_{2i} \tag{30}$$

The output of Restart is different from the output with SIMMUST. Besides the probability of event A and the variance, parameters needed for the analysis of the results are written to the output file 'results'. The exact output specification can be found in appendix C.

The algorithm is implemented as a process. A pascal alike representation of the algorithm is listed in appendix C. The restart process checks the simulation model if an event B is occurred. This checking is done at the beginning of each timeslot. When an event B occurs, R retrials are made. After which the process checks for the next event B. The definition of the events is not implemented in this version since these are dependent of the VC model used.

5.5 Restart on a specific VC model

The implementation of Restart on a model consists of the definition of the following events: B, D, F, G, V and A. The definition of the events B, D, F, G and V are defined in pascal functions which are invoked each time the algorithm wants to check the simulation model for such event. Event A is slightly different. The moment event A occurs a boolean variable in the perfobj is set. This boolean variable is checked only during the retrials at the beginning of each timeslot. If this boolean variable is set, it does the necessary operations and the boolean is reset. In this way no events A are left undetected. Though it is necessary to define event B such that no events A can occur outside this interval.

The problem is not implementing the events but to define the events. With events B,F,G and D the problem is to define the thresholds T and T', on which these events are based. In 5.2.3 the optimum values P1 and y are given. But this optimum values require knowledge about K1, K2, b and μ. If this optimum is known than the problem is to define the thresholds such that this results in the optimum of P1.

Another problem is to find the generation points. Mostly this is when the system is empty. This is trivial for the nodes but not for the sources.

In the next chapter some simulations on a specific model are done to find out what Gain can be achieved and in what way the choice of the events B and D influences the gain.

5.6 Topics

Restart is a method which estimates the probability of rare events. It uses repetitive simulation trials when the system exceeds some threshold. The hysteresis effect described in [Vil91b] is an improvement of the method. Besides reducing the effort spend on restoring the system state, it improves the variance reduction by reducing the factors K1 and b.

To implement Restart first a simulation environment SIMMUST is developed. This simulation environment is developed using an object oriented approach. SIMMUST is developed in Turbo Pascal using simulation tools from Must. With SIMMUST it is possible to simulate a diversity of VC models.

SIMMUST is the basis of the implementation of Restart with hysteresis. Restart concentrates on the estimation of the probability of an rare event. First Restart is implemented independently of the model to be used.

With this software Restart can be implemented on a specific model by defining the events and implementing them in functions. The problem here is not the implementation but more the definition.

In the next chapter some simulation results are discussed. Plus the results of simulations of restart on a specific model are presented to find out the gain that can be achieved, and how the definition of events B and D affects the gain.

References:

[Sch86]: Schwetman H., "CSIM a C-based process oriented simulation language", *Proceedings of the 1986 winter simulation conference*, 1986, p 387-96.

[Sch88]: Schwetman H., "Using CSIM to model complex systems", *Proceedings of the 1988 winter simulation conference*, 1988, p. 246-53.

[Sco91]: Scoggins S., "A high performance modular design paradigm for teletraffic simulation with CSIM", *Proceedings of the 1991 summer computer simulation conference*, 23rd conference, 22-24 july 1991 Baltimore, p 363.

[Must]: Must reference manual.

[Rip88]: Ripley B.D.,"Simulation methodology- an introduction for queueing theorists", Queueing Systems, 3(1988), p 201-20.

[Smi91]: Smith, David N., "Concepts of object oriented programming", Book published by MacGraw-Hill 1991.

[Vil91]: Villen-Altamirano, M. et al, "Restart a method for accelerating rare event simulations", proc. ITC-13, 1991, p 71-77.

[Vil91b]: Villen-Altamirano, M. et al, "Accelerated simulation of rare events using restart method with hysteresis", *Telecommunication Services for developing economics*, published by Elseviers Science Publishers, 1991, p 675-86.

Chapter 6
Analysis of simulation results obtained with Restart

6.1 Introduction

This chapter gives the results of the application of Restart on a specific model. This application should illustrate that Restart is usable on a relevant model for a virtual connection and how the gain can be improved. In paragraph 6.2 the parameters that influence the gain will be reviewed. The accuracy of the estimated probability is indicated with a confidence interval. The meaning and the method by which the confidence interval can be calculated is explained in paragraph 6.3. Paragraph 6.4 explains the model on which Restart is applied. By simulating the model with several parameters the influence of those parameters on the gain is shown in paragraph 6.5. The results are summarized in paragraph 6.6. Detailed information of the model simulated and the results of the simulation runs are enclosed in appendix D.

6.2 The gain of using Restart

In 5.2.3 the gain of the method has been discussed. The performance of the simulation has been measured as the product of the variance and the number of timeslots to be simulated. The gain is defined as the ratio of this product without and with Restart. In [Vil91b] the gain is derived for optimal parameters. This gain is discussed in paragraph 6.2.1. It is not always possible to choose optimal parameters. The gain achieved when the parameters are not optimal is discussed in paragraph 6.2.2. The meaning of the formulas of the gain is reviewed in 6.2.3.

6.2.1 The Gain with optimal parameters

The optimal parameters are reviewed in 5.2.4. In that paragraph the following formula for the gain has been reviewed:

$$G_1 = \frac{1}{4\sqrt{(\frac{K_1}{K_2} + b) y_0 P}} \tag{31}$$

Not taken into account are the costs for saving the system state and checking for the occurrences of several events.

The factor y_0 is difficult to evaluate. The parameters needed, are not provided by the output of the simulation runs. In the analysis of the gain,presented here, y_0 is left out. Instead a factor is used representing the reduction of the number of timeslots simulated per second.

This reduction lead to a smaller gain caused by extra costs made special for Restart and can be taken into account by measuring the number of timeslots simulated in one second with and without Restart. This factor does not take into account the extra timeslots in the retrials, introduced by hysteresis, that do not contribute to the result.

Let t_r and t_w represent the following values.

- t_r: The average number of timeslots simulated per second when Restart is applied.
- t_w: The average number of timeslots simulated per second when Restart is not applied.

Both values depend on the model to be simulated and the hardware used. If both are measured with the same hardware and the same model the ratio t_r/t_w gives the reduction of the gain caused by the extra costs made specially for Restart. Not taken into account are the timeslots in the retrials that do not contribute to the results. These timeslots occur when the system is between T and T'. This ratio is independent of the hardware used.

The formula of the gain becomes then:

$$G_2 = \frac{t_r}{4t_w\sqrt{(\frac{K_1}{K_2}+b)P}} \quad (32)$$

In this formula the gain is dependent on five parameters:

K1
K2
P
b
t_r/t_w

The parameter K1

This parameter is dependent on the model used and the choice of the hysteresis levels T and T'. Restart with hysteresis decreases the factor K1 which results in an improvement of the variance. The optimum of T' has been discussed in 5.2.2.

The parameter K2

K2 depends only on the model of the VC. The effect on K2 by Restart with hysteresis is negligible[Vil91b]. So little can be done with this factor to improve the gain of Restart.

The parameter P

This parameter, the probability to be estimated, depends naturally only on the model. But in formula 32 can be seen that more gain can be achieved on models with smaller probability P. In table 6.1 the maximum gain assuming optimal parameters and no extra costs is calculated with formula 32. Assumed also is that K1/K2+b=1.

Table 6.1

P	Gain
10E-9	7900
10E-8	2500
10E-7	790
10E-6	250

This table illustrates that the gain increases as the probability to be estimated decreases. The characteristic that the gain increases with smaller probabilities is very useful in the performance study of virtual connection in ATM networks, since low probabilities have to be estimated.

The parameter b

At the occurrence of each event B, a system state is saved. With the saved system state a set of retrials is simulated. This set is named a primary unit. The system state need not to be the same for all events B. Due to the different system states belonging to the occurrences of the events B, the expectation of the number of events A conditioned on the system state, might be different for several primary units.

The variance between the primary units(V1) lead to a reduction of the gain. The parameter b is defined as:

$$b=\frac{V_1}{P_2(V_1+V_2)} \tag{33}$$

In formula 33 can be seen that a small variance between the primary units can lead to significant values of b, due to the fact that P2 becomes small when P is small.
In table 6.2 the reduction in the gain is illustrated by increasing b for P=10E-7, $t_w=t_r$ and K1=K2. The gain is calculated with formula 32.

table 6.2

b	Gain
0.1	753
1	559
10	238
100	78
1000	25

In this table can be seen that the gain is reduced dramatically if b increases. The factor b can be decreased by a sophisticated definition of event B, as will be seen later on.

The ratio t_r/t_w

This ratio is dependent of the model used. If a model is used of which a lot of components must be saved and restored, than the ratio will decrease as will the gain. Furthermore the choice of the number of retrials and the probability of event B influences the ratio. But these values must be chosen as close as possible to the optimum values as given in 5.2.3.

The choice of T' influences the restoration costs. Choosing an optimum for T' is discussed in 5.2.2.

The evaluation costs of functions checking the system states, decrease the ratio. But a complex evaluation of the system state checking for the events could lead to more gain. This lead to a choice between evaluation costs of the functions and the improvement of the gain by complex definition of the events.

6.2.2 The Gain with non-optimal parameters

The gain discussed in the previous sub-paragraph is derived assuming optimal values for P1 and R. These optimal values are derived in [Vil91b]. If the parameters are not equal to these optimal values, the gain can still be calculated by the ratio of the product of costs times variance of simulating without and with Restart. The variance of the simulation without Restart can assumed to be[Vill91]:

$$V(\hat{P}) = \frac{P}{N}\left(K_1 P_2 \frac{K_2}{NP} + K_1 P_2 + K_2\right) \approx \frac{P}{N} K_2 \tag{34}$$

Let us define:

N_r: Total number of timeslots simulated with Restart

G_3: Gain in the non optimal case

For G_3 the following formula can be derived:

$$G_3 = \frac{\frac{1}{t_w} P K_2}{\frac{N_r}{t_r} V(P)} = \frac{N}{N_r P_2 \left(\frac{K_1}{K_2} + b + \frac{1}{RP_2}\right)} \frac{t_r}{t_w} \tag{35}$$

Suppose the parameters P2 and R are as close as possible to the optimal case. Then the gain can be improved by decreasing the factor $\frac{K_1}{K_2} + b$ or increasing t_r. Note that a change in $\frac{K_1}{K_2} + b$ will lead to a change of the optimal parameter values.

6.2.3 The three formulas of the gain

In this paragraph three formulas for the gain have been discussed. The meaning of those gains are reviewed below:

G_1 This gain is the gain assuming optimal parameters are chosen. The gain is calculated by formula 31. With the calculations of G_1 in the simulation runs the factor y_0 is assumed to be one. The gain is calculated assuming no extra costs for Restart. G_1 is an upper bound of the gain assuming ideal circumstances.

G_2: Calculated with formula 32. G_2 is the optimal gain taking into account extra costs made by Restart. Not taken into account are the extra timeslots in the retrials, introduced by hysteresis, that do not contribute to the results.

G_3: Calculated with formula 35. This gain is the gain achieved with that particular simulation, not making the assumptions of optimal parameters.

6.3 Calculation of the confidence interval

The confidence interval indicates the accuracy of the estimated probability. The confidence interval can be calculated with several methods. Two methods are discussed here.

1. Chebysev inequality
2. Assuming normal distribution

ad 1.Chebysev inequality:

$$Prob(|\hat{P}-P|>\epsilon) \leq \frac{V(\hat{P})}{\epsilon^2} \tag{38}$$

Suppose an 95% confidence interval need to be calculated. Then the value of epsilon becomes[Jai91]:

$$\epsilon = \sqrt{\frac{1}{0.05}\sqrt{V(\hat{P})}} \tag{39}$$

This method results in an upper limit of the confidence interval.

ad 2. Assuming normal distribution.

When assuming that P is normal distributed with mean $\hat{P}$ and variance $V(\hat{P})$ than the confidence interval can be calculated by[Jai91]:

$$(\hat{P}-acc,\hat{P}+acc) \quad acc=z_{1-\frac{\alpha}{2}}\sqrt{V(\hat{P})} \tag{42}$$

In the simulation runs the 95% confidence interval is calculated assuming normal distribution. The z-value is then 1.96. This z-value can be compared to $\sqrt{\frac{1}{0.05}}\approx 4.4$ when Chebysev is used. This means that the confidence interval calculated with chebysev is twice as big as the confidence interval calculated assuming a normal distribution. One reason therefore is that Chebysev calculates an upperbound. The second reason is the assumption of a normal distribution.

6.4 Model to which Restart is applied

In the past paragraph the parameters that influence the gain have been reviewed. To illustrate the working of Restart a relevant model of a VC is simulated.

Restart is applied on a VC model with two nodes. The event A is defined as cell loss in the second node. There are two motivations for this choice:

1. The cell loss at the first node can be determined analytically.
2. The definition of an event B is less complex when focusing on one node.

The background traffic consists of both joining and crossing traffic. The choice of the type of traffic is derived from a model used by ir B.J. van Rijnsoever. This model did not contain joining traffic. A source representing the joining traffic is added to illustrate the possibility of this type of traffic.

The joining traffic is a TwoStateMMBP source with as sojourn times in the High and Low state respectively 250 and 750 timeslots. In the High period batches are generated independent from slot to slot with a maximum batchsize of four cells. In the Low state no cells are generated at all.

The crossing traffic sources are also of the type TwoStateMMBP. Both crossing traffic sources are identical to each other. In both High and Low states cells are generated in batches with a maximal batchsize of four. In the Low state the traffic load is low. The mean sojourn time of the low state is very high approximately 40000 timeslots. The mean sojourn time in the High state on the contrary is short, approximately 140 timeslots, and the traffic load is such that the buffer overflow could occur.

The type of VC source is different from case to case to illustrate the influence of the type of VC traffic. Which sources are simulated is reviewed in appendix D. A schematic overview of the model and the exact parameters can also be found in this appendix.

6.5 Simulation results

To study the gain and the effect of various parameters of Restart on the gain, the model is simulated in four cases. These four cases are explained in appendix D. The detailed results of the simulation can be found in this appendix. With the results of the four cases three effects on the gain are envisioned:

1. Effect of a smaller probability of event A.
2. Increasing the hysteresis effect.
3. Different choices of event B.

Case 1 is the case to which all other cases are compared. The VC source is a ON OFF source where in the ON state cells are generated in batches independent from slot to slot. To illustrate the characteristic of Restart that small probability lead to an higher gain, case 2 is simulated. In case 2 the probability of cell loss is forced to be higher, by extending the on period. This will lead to an higher probability and a decrease of the gain. The second effect to be studied is the increase of the hysteresis effect. To study this effect case 1 is simulated with two levels of hysteresis(T'). There are two different definitions of event B applied. They are simulated with cases 3 and 4. Case 3 takes into account the status of the source generating crossing traffic. Case 4 tries to improve the gain by taking into account the period in which cells can be lost.

6.5.1. Effect of a smaller probability of event A

This effect is illustrated by comparing the simulation results of case 1.2 and case 2.1. In case 2.1 the time that the VC source is in the state where it generates cells is increased. Therefore the probability of event A is higher with case 2.1. Comparing the results lead to the following gains

table 6.3

case	$\hat{P}$	acc 95%	$\sqrt{\frac{K_1}{K_2}+b}$	G_1	G_2	G_3
1.2	1.2 E-07	4.8E-8	8.2	87	77	43
2.1	5.0 E-07	1.7E-7	6.0	58	51	28

Table 6.3 illustrates that the gain is reduced when the probability is greater. This is a characteristic of Restart. When small probabilities have to be estimated, this characteristic is very useful.

The factor $\sqrt{\frac{K_1}{K_2}+b}$ gives the reduction of the gain due to the correlation factors K1 and K2, and the differences in system states belonging to occurrences of event B. The decrease of the gain is not due to this factor since this factor is lower in case 2.1.

6.5.2 Increasing of the hysteresis effect

Hysteresis has been applied to reduce the restoration costs. Besides reducing these costs, the variance reduction should be improved by reducing the factor K1 and b. In this paragraph is envisioned if the gain is improved as the hysteresis effect is increased. This is done by simulating case 1 in two sub-cases(case1.1 and case1.2). In case 1.2 the hysteresis level T' is lowered from 4 to 3. The exact results can be found in appendix D. Below the values of K1, b and the Gain are illustrated.

table 6.4

case	$\hat{P}$	acc 95%	$\hat{P}_2'$	acc 95%	K1	b	acc b	G_1	G_2	G_3
1	1.6 E-07	4.7E-8	5.9E-3	1.7E-3	36.7	42.7	12	89	78	45
2	1.2 E-07	4.8E-8	4.5E-3	1.7E-3	35.4	61.4	9	87	77	43

From the results in table 6.4 can be seen that hysteresis does reduce K1. But instead of a decrease in b, the results show an increase in b. Looking closer to the results, P2' decreases. The decrease of P2' is contradictory to the article [Vil91b]. The accuracy of the results is not accurate enough to draw conclusions from these results, concerning the decrease of P2'. The accuracy of b is calculated by:

$$acc_b = \frac{d(b)}{d(P_2')} acc_{P_2'} = \frac{b}{P_2'} \Delta P_2' \qquad (49)$$

In this calculation other inaccuracies are not taken into account. These simulation runs are not accurate enough to draw conclusions from concerning the parameters b and P2', but it stresses the need to perform further research on the effect of hysteresis on those parameters.

The gain G_3 is reduced in this case by hysteresis. This can be explained by the assumption that the number of intervals [F,G) in [B,D) is close to 1. If hysteresis is applied extra timeslots are repeatedly simulated that don't contribute to the variance reduction. This example stresses the need for caution to apply hysteresis.

6.5.3 Different choices of event B

To improve the gain the choice of event B is changed. In cases 1 and 2 event B occurred when the occupance of the queue of the second node becomes above some level T. A change in event B should lead to a decrease of the factor b, to improve the gain. The factor b is a measure of the influence on the gain, of the differences between the system states belonging to the occurrences of event B. This means that the variance in the expectation of the number of events A conditioned on event B should be minimized. To decrease b more system variables must be involved in the definition of event B.

One system variable that influences the expected number of events A conditioned on event B, is the state of source 4. If source 4 is in the High state, when the queue of node 2 exceeds some level T, more events A are likely to occur. This idea is the basis of the definition of event B in case 3.

This case is simulated with T=5 and T'=3 therefore it can be compared with case 1.2. The exact results can be found in appendix D. Below the gains are reviewed.

table 6.5

case	$\hat{P}$	acc 95%	$\sqrt{\frac{K_1}{K_2}+b}$	b	acc b	G_1	G_2	G_3
1.2	1.2E-7	4.8E-8	8.2	61	9	87	77	43
3.1	1.8E-7	5.7E-8	6.6	33	9	89	78	40

This definition did lead to a decrease of the factor b but not to a increase of G_3. This can be explained by the fact that the choice of event B affects also the parameters P1' and P2', which has lead to an reduction of the gain. Taking the state of source 4 into account by the definition of B can improve the gain but the parameters R, T and T' should be adjusted afterwards.

A second system variable that influences the gain is the state of the VC source. If the VC source does not generate cells(In the Low period) and no cells are in node 1, no events A will occur. This fact is the basis of the second choice of event B represented by case 4. With this case the repeated simulation trials are carried out if two conditions are fulfilled:

1. The number of cells queued in node 2 exceeds some level T.
2. The VC source is in the High state or cells of the VC source are in node 1.

This case is simulated for several levels T and T'. The exact results can be found in appendix D. The Gains are reviewed below.

Table 6.6

case	$\hat{P}$	acc 95%	b	K1	G_1	t_r/t_w	G_2	G_3
4.1	1.7E-7	7E-8	36	19	96	0.89	86	44
4.2	8.8E-8	4E-8	45	17	123	0.89	109	52
4.3	1.1E-7	5E-8	24	28	141	0.93	131	55
4.4	2E-7	9E-8	25	32	101	0.93	94	36
4.5	1.3E-7	5E-8	17	37	170	0.93	157	45

The results show that this choice of the definition can lead to an improvement of the gain. The improvement is dependent on the hysteresis level used. An extra aspect of this definition of event B is the increase of the factor t_r/t_w. This seems contradictory with the extra evaluation costs. But the increase of the factor can be explained by the fact that the change in the definition of event B led to a change of P1'. A decrease of P1' lead to an decrease of the number of times a system has to be restored. Then the ratio t_r/t_w increases.

The best gain achieved is in case 4.3. In this case the level T is 5 and the hysteresis level T' is 4. The gain achieved is 55. This means that simulating with restart is 55 times faster than without for this model.

6.6 Review of the Results

The application of Restart on the model described in 6.4 has made clear that a lower probability lead to an higher gain of Restart. This characteristic of Restart is very useful for simulating low probabilities. The simulation of the cases, described in appendix D, made clear that an increase of hysteresis not necessarily lead to an improve of the gain. This stresses the need of caution to apply hysteresis. The simulation runs done are not accurate enough to draw conclusions on the decrease of P2' and b. Longer simulation runs must be carried out to study the effect of hysteresis on the parameters P2' and b.

The gain can be improved by an appropriate choice of the event B. In case 3 the state of source 4 is taken into account and in case 4 the period that cells of the VC source can enter node 2. The change in the definition of event B can lead to a change in the parameters P1' and P2'.

A problem is to define the levels T, T' and R such that P1 and R become as close as possible to the optimum values. Further study on the effect of hysteresis and the optimal parameter values should give insight in this problem.

From the cases simulated the best gain achieved is 55. This means that the probability P{A} can be estimated 55 times faster with the same accuracy with Restart than without Restart.

6.7 References

[Jai91]: Jain R.,"The art of computer system performance analysis", John Wesley(1991).

[Vill91]: Villen-Altamirano, M. et al, "Restart a method for accelerating rare event simulations", proc. ITC-13, 1991, p 71-77.

[Vil91b]: Villen-Altamirano, M. et al, "Accelerated simulation of rare events using restart method with hysteresis", *Telecommunication Services for developing economics*, published by Elseviers Science Publishers, 1991, p 675-86.

Chapter 7 Conclusions and recommendations

Conclusions

The assignment was to study and implement a simulation technique to cope with the problem of estimating low probabilities. This problem arises when simulating models for virtual connections in ATM networks to evaluate the performance requirements. The performance measures of a virtual connection are cell loss probability and delay jitter due to queuing in ATM switches and multiplexers.

From the techniques studied, Restart is the most promising technique to cope with the problem of low probabilities. Restart is a method which estimates the probability of rare events. It uses repetitive simulation trials when the system exceeds some threshold.

To implement Restart a simulation environment SIMMUST is developed using an object oriented approach. With SIMMUST it is possible to simulate a diversity of VC models.

Simmust is the basis of the implementation of Restart. Restart is implemented including an hysteresis effect. Hysteresis is used to reduce the effort spend on restoring the system state. Hysteresis should improve the variance reduction by reducing the factors K1 and b.

The performance of Restart is measured by comparing the costs times variance of the simulation using Restart and straight simulation. Restart with hysteresis is applied on a model of a virtual connection with two nodes. Restart is used to estimate the probability of cell loss in the second node. The application of Restart on this model has made clear that a lower probability to be estimated lead to an higher gain of Restart. This characteristic of Restart is very useful for estimating low probabilities.

The simulations with an increased hysteresis effect has made clear that hysteresis not necessarily lead to an improvement of the gain of the method. This stresses the need for caution to apply hysteresis. Instead of a decrease of the factor b the simulations show an increase of b by an increase of the hysteresis effect. The simulations done are not accurate enough to draw conclusions concerning the decrease of P2' and b. Further study must be done on the effect of hysteresis on the parameters b and P2'.

The gain can be improved by an appropriate definition of event B. Two improvements on the definition of event B have been simulated. Taking into account the period when cells of the VC source can enter node 2 has led to the best gain. The best gain achieved on the model simulated was 55. This means that the probability with a certain accuracy can be estimated 55 times faster by using Restart on that model.

Recommendations

Restart can be applied to estimate low probabilities of performance measures in a model of a VC. The hysteresis effect has led to a reduction of the gain by the simulation of one particular model. This plus the effect of hysteresis on the parameters of Restart should be studied further. This study should give insight in the question if or when to apply hysteresis.

The gain of Restart can be improved by changes in the definition of event B. These changes must be matched onto the simulated model. This aspect of Restart is an interesting topic for further study. One possibility of a new definition of event B on the model simulated is a combination of the two cases presented.

Another topic for further research on Restart is to study the effect when different events B within one simulation are defined to estimate the probability of an event A. If a set of events B is defined the conditional expectation of event A can be determined for each event B separately. These values can be combined to estimate the probability of event A.

In appendix B.4 two recommendations are described to improve the processes of source components in the model description. The first recommendation involves the scheduling of sources. And has the purpose to exclude typical behaviour caused by the scheduling algorithm of the underlying software. The second recommendation involves a change of the processes of sources generating batches of cells. Now the behaviour is such that all cells in a batch are scheduled after each other. This can be altered by scheduling the sources, that generate batches of cells, again in the same timeslot after each cell is generated. In this way other sources can generate cells in between. This results in a mix of the traffic on the link. Both recommendations improves the behaviour of the model. For more details I refer to B.4.

Restart is now used to estimate the cell loss probability in one node. Extending Restart on estimating the probability other events is possible. One example is the estimation of excessive delay.

Only one type of node is defined in the simulation software. An interesting extension is the implementation of priority based schemes of nodes.

A Notations Restart

S(k): System state.
T: System state when the system enters region C.
T' System State when the system enters region C'
L: System state where event A occurs.
B: The first event C after event D occurred in the timeslot having occurred NOT(C) in the previous one.
C: S(k) > T
C': S(k) > T'
F: S(k) > T and in interval [B,D)
G: S(K) < T and in interval [B,D)
D: The event NOT(C') occurred in the timeslot, having C' occurred in the previous one.
P: P{A}
P_1: P{C}
P_2: P{A|C}
P'_1: P{B}
P'_2: expected value of the number of timeslots in interval [B,D] where event A occurs;
a: expected value of the number of timeslots with event C in interval [B,D];
N: Number of timeslots simulated without counting those of the retrials;
N_1: Number of events B which occur in the simulation without counting the retrials.
N_2: Number of events which occur in the whole simulation.
Y_i: A random variable which indicates the number of events A occurred in the i-th interval [B,D]
l: expected value of the number of timeslots in an interval [B,D)
m: expected value of the number of intervals[F,G) in an interval [B,D);
μ: Expected value of the number of timeslots in an interval [F,G)

The following relations between the variables are given:

$P = P_1P_2 = P'_1P'_2$
$P_1 = a\,P'_1$
$P_2 = P'_2/a$
$a = m * \mu$

K_1: K_1 can be defined as the average value of the number of intervals [B,D] which occur in or around a given event B, minus the number of them which would occur if there were independence between near Intervals.
K_2: K_2 can be defined as the average value of the number of events A which occur around a given event A minus the number of them which would occur if there were independence between near instants.

- Y_{ij} is the random variable which indicates the number of events A during the j-th retrial made starting from B_i i.e. from the i-th event B.
- Y_i': the conditional expectation of Y_{ij} to B_i.

$V1 = E[Yi'-P2']^2$
$V2 = E[E[Yij-Yi']^2]$
Vp1 : Estimated variance of P1'
Vp2: Estimated variance of P2'
V: Regeneration event
r: Restoration costs in units of time necessary to simulate a timeslot.

Appendix B
SIMMUST

B1 Introduction

This appendix provides information about the working and structure of the developed simulation software. The software and the units referred to, belong to SIMMUST version 2.

With a unit is meant a Turbo Pascal unit. A Turbo Pascal unit is a collection of constant, datatypes, variables, procedures and functions. Each unit consists of an interface and implementation section. In the interface section the types, variables, procedures and functions that can be used outside the unit are defined. The implementation sections contain the implementation of the procedures and functions. I refer to the Turbo Pascal reference manual for more information on this subject.

This software consists of three Units:

Constant
Modelobj
Perfobj

A procedure that handles the input and initialize the model is also available. Int the next three sections the units are discussed. Section 5 explains the initialization procedure. The input and output specification is discussed in section 6 and 7. Section 8 gives an example and information on starting up the program.

B2. Constant

This unit contains all the constants used in other units and the main program. The constants can be classified to:

Model objects
Performance object
General use

All of these constants could be declared in the other Units and the Main file as well. I have chosen for gathering the constants in one unit for convenience in developing the software. In the finite version the constants can be distributed among the units and the constant unit could be removed.

Included in this unit are the seeds for the random generation. A random generator generates a number which is function of the previous numbers:

$$x_n = f(x_{n-1}, x_{n-2}, \ldots)$$

for example:

$$x_n = (5x_{n-1} + 1) \bmod 32$$

When starting this sequence an initial value must be given. This initial value is called a seed.

The sequence can be regenerated when starting with the same seed. The selection of the seeds must be done carefully. The simulation needs several streams with random numbers. Each stream should be given a different seed. The sequences of the streams should be as non-overlapping as possible. The gathering of correct seeds must be done with a separate program. The seeds are stored in an array which is initialized in the Unit constant. The maximum number of seeds depends on the number of sources and the

number of random streams each source needs. The maximum number of sources depends on the number of servers. In this version, four is determined as the maximum number of random streams one source uses

The seeds are determined for a maximum of 8 servers. With 8 servers there are maximal 36 sources. And for each source a maximum of 4 seeds leads to maximal 144 seeds. These seeds are determined with a separate program. The seeds are equally distributed in the sequence of the random generator. When less then 8 servers are used, the seeds used for the sources are chosen such that the distance between them is maximal. This is done by defining an Offsetsource. This value is an offset that is added to the index of the seeds array. The four seeds for each source are chosen similar to a maximum distance by defining a constant OffsetSeed.

B3. Modelobj:

The unit describes the model objects. This description is object oriented. Each component of the simulation model is declared as an object. The object consists of a data field and methods. This data field describes the attributes of the component. Besides this data field the object consists of methods which are procedures and functions operating on the attributes. One method could be the process invoked by Must. Must requires therefore a parameterless procedure compiled in the far mode. Therefore processes are declared as parameterless far procedures outside the objects. But the process should be considered as a method of the object.

Modelobj contains the type definitions and declarations of all the objects in the simulation model. Also the processes are declared in this unit.

In the next figure the objects and the inheritance relation is given. This inheritance relation is used to implement virtual methods. Virtual methods are methods that are inherited from the ancestor component. For more information on this inheritance relation of objects and virtual methods I refer to the Turbo Pascal reference manual.

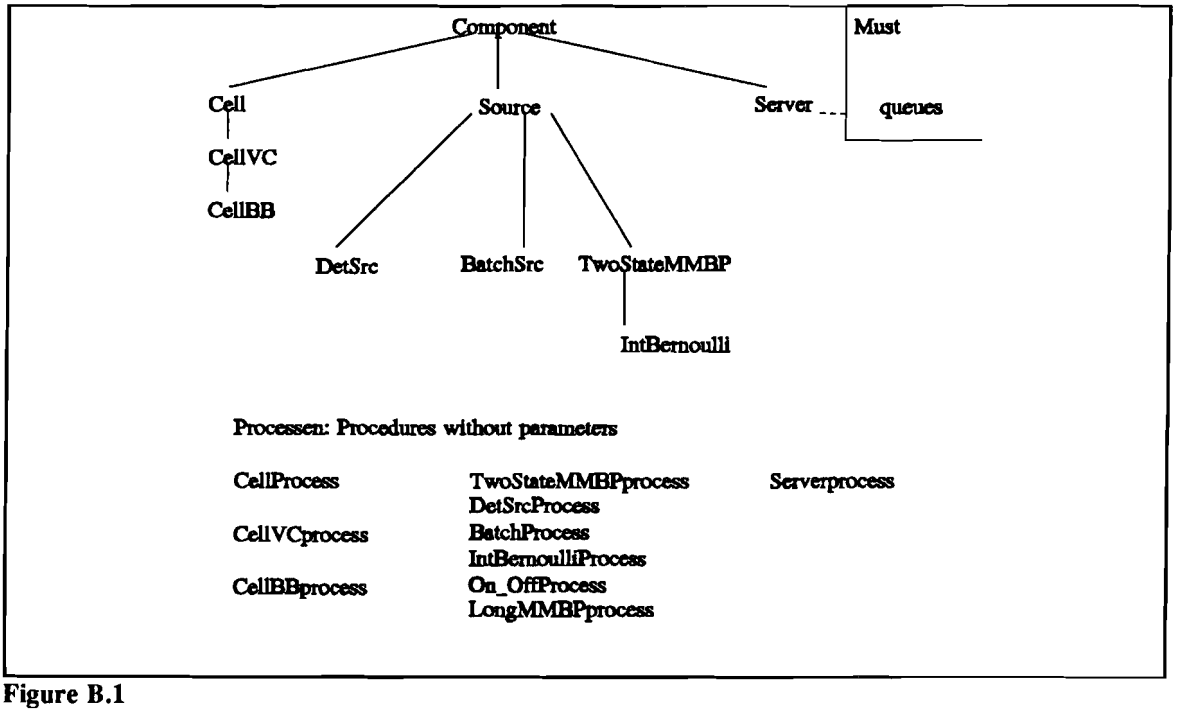


Figure B.1

Below the components are explained.

Component:	The ancestor of all component objects.
Cell:	Represents a cell in the model.
CellVC:	Cell virtual Channel, represents a cell with as additional characteristic that the cell belongs to the data flow under study.
CellBB:	Cell Batch Burst, represents a cell of the Virtual Connection with the additional characteristic that the cell is part of a batch or a burst of cells.
Source:	The ancestor of all source objects.
DetSrc:	Deterministic source, a detSrc generates cells with a deterministic interval.
Batchsrc:	Batch source, this source generates batches of cells where the batch size is independent from slot to slot.
TwoStateMMBP	Two State Markov Modulated Bernoulli Process, a TwoStateMMBP is a double stochastic batchprocess where the rate of cells is determined by the state of a two state Markov chain.

A state diagram is depicted below.

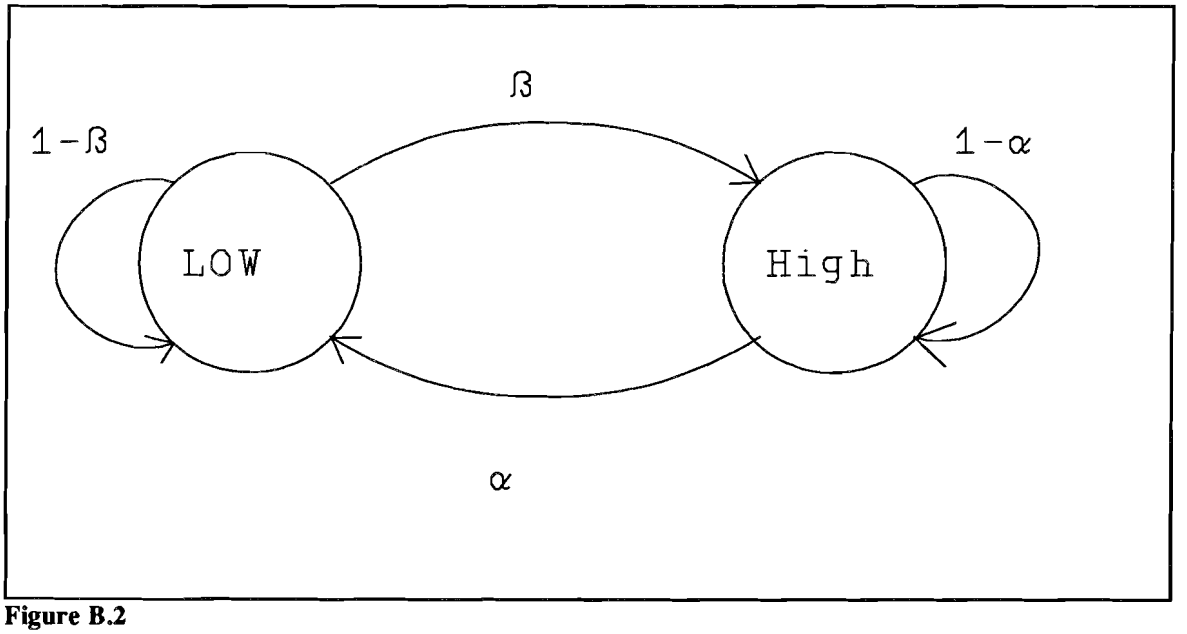


Figure B.2

The state probabilities are:

$$P(low)=\frac{\alpha}{\alpha+\beta} \quad P(High)=\frac{\beta}{\alpha+\beta}$$

The mean sojourn times are:

$$E(low\ interval)=\frac{1}{\beta} \quad E(High\ interval)=\frac{1}{\alpha}$$

Besides a TwoStateMMBPprocess a LongMMBPprocess is programmed. This process is for TwoStateMMBP sources with very long sojourn times. When the TwoStateMMBP process is used the cells are generated for the whole state. This could give memory problems when long sojourn times are given. Therefore a LongMMBPprocess is programmed which is scheduled each timeslot to determine if cells must be generated.

An ON-OFF source can be seen as a special case of a twostateMMBP object. This can be done by initializing the TwoStateMMBP as an ON OFF source. The process activated each time the component becomes active is named the ON_OFFProcess.

This is possible since only the behaviour is different. This behaviour is implemented in a parameterless procedure.

IntBernoulli: Interrupted Bernoulli process, this is a twostateMMBP with as additional characteristic that in the Low state no cells are generated and in the High state cells are generated according to a Bernoulli distribution.

Server: Object representing the server. When cells arrive when the server is occupied, the cells should be placed in a queue. All queue handling is left to Must. The queue belongs to the server is attached to the server by declaring the pointer to the queue as an attribute of the server.

For each component a pointer and an object is declared. Furthermore processes of all objects except 'component' and 'source' are declared.

B.4 Timing of the processes

The scheduling is done by Must, A process is scheduled when the Hold(time) procedure is invoked. Then the process is scheduled to become active after time. Since a time slotted structure is simulated more than one process can be scheduled in one timeslot.

The order in which they are activated is the same order they are scheduled. This can be named First Scheduled First Activated. One exception on this rule is when processes are scheduled with Urgent(see Must manual for more details). This Urgent scheduling is used in the MainProcess. So caution should be applied when Urgent is used in processes.

Two aspects should be considered. Processes like LongMMBP are scheduled each timeslot. When all sources are declared as LongMMBP sources, all sources are scheduled each timeslot. The order in which they are scheduled will not change. So will the order in which they are activated. This might result in typical behaviour caused by the scheduling algorithm of Must and not by the model of the VC. This can be overcome by scheduling sources only when cells will be generated in that timeslot. On this manner the order in which sources are being activated alters during the simulation.

A second consideration on the scheduling algorithm is the generation of batches of cells. Sources which generates batches of cells, activate a number of cells in one timeslot. With the current implementation a source generates the whole batch when it becomes active. On this way all cells of that batch are scheduled after each other. In this order they are placed in the queue of the node. It might be useful to mix the batch of cells from one source with cells from other sources, generated in the same timeslot. This can be done by scheduling the source, which must generate a batch of cells, again in the same timeslot after generation of each cell. In this way the source is scheduled in the same timeslot after already scheduled components. This will give other sources the opportunity to schedule cells between the cells of a batch.

B5. Perfobj

The unit perfobj contains the object with the results of the simulation and the methods to use them. In this object the data field contains the variables needed to calculate the results. The methods are procedures and functions to keep track of the results and to calculate performance measures afterwards. Next the specification of these results is discussed.

B5.1 Introduction

This section will give the specification of the output of the simulation of a virtual connection. Not only what but also how the output is represented will be discussed here. The output values represent the performance of the VC. These parameters can be classified according to the objects of the simulation model:

1. Cell
2. Server
3. Source

Not taken into account are the measures related to queues. But it should be noted that a number of parameters such as mean waiting time and the maximum number of cells in a queue are already available via the Must unit. Only the measures according to the Cell objects represent performance measures of the VC.

B5.2 Cell

Two performance measures are important here

1. Cell loss.
2. Cell delay.

These performance are distributed according three parameters.

- The distribution in time
- The distribution in sequence
- The distribution over the nodes.

In the simulation only the dataflow under study will be measured.

B5.3 Cell loss

By loss is meant that the cell is no longer part of the data flow. Storing only the simulation time cell losses occur is not sufficient since information is also needed on which cell in the sequence is lost.

For each cell loss a triple (time,place,sequencenr) should be saved. During the simulation these values will be saved in a textfile, from which additional software can calculate and plot the distributions.

When batch or burst arrivals are involved, the probability of cell loss given the size of the batch/burst is also important. With the size is meant the number of cells within that burst/batch. In this case there is a quadruple output (time,place,sequencenr,batchsize).

B5.4 Cell Delay

With Cell delay the variance(jitter) is of importance rather than the mean. The delay can be measured in three ways:

1. Measuring the response time after each server and between servers.
2. Measuring the intervals of two successive arrivals of the cell after each server.
3. Measuring the delay of a cell relative to the batch/burst arrival.

The first two measures will be put in an histogram during the simulation. This is

supported by the unit MustHist. These measures are measured for the dataflow under study.

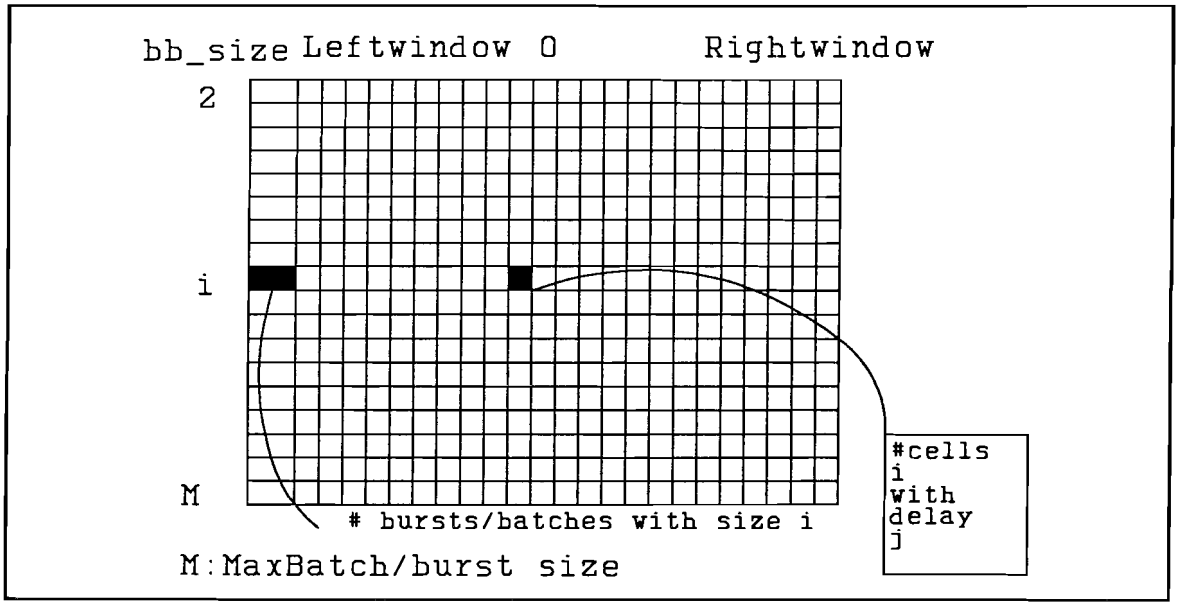
When batch or burst arrivals are involved for the data flow under study it is treated as a special case. For the batch/burst arrivals the third measurement becomes important. The following distribution must be calculated:

P{i-th cell from a batch has a delay k relative to the batch/burst arrival}

The specification of this probability becomes:

$$(V_k:-C\le k\le C:\ (V_i:i>1\wedge i\in sequencenumbers\wedge i\le Maxbatchsize:\ \left(\frac{(\#Batches\ Bursts: arr(cell_i)-arr(cell_1)-(start(cell_i)-start(cell_1)))}{(\#Batches\ Bursts:\ BBsize\ge i)}\right)\ )\ )$$

The values needed for calculation of these probabilities will be stored in a matric. The structure of the matrix is shown in the next figure.



This distribution must be measured only at the destination of the dataflow under study. The values will be stored in a matric. The size of memory needed for such storage is:

(2*Window+2)*MaxBatchSize of LongInt(4 bytes)

An example of the size of such matric is:

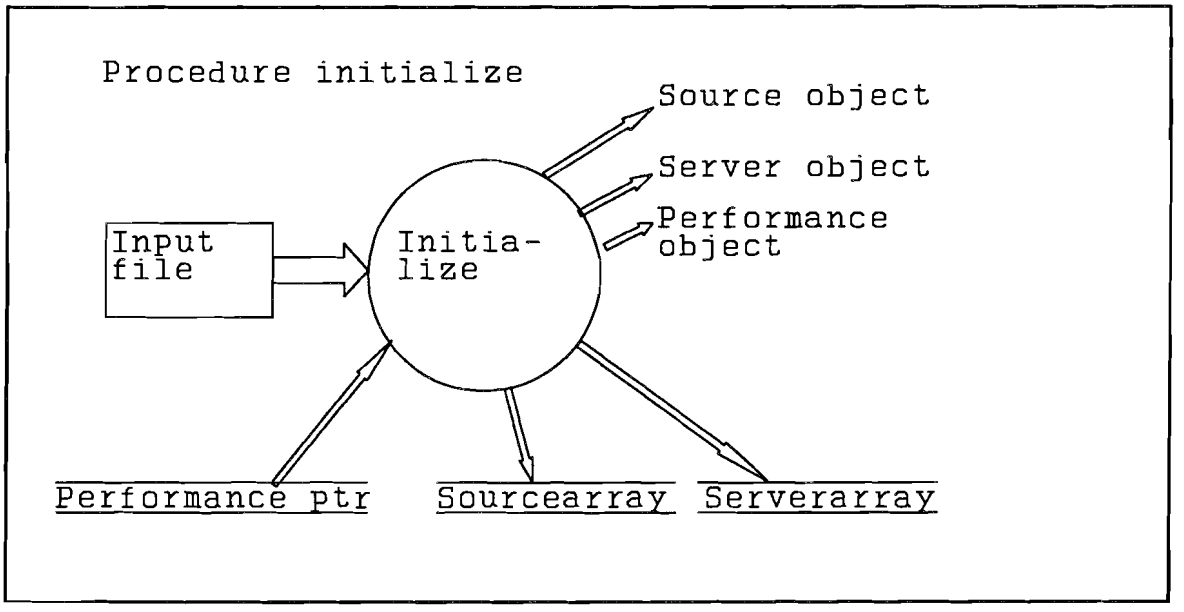
MaxBatchsize 100

Window 200

These values will result in a matrix of 80Kbytes

B6. Procedure Initialize

This procedure reads the input file and initialize the simulation model.



Available after executing the procedure are:

Sourcearray containing the pointers to all sources.
Serverarray containing the pointers to all servers
Performancepointer, pointer to the performance object.

B7. Input specification:

The input parameters are read from an input file. The number of parameters chosen as inputs have been kept to a minimum. Only parameters of sources, servers and histograms are inputs. This is not because other parameters of the model can not be changed. I have chosen for only those inputs that will change given the structure of the model. The structure of model such as the number of servers can be changed also but then the program has to be compiled again. This number of servers can be changed in the unit constant. The input file is a text file which can be edited by every ascii editor.

The first nine lines involves the parameters for the histograms. There are three types of histograms: Interarrival time, response time and cumulative responce time histograms. The first three lines for the histograms with interarrival times. The second three lines for the histograms with response times. And third three lines with the parameters of the cumulative response times. Every type of histogram has three parameters:

Number of classes
Lower bound
Class width.

See must manual for more details on histograms.

After these, the parameters of the servers are listed. They are listed for each server in the model. Starting with the first server to the last server. For each server three lines must be in the file. The first line must contain the word 'server'. This can be in capitals or non-capitals or a combination. The next line must contain the buffersize as the number of cells that can be queued. The third line must contain the servicetime in timeslots. For example the following three lines are for a deterministic server with a service time of 1 timeslot and a queue in which maximal 50 cells can be queued.

ServerNode1
50
1

After the server parameters, the parameters of the sources are listed. First the parameters of the VC source are listed. This starts with a line containing the word 'source'. The next line is for the number of cells to be generated by the source on the VC connection. The next line contains the type of the source. The following types are implemented:

DetSrc
BatchSrc
TwoStateMMBP
ON_OFF
LongMMBP.

After the type the parameters are listed. For each parameter a line is reserved (for details see B7.1). These differ for each type(see input specification for details). The lines containing a batch distribution should be in the format Must required. For more details I refer to appendix A in the Must reference manual. After the Vc source the sources simulating background traffic are listed. Each source starts with a line containing the word 'source'. The next line contains the type of the source. Then the parameters are listed. These differ for each type(see input specification for details). The lines containing a batch distribution should be in the format Must required. For more details I refer to appendix A in the Must reference manual.

The sources must be declared in a fixed order. The reason for this is that routing information is derived from this order. When a source is not used it must be declared with the type 'Empty'. The sources must be in the following order:

- First the VC source.
- Then the sources joining on the first node. Starting with the source that travels from the first node to the last node. Ending with the source crossing the first node.
- Then the sources joining on the second node. Starting with the source that travels from the first node to the last node. Ending with the source crossing the second node.
- And so on.
- Until the last source crossing the last node.

The next paragraph gives the input specification In section 8 an example of a VC model is given with the input file.

B7.1 specification of input file

<Parameters histograms inter arrival times>
<Parameters histograms response time>
<Parameters histograms cumulative response times>
<Server 1>
<Server 2>
..
<Source Virtual connection>
<Source background>
<Source background>
..
EOF

<Parameters histograms inter arrival times>
<Parameters histograms response time>
<Parameters histograms cumulative response times> : <Parameters>
<Parameters>: Number of classes : Integer
Lowerbound : Real
Classwidth : Real

<server>: Server
<buffer size>
<servicetime>
<buffersize>,
<servicetime>: Integer

<Source virtual connection>:Source
<type>
<nr of cells>
<attributes>
<Source background>: Source
<type>
<attributes>
<type>: {DetSrc, BatchSrc, TwoStateMMBP, Intbernoulli, On_Off, LongMMBP, Empty}

DetSrc<attributes>
Interval: Integer
BatchSrc<atrributes>
Batchdistribution: string
TwostateMMBP<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
BatchDistribution High: String
BatchDistribution Low: String
Interval between Batches: Integer
IntBernoulli<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
Probability(succes): Real
On_Off<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
Interval : Integer
TwostateMMBP<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
BatchDistribution High: String
BatchDistribution Low: String
Interval between Batches: Integer
Empty<attributes>

B8. Output specification

The output differs according two situations:

a. VC source generates no batches or bursts with cells. With the sources implemented sofar this is only when the deterministic source is chosen.

b. VC source generates batches or bursts with cells.

ad a.

Cell loss:

File with of every cell loss from the VC the triple(time,queuename,sequencenr).

Cell delay:

For every server and for the response times between servers an histogram with response times of each cell of the VC.

For every server an histogram with the intervals of two successive cell arrivals.

The histograms are standard exported to the file 'Histfile'. The other measures to the file 'Results'.

ad b.

Cell loss:

File with for every cell loss from the VC a quadruple (time,queuename,sequence nr,batch/burstsize).

Cell Delay

For every server and for the response times between servers an histogram with response times of each cell of the VC.

For every server an histogram with the intervals of two successive cell arrivals.

One matrix with:

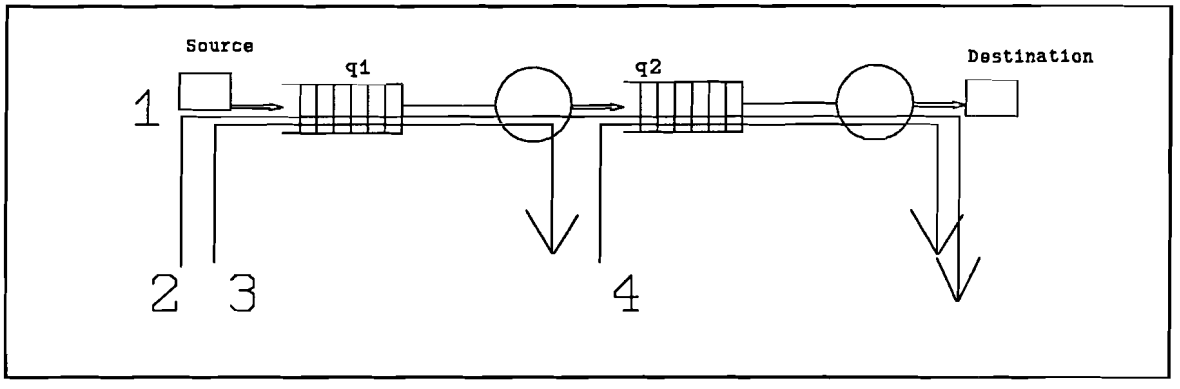
P{i-th cell from a batch has a delay k relative to the batch arrival},

This matrix is exported standard to the file 'Arrfile'. The probabilities are calculated and listed in 'Results'.

The generated histograms are saved in ascci format to 'Histfile'.

B9. Example

From the model depicted in the figure below the input file is listed.



Occupance related to the service time
source1 0.1
source2 0.3 (high:0.5 low:0.1)
source3,
source4 0.4 (high:0.6 low:0.2)
type
source1 Deterministic source.
source2-4:

Markov modulated Batch processes. With binomonal batch size distributions. The batch sizes are independent from slot to slot. Maximum batch size is 7. Mean sojourn times in the states are 100 slots.

service time 1 slot
buffersize 40

The input needed to initialize this model is given on the next page. The output is presented also.

How to start SIMMUST

From the Dos commandline start with:
SIMMUST <return>

Then the inputfile is asked.
name_inputfile <return>

Then the status of the system and the time the simulation is started is given.

How to trap the simulation

Press the <spacebar>. Then the time the simulation is trapped is given. When pressing <return> the status of the system and all histograms are scrolled on the screen. After that a question pops up if the simulation should be continued or not. When answering with 'n' or 'N' the simulation is stopped.

The output by the trap can be changed by programming the trap procedure.

Rapport:
Inputfile:

37
0
1
42
0
1
42
2
2
Server 1
40
1
Server2
40
1
Source 1
100001
DetSrc
10
Source2
TwoStateMMBPsrc
0.01
0.01
Discrete(0, 5.9526070777E-01, 1, 3.2052499649E-01, 2, 7.3967306883E-02, 3, 9.4829880620E-03, 4, 7.2946062015E-04, 5, 3.3667413238E-05, 6, 8.6326700609E-07, 7, 9.4864506164E-09)
Discrete(0, 9.0418511876E-01, 1, 9.1728925092E-02, 2, 3.9882141344E-03, 3, 9.6333674744E-05, 4, 1.3961402137E-06, 5, 1.2140349684E-08, 6, 5.8649032291E-11, 7, 1.2142656789E-13)
1
Source3
TwoStateMMBPsrc
0.01
0.01
Discrete(0, 5.3403969327E-01, 1, 3.5046354871E-01, 2, 9.8567873073E-02, 3, 1.5401230168E-02, 4, 1.4438653282E-03, 5, 8.1217424713E-05, 6, 2.5380445223E-06, 7, 3.3991667709E-08)
Discrete(0, 8.1634945819E-01, 1, 1.6807194727E-01, 2, 1.4829877701E-02, 3, 7.2695478925E-04, 4, 2.1381023213E-05, 5, 3.7731217435E-07, 6, 3.6991389642E-09, 7, 1.5542600690E-11)
1
Source4
TwoStateMMBPsrc
0.01
0.01
Discrete(0, 5.3403969327E-01, 1, 3.5046354871E-01, 2, 9.8567873073E-02, 3, 1.5401230168E-02, 4, 1.4438653282E-03, 5, 8.1217424713E-05, 6, 2.5380445223E-06, 7, 3.3991667709E-08)
Discrete(0, 8.1634945819E-01, 1, 1.6807194727E-01, 2, 1.4829877701E-02, 3, 7.2695478925E-04, 4, 2.1381023213E-05, 5, 3.7731217435E-07, 6, 3.6991389642E-09, 7, 1.5542600690E-11)
1

Histgrams

```
Results:
 in queuequeue       1sequencenr2715
 in queuequeue       2sequencenr29274
 in queuequeue       2sequencenr43604

Interarrival time after server:0

Total                    Excluding zero           Minimum            0.000
 Entries       100000     Entries        99999    90% Quantile       9.900
 Mean           10.000     Mean           10.000  95% Quantile       9.950
 Std.Deviation   0.032     Std.deviation   0.000  Maximum           10.000


     Range Numb Perc Cum % |  10|  20|  30|  40|  50|  60|  70|  80|  90| 100|
                           |
<=    3.00     0  0.0   0.0 |
<=    4.00     0  0.0   0.0 |
<=    5.00     0  0.0   0.0 |
<=    6.00     0  0.0   0.0 |
<=    7.00     0  0.0   0.0 |
<=    8.00     0  0.0   0.0 |
<=    9.00     0  0.0   0.0 |
<=   10.0099999100.0 100.0 |**************************************************
<=   11.00     0  0.0 100.0 |                                                 +
<=   12.00     0  0.0 100.0 |                                                 +
<=   13.00     0  0.0 100.0 |                                                 +
<=   14.00     0  0.0 100.0 |                                                 +
<=   15.00     0  0.0 100.0 |                                                 +
<=   16.00     0  0.0 100.0 |                                                 +
<=   17.00     0  0.0 100.0 |                                                 +
<=   18.00     0  0.0 100.0 |                                                 +
<=   19.00     0  0.0 100.0 |                                                 +
<=   20.00     0  0.0 100.0 |                                                 +
<=   21.00     0  0.0 100.0 |                                                 +
<=   22.00     0  0.0 100.0 |                                                 +
<=   23.00     0  0.0 100.0 |                                                 +
<=   24.00     0  0.0 100.0 |                                                 +
<=   25.00     0  0.0 100.0 |                                                 +
<=   26.00     0  0.0 100.0 |                                                 +
<=   27.00     0  0.0 100.0 |                                                 +
<=   28.00     0  0.0 100.0 |                                                 +
<=   29.00     0  0.0 100.0 |                                                 +
<=   30.00     0  0.0 100.0 |                                                 +
<=   31.00     0  0.0 100.0 |                                                 +
<=   32.00     0  0.0 100.0 |                                                 +
<=   33.00     0  0.0 100.0 |                                                 +
<=   34.00     0  0.0 100.0 |                                                 +
<=   35.00     0  0.0 100.0 |                                                 +
<=   36.00     0  0.0 100.0 |                                                 +
Interarrival time after server:1

Total                    Excluding zero           Minimum            1.000
 Entries        99999     Entries        99999    90% Quantile      12.717
 Mean           10.000     Mean           10.000  95% Quantile      14.124
 Std.Deviation   2.669     Std.deviation   2.669  Maximum           27.000


     Range Numb Perc Cum % |  10|  20|  30|  40|  50|  60|  70|  80|  90| 100|
                           |
<=    0.00     0  0.0   0.0 |
<=    1.00   100  0.1   0.1 |
<=    2.00   379  0.4   0.5 |
<=    3.00   809  0.8   1.3 |+
<=    4.00  1436  1.4   2.7 |*
<=    5.00  2294  2.3   5.0 |*++
<=    6.00  3772  3.8   8.8 |**++
<=    7.00  5576  5.6  14.4 |***++++
<=    8.00  8703  8.7  23.1 |****  ++++++
<=    9.0015094 15.1  38.2 |********   ++++++++
<=   10.0024454 24.5  62.6 |************      ++++++++++++
<=   11.0015189 15.2  77.8 |********                    +++++++++
<=   12.00  8518  8.5  86.3 |****                              +++++
<=   13.00  5125  5.1  91.4 |***                                   ++++
<=   14.00  3284  3.3  94.7 |**                                      ++
<=   15.00  2144  2.1  96.9 |*                                       ++
<=   16.00  1311  1.3  98.2 |*                                         ++
<=   17.00   775  0.8  99.0 |                                           +
<=   18.00   520  0.5  99.5 |                                           ++
<=   19.00   269  0.3  99.8 |                                            +
<=   20.00   134  0.1  99.9 |                                            +
<=   21.00    68  0.1 100.0 |                                            +
<=   22.00    19  0.0 100.0 |                                            +
<=   23.00    14  0.0 100.0 |                                            +
<=   24.00     9  0.0 100.0 |                                            +
<=   25.00     0  0.0 100.0 |                                            +
<=   26.00     1  0.0 100.0 |                                            +
<=   27.00     2  0.0 100.0 |                                            +
<=   28.00     0  0.0 100.0 |                                            +
<=   29.00     0  0.0 100.0 |                                            +
<=   30.00     0  0.0 100.0 |                                            +
<=   31.00     0  0.0 100.0 |                                            +
<=   32.00     0  0.0 100.0 |                                            +
<=   33.00     0  0.0 100.0 |                                            +
<=   34.00     0  0.0 100.0 |                                            +
<=   35.00     0  0.0 100.0 |                                            +
<=   36.00     0  0.0 100.0 |                                            +
```

Interarrival time after server:2

Total		Excluding zero		Minimum	1.000
Entries	99997	Entries	99997	90% Quantile	14.004
Mean	10.000	Mean	10.000	95% Quantile	15.924
Std.Deviation	3.646	Std.deviation	3.646	Maximum	35.000

	Range	Numb	Perc	Cum %	10	20	30	40	50	60	70	80	90	100
<=	0.00	0	0.0	0.0										
<=	1.00	372	0.4	0.4										
<=	2.00	1042	1.0	1.4	*									
<=	3.00	1767	1.8	3.2	*+									
<=	4.00	2876	2.9	6.1	*++									
<=	5.00	4012	4.0	10.1	**+++									
<=	6.00	5445	5.4	15.5	*** ++++									
<=	7.00	7271	7.3	22.8	**** ++++									
<=	8.00	9375	9.4	32.2	***** ++++++									
<=	9.00	12173	12.2	44.3	****** +++++++									
<=	10.00	14898	14.9	59.2	******* +++++++++									
<=	11.00	11850	11.9	71.1	****** +++++++									
<=	12.00	8567	8.6	79.7	**** +++++									
<=	13.00	5965	6.0	85.6	*** ++++									
<=	14.00	4373	4.4	90.0	** +++									
<=	15.00	2960	3.0	92.9	* ++									
<=	16.00	2221	2.2	95.2	* +++									
<=	17.00	1507	1.5	96.7	* +									
<=	18.00	1105	1.1	97.8	* ++									
<=	19.00	751	0.8	98.5	+									
<=	20.00	535	0.5	99.1	++									
<=	21.00	322	0.3	99.4	+									
<=	22.00	205	0.2	99.6	+									
<=	23.00	142	0.1	99.7	+									
<=	24.00	105	0.1	99.8	+									
<=	25.00	62	0.1	99.9	+									
<=	26.00	33	0.0	99.9	+									
<=	27.00	28	0.0	100.0	+									
<=	28.00	17	0.0	100.0	+									
<=	29.00	5	0.0	100.0	+									
<=	30.00	5	0.0	100.0	+									
<=	31.00	1	0.0	100.0	+									
<=	32.00	3	0.0	100.0	+									
<=	33.00	2	0.0	100.0	+									
<=	34.00	1	0.0	100.0	+									
<=	35.00	1	0.0	100.0	+									
<=	36.00	0	0.0	100.0	+									

responce time from queue+server:1

Total		Excluding zero		Minimum	1.000
Entries	99999	Entries	99999	90% Quantile	25.683
Mean	8.604	Mean	8.604	95% Quantile	33.577
Std.Deviation	10.509	Std.deviation	10.509	Maximum	41.000

	Range	Numb	Perc	Cum %	10	20	30	40	50	60	70	80	90	100
<=	0.00	0	0.0	0.0										
<=	1.00	25262	25.3	25.3	**********									
<=	2.00	17216	17.2	42.5	********* +++++++++									
<=	3.00	9221	9.2	51.7	***** ++++++									
<=	4.00	5576	5.6	57.3	*** ++++									
<=	5.00	3763	3.8	61.0	** +++									
<=	6.00	2940	2.9	64.0	* ++									
<=	7.00	2441	2.4	66.4	* ++									
<=	8.00	2185	2.2	68.6	* ++									
<=	9.00	1940	1.9	70.5	* ++									
<=	10.00	1762	1.8	72.3	* ++									
<=	11.00	1737	1.7	74.0	* ++									
<=	12.00	1570	1.6	75.6	* ++									
<=	13.00	1472	1.5	77.1	* ++									
<=	14.00	1387	1.4	78.5	* +									
<=	15.00	1330	1.3	79.8	* ++									
<=	16.00	1188	1.2	81.0	* +									
<=	17.00	1137	1.1	82.1	* ++									
<=	18.00	1073	1.1	83.2	* ++									
<=	19.00	1059	1.1	84.3	* +									
<=	20.00	994	1.0	85.3	++									
<=	21.00	952	1.0	86.2	+									
<=	22.00	870	0.9	87.1	++									
<=	23.00	859	0.9	87.9	+									
<=	24.00	807	0.8	88.7	+									
<=	25.00	763	0.8	89.5	++									
<=	26.00	725	0.7	90.2	+									
<=	27.00	714	0.7	90.9	+									
<=	28.00	735	0.7	91.7	++									
<=	29.00	661	0.7	92.3	+									
<=	30.00	595	0.6	92.9	+									
<=	31.00	605	0.6	93.5	++									
<=	32.00	582	0.6	94.1	+									
<=	33.00	563	0.6	94.7	+									
<=	34.00	546	0.5	95.2	++									
<=	35.00	511	0.5	95.7	+									
<=	36.00	557	0.6	96.3	+									
<=	37.00	638	0.6	96.9	+									
<=	38.00	626	0.6	97.6	++									
<=	39.00	741	0.7	98.3	+									
<=	40.00	848	0.8	99.2	++									
<=	41.00	848	0.8	100.0	+									
<=	42.00	0	0.0	100.0	+									

```
responce time from queue+server:2

Total                    Excluding zero           Minimum          1.000
 Entries        99997     Entries        99997   90% Quantile    22.922
 Mean             7.728   Mean             7.728 95% Quantile    31.721
 Std.Deviation    9.995   Std.deviation    9.995 Maximum         41.000


     Range Numb Perc Cum % |  10|  20|  30|  40|  50|  60|  70|  80|  90| 100|
                           |
<=    0.00    0  0.0   0.0 |
<=    1.0032419 32.4  32.4 |*****************
<=    2.0015982 16.0  48.4 |********       +++++++++
<=    3.00 7691  7.7  56.1 |****                   +++++
<=    4.00 4616  4.6  60.7 |**                        +++
<=    5.00 3455  3.5  64.2 |**                          +++
<=    6.00 2804  2.8  67.0 |*                             ++
<=    7.00 2362  2.4  69.3 |*                              +++
<=    8.00 2092  2.1  71.4 |*                                ++
<=    9.00 1943  1.9  73.4 |*                                 ++
<=   10.00 1780  1.8  75.1 |*                                  ++
<=   11.00 1581  1.6  76.7 |*                                   +
<=   12.00 1551  1.6  78.3 |*                                   ++
<=   13.00 1404  1.4  79.7 |*                                    ++
<=   14.00 1332  1.3  81.0 |*                                     ++
<=   15.00 1343  1.3  82.4 |*                                      +
<=   16.00 1251  1.3  83.6 |*                                      ++
<=   17.00 1129  1.1  84.7 |*                                       +
<=   18.00 1086  1.1  85.8 |*                                       ++
<=   19.00  985  1.0  86.8 |                                        +
<=   20.00  908  0.9  87.7 |                                        ++
<=   21.00  872  0.9  88.6 |                                         +
<=   22.00  766  0.8  89.4 |                                         ++
<=   23.00  700  0.7  90.1 |                                          +
<=   24.00  651  0.7  90.7 |                                          +
<=   25.00  655  0.7  91.4 |                                          ++
<=   26.00  638  0.6  92.0 |                                           +
<=   27.00  585  0.6  92.6 |                                           +
<=   28.00  543  0.5  93.1 |                                           ++
<=   29.00  540  0.5  93.7 |                                            +
<=   30.00  488  0.5  94.2 |                                            +
<=   31.00  486  0.5  94.6 |                                            +
<=   32.00  498  0.5  95.1 |                                            ++
<=   33.00  427  0.4  95.6 |                                             +
<=   34.00  402  0.4  96.0 |                                             +
<=   35.00  405  0.4  96.4 |                                             +
<=   36.00  439  0.4  96.8 |                                             +
<=   37.00  468  0.5  97.3 |                                             ++
<=   38.00  467  0.5  97.7 |                                              +
<=   39.00  520  0.5  98.3 |                                              +
<=   40.00  737  0.7  99.0 |                                              ++
<=   41.00  996  1.0 100.0 |                                               +
<=   42.00    0  0.0 100.0 |                                               +
cumulative responce time after server:2

Total                    Excluding zero           Minimum          2.000
 Entries        99997     Entries        99997   90% Quantile    39.929
 Mean            16.332   Mean            16.332 95% Quantile    47.264
 Std.Deviation   15.701   Std.deviation   15.701 Maximum         82.000


     Range Numb Perc Cum % |  10|  20|  30|  40|  50|  60|  70|  80|  90| 100|
                           |
<=    0.00    0  0.0   0.0 |
<=    2.0010519 10.5  10.5 |*****
<=    4.0019282 19.3  29.8 |**********+++++
<=    6.0010334 10.3  40.1 |*****         ++++++
<=    8.00 6374  6.4  46.5 |***                ++++
<=   10.00 4830  4.8  51.3 |**                    ++++
<=   12.00 4235  4.2  55.6 |**                       +++
<=   14.00 3690  3.7  59.3 |**                          +++
<=   16.00 3479  3.5  62.7 |**                            ++
<=   18.00 3210  3.2  66.0 |**                              +++
<=   20.00 2962  3.0  68.9 |*                                 ++
<=   22.00 2736  2.7  71.7 |*                                  +++
<=   24.00 2619  2.6  74.3 |*                                    ++
<=   26.00 2382  2.4  76.7 |*                                     ++
<=   28.00 2260  2.3  78.9 |*                                      ++
<=   30.00 2112  2.1  81.0 |*                                       +++
<=   32.00 1918  1.9  82.9 |*                                         +
<=   34.00 1850  1.9  84.8 |*                                         ++
<=   36.00 1783  1.8  86.6 |*                                          ++
<=   38.00 1685  1.7  88.3 |*                                           ++
<=   40.00 1801  1.8  90.1 |*                                            ++
<=   42.00 1988  2.0  92.1 |*                                             ++
<=   44.00 1418  1.4  93.5 |*                                              ++
<=   46.00 1037  1.0  94.5 |*                                               +
<=   48.00  780  0.8  95.3 |                                                ++
<=   50.00  733  0.7  96.0 |                                                 +
<=   52.00  564  0.6  96.6 |                                                 +
<=   54.00  560  0.6  97.1 |                                                 ++
<=   56.00  462  0.5  97.6 |                                                  +
<=   58.00  400  0.4  98.0 |                                                  +
<=   60.00  349  0.3  98.4 |                                                  +
<=   62.00  287  0.3  98.6 |                                                  +
<=   64.00  254  0.3  98.9 |                                                  +
<=   66.00  222  0.2  99.1 |                                                  ++
<=   68.00  195  0.2  99.3 |                                                   +
<=   70.00  185  0.2  99.5 |                                                   +
<=   72.00  143  0.1  99.6 |                                                   +
<=   74.00  113  0.1  99.8 |                                                   +
<=   76.00   79  0.1  99.8 |                                                   +
<=   78.00   65  0.1  99.9 |                                                   +
<=   80.00   62  0.1 100.0 |                                                   +
<=   82.00   40  0.0 100.0 |                                                   +
```

Appendix C
Restart

C1 Introduction

Restart with hysteresis is described in [Vil91b]. The method is implemented using the simulation software sofar. Restart concentrates on the probability of an event A. Therefore no effort has to be spend on other measurements.

C2. Overview of the software

The software consists of 4 units:

- Constant
- Init
- Modelobj
- Perfobj

And a main program with procedures and functions:

- Mytrap
- RestartProcess
- save
- restore
- Fun_B
- Fun_F
- Fun_G
- Fun_D
- Proc_V

Constant unit

This unit contains just like SIMMUST the constants used by the Restart software. Specific constants for RESTART are:

NM_VP1FILE= 'Vp1File' ;(* The name of the file in which the values are stored to calculate vp1 *)

T= .. ;(* Threshold from event C *)

Taccent ; (* threshold for event D *)

Init unit

This unit reads the input file and initialize the model. This unit differs from the procedure initialize of SIMMUST only when it concerns histograms and reading R the number of retrials. The input specification is given in section C4.1 of this appendix.

Modelobj

Modelobj contains the objects and the processes of the simulation model. this unit is adjusted to Restart in two ways:

1. leave out the code and variables involving histograms
2. defining an object for the system state to be saved cq restored.

Perfobj

This unit is completely different from version 2. The attributes of the objects are the results of the restart method and the variables needed to calculate the results. For each event an procedure is defined which performs the necessary actions to be done by that event. The following events are defined:

- Event A
- Event B
- Event C
- Event D
- Event BR
- Event DR
- Event V

Event A: The event of which the probability must be estimated. The occurrences of an event A is forced by setting the boolean variable 'A'. This boolean variable 'A' is checked every timeslot. If 'A' is set special actions are taken and ;A: is reset. The forcing of event A, must be done by the simulation model.

Event B: Entering of interval [B,D). When this interval is entered diverse variables must be updated. This is done in the method EventB.

Event BR: This event is the start of the R-th retrial.

Event DR: This event is the end of the R-th retrial, and thus the end of a primary unit(a set of retrials).

Event V: Event V occurs when the system returns in a regenerative state.

Furthermore a method calc_results is defined. In which the results of the simulation are calculated and written to the result file.

Mytrap

Mytrap is a procedure with no parameters. This procedure is invoked each time the space bar is touched. Then the status is printed to the screen. At this point can be decided if the simulation should be continued or stopped.

RestartProcess

The trap facility can only be used if the Main process is passive. Therefore a restart component is defined which is the main program from which the restartprocess is controlled. In this procedure the restart algorithm is implemented.

C3. Algorithm restart with hysteresis:

```
Restart(inputfile, outputfile,screen);
Pre Inputfile
Post Results
Begin
  Initialize Model
  Print start time
  While Not(Ready) Do
  Begin
    while Not(fun_B) or Ready do
    Begin
      schedule(restart, timeslot)
      proc_V : check for regeneration event
    End (* event B or all cells have been send *)
    If Not(Ready) Then
    Begin
    (* Event B occured *)
      Save system state
      i:= 0
      repeat
        i:= i + 1 (* nr of retrial*)
        while (not(Fun_D)) Do
        Begin
          schedule(restart, timeslot)
          case event
            A: EventA ;
            F: EventF ;
            G: EventG ;
         end ;
       end
       EventD
       Restore system state
       schedule( restart,Timeslot)
      Until i=R-1 ;
      (* start r-th retrial *)
      Event BR
      while (not(Fun_D)) Do
        Begin
          schedule(restart, timeslot)
          case event
            A: EventA ;
            F: EventF ;
            G: EventG ;
          end ;
        end ;
        EventDR (* system need not to be restored here *)
      end ;
  end ;
  Cancel sources
  calculate results
  close files
  MustStop
End.
```

C3.1 Save and restore

For implementing restart the ability to save and restore the system state is necessary. The restart process is always scheduled as the first event in a timeslot. Next figure gives what components are present and what status they have at the beginning of a certain timeslot.

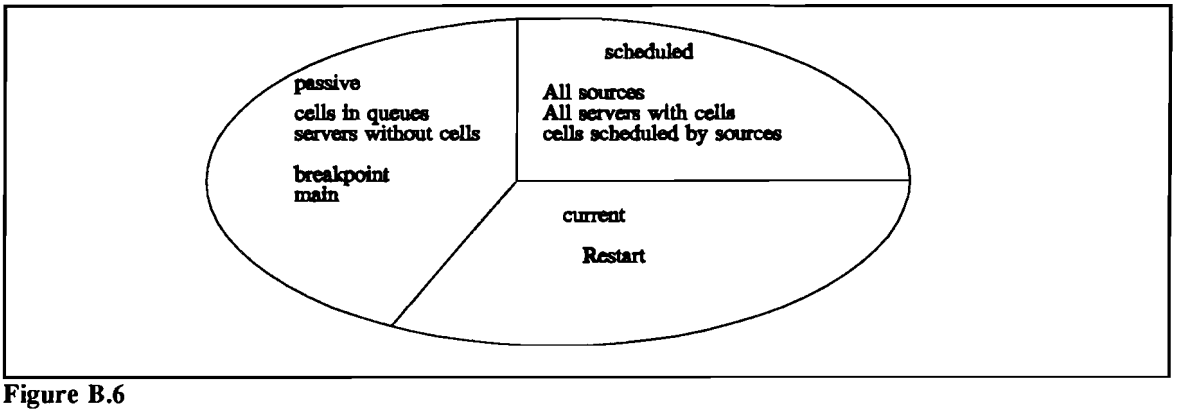


Figure B.6

What components should be saved to be restored?

- All cells in queues + servers
- The states of all sources

The cells in the servers are saved by making a copy of the cell and saving the pointer to that cell. If the server does not contain any cell than the pointer is made nil.
The state of the source depends on the type of sources:

Detsrc : The number of timeslots till it is scheduled.
TwostateMMBP sources: State{On,OFF,LOW,HIGH}
BatchSrc: none, since it is independent from slot to slot.

To save the cells in the queues an array of queues is defined in which copies of the cells are saved. To update the time dependent attributes the time event B occurs is also saved.

For saving the data and to retrieve the data to restore, an object state_space is designed. State_space has the following attributes:

- B_time (* simulation time of event B *)
- srvstate array of cellptr (* contains the pointer to the copied datacomponent*)
- srcstate (* contains the state of the source at event B *)
- queuemat array[1..Nr_servers] of queueptr

With making copies of cells and saving the state of sources virtual methods are used. This way the model dependent part of the algorithm is hidden in the object type definition of the model.

With this the algorithm for saving the system state becomes:

```
Save(state_spacePtr,sourcearray, serverarray)
Begin
  for all sources Do
     Save source state ;
  for all servers Do
  begin
    Copy cell from server if any
    empty cell in state_space if any
    Put pointer to copied cell in state_space
    Empty queue of statespace
    Copy cells in queue to queue of state_space
  end;
end ;
```

When event D occurs the system state that is saved must be restored. All cells in the model must be destroyed first. Then the model must be filled with copied cells. the filling is done by scheduling the components in the right sequence.
First the server is scheduled
Then the cell in server if any is scheduled.
Then the cell in the copied queue are scheduled by a First In First Out discipline.
When this is done for all servers the state of the sources are restored.
The restore algorithm becomes:

```
Restore(state_spacePtr,sourcearray, serverarray)
Begin
  destroy all cells that are not data
  for i:= 1 to Nr_servers do
  Begin
    activate server
    copy and activate cell for server
    copy and activate cells in queue of state_space ;
  end ;
  restore sources
end ;
```

Boolean Functions

Fun_B, Fun_F, Fun_G and Fun_D are boolean functions scanning the system for an event B, F etc.

Proc_V

Procedure proc_V is somewhat different. because this does not have to return a boolean. If event V occurs is not important in the restart algorithm. Only when event V is checked in finishing the last regeneration cycle. therefore an function Fun_V is implemented.

C4. Input output specification

Input

The input parameters are read from an input file. The number of parameters chosen as inputs have been kept to a minimum. Only parameters of sources, servers and the number of retrials are inputs. This is not because other parameters of the model can not be changed. I have chosen for only those inputs that will change given the structure of the model. The structure of model such as the number of servers can be changed also but then the program has to be compiled again. The input file is a text file which can be edited by every ascii editor.

The first line must contain R, the number of retrials. The second line T, and the third line T'. The rest of the input file contains parameters of servers and sources. This analog to the inpput specification of simmust.

C4.1 specification of input file

<Number of retrials R>
<T>
<T'>
<Server 1>
<Server 2>
..
<Source Virtual connection>
<Source background>
<Source background>
..
EOF

<Number of retrials R>,
<T>,
<T'> : Integer
<server>: Server
<buffer size>
<servicetime>
<buffersize>,
<servicetime>: Integer

<number of cells to be simulated> : LongInt

<Source virtual connection>:Source
<nr of cells>
<type>
<attributes>
<Source background>: Source
<type>
<attributes>
<type>: {DetSrc, BatchSrc, TwoStateMMBP, Intbernoulli, On_Off, LongMMBP, Empty}

DetSrc<attributes>
Interval: Integer
BatchSrc<atrributes>
Batchdistribution: string
TwostateMMBP<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
BatchDistribution High: String
BatchDistribution Low: String
Interval between Batches: Integer
IntBernoulli<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
Probability(succes): Real
On_Off<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
Interval : Integer
TwostateMMBP<attributes>
Probability{Low|High}: Real
Probability{High|Low}: Real
BatchDistribution High: String
BatchDistribution Low: String
Interval between Batches: Integer
Empty<attributes>

C4.2 Output specification

Two output consists of two output files
Results
Vp1file

Results
These output specification assumes that event A is a Cell loss. When another type of event A is defined this output will be different.

This file contains for every cell loss:
No batch/burst arrivals involved:
Cell loss:
File with of every cell loss from the VC the triple(time,queuename,sequencenr).

When batches are involved:
Cell loss:
File with for every cell loss from the VC a quadruple (time,queuename,sequence nr,batch/burstsize).
The following results of RESTART are also listed in this file:

P1'		Probability event B
P2'		Expectation of the number of events A conditioned on event B
P		Probability event A
Vp1		Variance P1'
Vp2		Variance P2'
V(p)		Variance P
N2		Number of events A occured
N1		Number of events B occured
N		Number of timeslots not counting the retrials
V1		Variance between primary units
V2	V2i	Variance between the retrials
a		Expectation of the number timeslots with event C in one interval [B,D)
m		Number of regeneration intervals

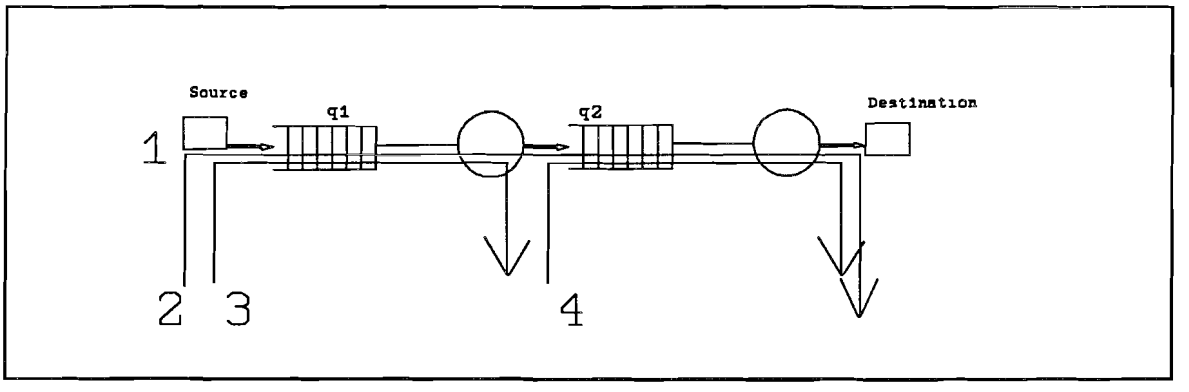
last timeslot simulated

Vp1file

This file contains for each regeneration interval in which events B occured the length and the number of events B occurs.

Appendix D
Model of a VC
and
the results of Restart

D.1 Model on which Restart is applied



Node 1: Service time: 1 timeslot
Buffer length: 50 cells
Node 2: Service time 1 timeslot
Buffer length: 40 cells
source 1 VC source
source 2 LongMMBP,
High state sojourn time: 250 timeslots
Low state sojourn time: 750 timeslots
generation process in the high state:
P(n) probability of generating n cells in one slot
P(0): 0.81873
P(1): 0.16375
P(2): 0.01637
P(3): 0.00109
P(4): 0.00006
No cells are generated in low state
Source3 LongMMBP
High state sojourn time: 143.21 timeslots
Low state sojourn time: 39224.8 timeslots
generation process in the high state:
P(N) probability of generating n cells in one slot
P(0): 0.44659
P(1): 0.35977
P(2): 0.14492
P(3): 0.03891
P(4): 0.00981
Cell generation process Low state
P(0): 0.81996
P(1): 0.16276
P(2): 0.01615
P(3): 0.00107
P(4): 0.00006
Source 4 as source 3

D.2 The four simulated cases

Case 1

Case 1 is the case to which all other cases are compared. This case has the following VC source:

LongMMBPVC source,

Generating no cells in the low state
High state sojourn time: 125 timeslots
Low state sojourn time: 750 timeslots
generation process in the high state:
P(N) probability of generating n cells in one slot
P(0): 0.81873
P(1): 0.16375
P(2): 0.01637
P(3): 0.00109
P(4): 0.00006
No cells are generated in low state

Event B is defined to be TRUE in the timeslot after which the occupance of the queue of the second node exceeds some level T.

Let us define

q2: Number of cells that are queued in Node 2.

Then event B is defined in this case as:

B: q2>T

This case is simulated for two levels of hysteresis:

1.1 T=5, T'=4, R=418

1.2 T=5 T'=3, R=418

The value of R is determined by first simulating the case with R=1000 for 100000 cells. With these results R is calculated by the formula in 5.2.3 assuming $y_o = 1$. This value of R is taken for both simulations.

Case 2

Case 2 results in a higher probability of event A and illustrates the effect on the gain of this higher probability. To achieve a higher probability of event A, the sojourn time in the ON state is doubled. This leads to the following VC source:

LongMMBPVC source,

Generating no cells in the low state
High state sojourn time: 250 timeslots
Low state sojourn time: 750 timeslots
generation process in the high state:
P(N) probability of generating n cells in one slot
P(0): 0.81873
P(1): 0.16375
P(2): 0.01637
P(3): 0.00109
P(4): 0.00006
No cells are generated in low state

The event B is defined on the same manner as in case 1.

fun_b q2>T

This case is simulated once with the same hysteresis level as in case 1.1. The value of R is first determined simulating with R=1000, from those results R is calculated.

2.1 T=5 T'=3 R=236

Case 3 and 4 are simulated to illustrate influence of the choice of event B on the gain. Two choices of events B are simulated

case 3

This case uses the same source as in case 1, only the definition of event B is different. Let us define:

q2: Occupance of the queue of node 2.
srcstate[4]: The state of source 4.

Then event B is defined as:

B: ((q2>T) and (srcstate[4]=High))
or
((q2>T+10) and (srcstate[4]=Low))

The constant 10 is chosen somewhat arbitrary. If q2 exceeds T+10 then cells can also be lost. This situation is not likely to occur but no events A must be excluded.

case 4

In this case the same VC source is used as in case 1 but with a different definition of event B

Let us define:

srcstate[1]: State of the vc source.
node1: Boolean which is true if cells of the VC source are in node 1.

B: ((srcstate[1]=High) or node1) and (q2>T)

4.1 T=4 T'=3 R=418
4.2 T=4 T'=2 R=418
4.3 T=5 T'=4 ,,
4.4 T=5 T'=3 ,,
4.5 T=6 T'=4 ,,

D.3 Results

case1.1

Results
P1'= 2.64517195954637E-0005
P2'= 5.89991725725798E-0003
P^= 1.56062956925425E-0007
vp1: 2.05283389085618E-0012
Vp2: 7.23285252449482E-0007
V(P^): 5.79019895685561E-0016
k1: 3.67268915546010E+0001
k2: 4.98465624775417E+0000
b: 4.27430285616429E+0001
Gain: 8.93971107524684E+0001
N2:1312
N1:532
N: 2.01121140000000E+0007
V1: 3.15185184367017E-0004
V2: 2.90938742332401E-0002, V2i: 1.54779410920892E+0001
a: 2.35302505664295E+0001
l: 2.60329487099953E+0001
number of regeneration cycles;163972
last timeslot 2.58886190000000E+0007
yi2:32478
t_r/t_w= 0.877

case 1.2

Results
P1'= 2.67830097938665E-0005
P2'= 4.54260651629073E-0003
P^= 1.21664674815496E-0007
vp1: 2.13579484244209E-0012
Vp2: 7.57156006507727E-0007
V(P^): 5.88820267190633E-0016
k1: 3.54682643326232E+0001
k2: 5.93334338643763E+0000
b: 6.13664740148815E+0001
Gain: 8.73388485997989E+0001
N2:957
N1:504
N: 1.88179000000000E+0007
V1: 3.17886632787534E-0004
V2: 2.66349576978371E-0002, V2i: 1.34240186797106E+0001
a: 2.36357085894560E+0001
l: 2.90620823234785E+0001
number of regeneration cycles;156431
last timeslot 2.49265210000000E+0007
yi2:29755
t_r/t_w= 0.877

case 2.1
Results
P1'= 3.24323437314472E-0005
P2'= 1.55333522106711E-0002
P^= 5.03783018198120E-0007
vp1: 3.99136658655380E-0012
Vp2: 5.96442850268886E-0006
V(P^): 7.26058862469992E-0015
k1: 4.21421626186348E+0001
k2: 7.24670190295001E+0000
b: 3.07835521357774E+0001
Gain: 5.82216049207830E+0001
N2:1536
N1:419
N: 1.29192020000000E+0007
V1: 2.03072755334333E-0003
V2: 1.10534845470966E-0001, V2i: 4.63141002523480E+0001
a: 2.65056126372074E+0001
l: 3.19558674916334E+0001
number of regeneration cycles;178855
last timeslot 1.60658730000000E+0007
yi2:52908
t_r/t_w= 0.885
case3.1
Results
P1'= 1.67727489573170E-0005
P2'= 1.09551075901378E-0002
P^= 1.83747269409779E-0007
vp1: 1.37830270480050E-0012
Vp2: 2.40890986318368E-0006
V(P^): 8.46423181158169E-0016
k1: 5.43572790247854E+0001
k2: 5.59433677502966E+0000
b: 3.32668171324767E+0001
Gain: 8.89569087504961E+0001
N2:1676
N1:366
N: 2.18211100000000E+0007
V1: 7.36805134011753E-0004
V2: 6.05497561318771E-0002, V2i: 2.21612107442634E+0001
a: 3.03138089257700E+0001
l: 3.55401142783230E+0001
number of regeneration cycles;178538
last timeslot 2.72448950000000E+0007
yi2:54664
t_r/t_w= 0.885

case 4.1 T=4 T'=3
Results
P1'= 2.62064887668332E-0005
P2'= 6.63860301765807E-0003
P^= 1.73974475409721E-0007
vp1: 2.20044317432478E-0012
Vp2: 1.69890596080774E-0006
V(P^): 1.26748891387161E-0015
k1: 1.93465941285831E+0001
k2: 5.63727678824944E+0000
b: 3.58117605756852E+0001
Gain: 9.56780822852537E+0001
N2:1085
N1:391
N: 1.49199690000000E+0007
V1: 5.76120263128921E-0004
V2: 3.68475224346980E-0002, V2i: 1.44073812719580E+0001
a: 1.54431282808800E+0001
l: 1.74848319894518E+0001
number of regeneration cycles;2155635
last timeslot 1.77715320000000E+0007
yi2:42269
t_r/t_w= 0.893
case 4.2 T=4 T'=2
Results
P1'= 2.46079548281549E-0005
P2'= 3.48036966638264E-0003
P^= 8.56447795356246E-0008
vp1: 2.01414794070342E-0012
Vp2: 6.93939818702423E-0007
V(P^): 4.46011274171024E-0016
k1: 1.74514687205956E+0001
k2: 5.49556338101684E+0000
b: 4.49219156464678E+0001
Gain: 1.23176663916310E+0002
N2:531
N1:365
N: 1.48326020000000E+0007
V1: 2.08028311866837E-0004
V2: 1.89185637790956E-0002, V2i: 6.90527577936882E+0000
a: 1.43747001374868E+0001
l: 1.86872909847880E+0001
number of regeneration cycles;2146827
last timeslot 1.76769240000000E+0007
yi2:14003
t_r/t_w= 0.894

case 4.3
Results
P1'= 1.02623866604018E-0005
P2'= 1.06215085548235E-0002
P^= 1.09002027706364E-0007
vp1: 6.94798712832318E-0013
Vp2: 4.33389831491771E-0006
V(P^): 5.37827255489161E-0016
k1: 2.83586763787898E+0001
k2: 5.69422206010495E+0000
b: 2.39881636736682E+0001
Gain: 1.40689059973653E+0002
N2:848
N1:191
N: 1.86116550000000E+0007
V1: 6.84720732235355E-0004
V2: 5.97965075922389E-0002, V2i: 1.14211329501122E+0001
a: 2.25055863120651E+0001
l: 2.91154462677951E+0001
number of regeneration cycles;2702201
last timeslot 2.09309230000000E+0007
yi2:26496
t_r/t_w= 0.934

case 4.4 T=5 T'=3
Results
P1'= 1.15755702550725E-0005
P2'= 1.69938953967992E-0002
P^= 1.96714030073003E-0007
vp1: 8.54180086086817E-0013
Vp2: 9.31081768314190E-0006
V(P^): 1.50222599247498E-0015
k1: 3.19094699431153E+0001
k2: 5.59163867591269E+0000
b: 2.54510329990881E+0001
Gain: 1.00981038323354E+0002
N2:1442
N1:203
N: 1.75369330000000E+0007
V1: 1.66675395906601E-0003
V2: 9.33569687961153E-0002, V2i: 1.89514646656171E+0001
a: 2.46580597258289E+0001
l: 2.90490371182677E+0001
number of regeneration cycles;2543770
last timeslot 1.99964800000000E+0007
yi2:69070
t_r/t_w= 0.934

case 4.5 T=6 T'=4
Results
P1'= 6.12658419065125E-0006
P2'= 2.16382203813469E-0002
P^= 1.32568378902187E-0007
vp1: 3.61684449818338E-0013
Vp2: 1.41331697561808E-0005
V(P^): 7.04945962720269E-0016
k1: 3.70345383319072E+0001
k2: 5.73668541335064E+0000
b: 1.70929588118161E+0001
Gain: 1.41493534790546E+0002
N2:1212
N1:134
N: 2.18718940000000E+0007
V1: 1.60070849196892E-0003
V2: 1.22530954740569E-0001, V2i: 1.64191479352303E+0001
a: 2.86819788615394E+0001
l: 3.31265443058219E+0001
number of regeneration cycles;3169703
last timeslot 2.37231660000000E+0007
yi2:48160
t_r/t_w= 0.926