INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

VirtualClock With Priority Buffer : A Resource Sharing Algorithm

by

Jianmei Chen, B. Eng.

South China University of Technology, P.R.China

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering

Ottawa-Carleton Institute of Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University

Ottawa, Ontario

August 20, 1997

© copyright

1997, Jianmei Chen

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26988-4

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

VirtualClock With Priority Buffer : A Resource Sharing Algorithm

submitted by **Jianmei Chen, B. Eng.**

in partial fulfillment of the requirements for

the degree of Master of Engineering

Chair, Rafik Goubran
Department of Systems and Computer Engineering

Thesis Supervisor, Mihail Devetsikiotis

Carleton University

August 20, 1997

Abstract

Asynchronous Transfer Mode (ATM) has been proposed by the Consultative Committee for International Telegraphy and Telephony (CCITT) as the target information transport mechanism for the broadband network (B-ISDN) of the future. The packet-oriented information transfer using small, fixed size blocks called cells provides any arbitrary bit rate up to the link capacity. Traffic streams are statistically multiplexed so that the resources required to satisfy the Quality of Service (QoS) desired for each stream can be reduced by sharing both bandwidth and buffer memory. Without proper resource management mechanisms, resource sharing can cause disruptions among different bursty traffic streams and make networks fail to meet their commitments on QoS requirements.

In this thesis, we have proposed VirtualClock with Priority Buffer (VCPB) as a new resource management scheme which can meet individual QoS requirements while preserving maximum resource sharing. Furthermore, we have shown that, for VCPB, the performance of various traffic classes in heterogeneous environments can be upper bounded by their performance in homogeneous traffic environments. Therefore traffic descriptors for different traffic classes can be defined based on their performances under homogeneous traffic environments. This has greatly simplified the relationship between the characteristics of sources and network resources and made the effective bandwidth approach more useful.

The algorithm is verified via simulations with a hierarchical traffic model built in OPNET.

Acknowledgments

I would like to express my sincere thanks to my thesis supervisors, Prof. Mihail Devetsikiotis and Prof. A. R. Kaye for helping me achieve a focused and organized approach to research and for their guidance throughout the course of this thesis.

I gratefully acknowledge the funding provided by the Telecommunications Research Institute of Ontario. Also, I wish to thank Prof. I.E. Lambadaris for his valuable guidance and suggestions during the research effort that led to this thesis.

I would also like to extend my thanks to my friend, Dr. Changcheng Huang, for his enduring patience, taking the time to answer my numerous queries during this research. Special thanks are due to my family, especially to my mother for their constant support.

To my dear friend

Table of Contents

Abstract	iii
Acknowledgments.	iv
Table of Contents	vi
List of Figures	ix
List of Tables	xi
List of Acronyms	xiii
 Chapter 1: Introduction	 1
1.1 Background	1
1.2 Thesis Objectives	4
1.3 Thesis Organization	6
 Chapter 2: QoS and Resource Management	 8
2.1 ATM Quality of Service and Service Architecture	8
2.1.1 Quality of Service	8
2.1.2 ATM Service Architecture	10
2.2 Resource Management Mechanism	12
2.2.1 Two Fundamental Issues: Isolation and Sharing	12
2.2.2 Different Bandwidth Enforcement Algorithms	13
2.2.2.1 Round-Robin Schedulers	13
2.2.2.2 Schedulers Based on Virtual Finishing Times	16
2.2.3 VirtualClock Algorithm and Implementation	19

2.2.4	Buffer Sharing Mechanisms	23
Chapter 3:	Traffic Descriptors	27
3.1	Traffic Modelling	27
3.1.1	Mathematical Processes for Modeling	27
3.1.2	Layered Modeling Structure	29
3.1.3	Models for Each Level	30
3.1.3.1	Model for Connection Level	30
3.1.3.2	Models for Burst Level	30
3.1.3.2.1	Belcore Standard ON-OFF Model	31
3.1.3.2.2	QTES-Modelled Video	33
3.1.3.3	Models for Cell Level	35
3.2	Traffic Descriptor: Effective Bandwidth	37
3.2.1	Linearity Property and Subadditivity Property	37
3.2.2	Equivalent Bandwidth for Traditional Markovian Model	38
3.2.3	Equivalent Bandwidth for Video Model	42
Chapter 4:	Proposed VCPB Algorithm and Simulation Scheme	43
4.1	Overview	43
4.2	Proposed VirtualClock with Priority Buffer Algorithm.	44
4.3	Traffic Descriptor	49
4.4	Network Simulation.	51
4.4.1	Overview of Simulation Tools	51
4.4.2	Simulation Structure	52
4.4.3	Simulation Scenarios and Parameter Settings.	54
4.4.3.1	Parameter Settings of Burst Level and Cell Level	55

4.4.3.2	Parameter Settings for Connection Level in Homogeneous Simulations	56
4.4.3.2.1	Verification of Simulation Structure	56
4.4.3.2.1	Prediction of Traffic Performances	58
4.4.3.3	Parameter Settings for Connection Level in Heterogeneous Simulations	60
4.4.3.4	Parameter Settings for Finite Buffer Simulation	61
Chapter 5:	Simulation Results	62
5.1	Results for Homogeneous Simulations	62
5.1.1	Verification of Simulation Models	62
5.1.2	Performance Comparison of Two Traffic Models	64
5.2	Results for Heterogeneous Simulations.	66
5.2.1	Performance of FIFO Queueing discipline (FIFO Scheduling and FCFU Buffer Management)	66
5.2.2	Performance of VirtualClock Scheduling with FCFU Buffer Management	68
5.2.3	Performance of The VCPB Algorithm	70
5.2.3.1	Balanced traffic load case	71
5.2.3.2	Unbalanced traffic load case	73
5.2.3.3	Further Discussion on The VCPB Algorithm.	75
5.2.4	Using Finite Buffer in Simulations	75
Chapter 6:	Conclusions and Recommendations for Further Study	78
6.1	Conclusions	78
6.2	Recommendations for Further Study	81
References:		83

List of Figures

Figure 2.1:	Cell transfer delay probability density model.	9
Figure 2.2:	A Processor sharing node	13
Figure 2.3:	WRR scheduling	14
Figure 2.4:	HRR frames	15
Figure 2.5:	Real time, VirtualClock and packet-processing order	20
Figure 2.6:	A buffer system	24
Figure 3.1:	Hierarchical traffic modeling	29
Figure 3.2:	ON-OFF source state model	31
Figure 3.3:	The cell pattern of ON-OFF model	36
Figure 3.4:	Video ATM cell transmission and corresponding spreading factor. . .	36
Figure 3.5:	Effective bandwidth curve	40
Figure 4.1:	A processor sharing node with VCPB	46
Figure 4.2:	Simulation structure.	52
Figure 4.3:	The analyzed multiplexer configuration (implemented by C++ function code)	53
Figure 4.4:	Network simulator structure in OPNET	53
Figure 4.5:	Homogeneous traffic structure	54
Figure 5.1:	Queue tail distribution comparison of simulation result and analytic result	63
Figure 5.2:	Queue tail distribution comparison of analytical, static and dynamic source (different utilization) results for ON-OFF traffic	63

Figure 5.3:	Queue tail distribution comparison of ON-OFF traffic and QTES-modeled traffic in balanced traffic load case	64
Figure 5.4:	Queue tail distribution comparison of ON-OFF traffic and QTES-modeled traffic in unbalanced traffic load case	65
Figure 5.5:	Queue tail distribution of aggregated traffic compared with homogeneous case in balanced traffic load case	67
Figure 5.6:	Queue tail distribution of aggregated traffic compared with homogeneous Case in unbalanced traffic load case	69
Figure 5.7:	Queue tail distribution for ON-OFF traffic and QTES-modeled video with VCPB algorithm in balanced traffic load case	71
Figure 5.8:	Queue tail distribution for ON-OFF traffic with VCPB algorithm in balanced traffic load case	72
Figure 5.9:	Queue tail distribution for QTES-modeled video traffic with VCPB algorithm in balanced traffic load case	73
Figure 5.10:	Queue tail distribution for ON-OFF traffic and QTES-modeled video with VCPB algorithm in unbalanced traffic load case	74
Figure 5.11:	Queue tail distribution for ON-OFF traffic with VCPB algorithm in unbalanced traffic load case	74
Figure 5.12:	Queue tail distribution for QTES-modeled video traffic with VCPB algorithm in unbalanced traffic load case	75

List of Tables

Table 4.1:	Common broadband to the home services and the required transmission characteristics	45
Table 5.1:	Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) in balanced traffic load case	66
Table 5.2:	Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) in unbalanced traffic load case	66
Table 5.3:	Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) while using FIFO discipline in balanced traffic load case	67
Table 5.4:	Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) while using VirtualClock scheduling with FCFU buffer in balanced traffic load case	68
Table 5.5:	Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) while using VirtualClock Scheduling with FCFU buffer in unbalanced traffic load case	70
Table 5.6:	Comparison of mean delays for ON-OFF traffic with VCPB algorithm in balanced traffic load case	72
Table 5.7:	Comparison of mean delays for video traffic with VCPB algorithm in balanced traffic load case	72
Table 5.8:	Comparison of mean delays for ON-OFF traffic with VCPB algorithm in unbalanced traffic load case	74

Table 5.9:	Comparison of mean delays for QTES-modeled video traffic with VCPB algorithm in unbalanced traffic load case	74
Table 5.10:	Comparison of mean delays under finite and infinite buffer size schemes for ON-OFF traffic with balanced traffic load case	76
Table 5.11:	Comparison of mean delays under finite and infinite buffer size schemes for QTES-modeled video traffic with balanced traffic load case	76
Table 5.12:	Comparison of mean delays under finite and infinite buffer size schemes for ON-OFF traffic with balanced traffic load case	77
Table 5.13:	Comparison of mean delays under finite and infinite buffer size schemes for QTES-modeled video traffic with balanced traffic load case	77

List of Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
AMP	Autoregressive Modulo Process
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BNL	Broadband Networks Laboratory
CAC	Call Admission Control
CBR	Constant Bit Rate
CCITT	Consultative Committee for International Telegraphy and Telephony
CDV	Cell Delay Variation
CLR	Cell Loss Ratio
CLP	Cell Loss Probability
CP	Complete Partitioning
CS	Complete Sharing
CSVP	Complete Sharing with Virtual Partition
CTD	Cell Transfer Delay
FCFU	First Come First Use
FIFO	First In First Out
FQ	Fair Queueing algorithm
GoP	Group of Picture
HRR	Hierarchical Round-Robin
LAN	Local Area Network

LRD	Long Range Dependence
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MMPP	Markov Modulated Poisson Process
MPEG	Motion Picture Experts Group
nrt-VBR	Non-Real-Time Variable Bit Rate
OPNET	OPtimized Network Engineering Tools
PCR	Peak Cell Rate
PGPS	Packetized Generalized Processor Sharing
PS	Partial Sharing
PSN	Processor Sharing Node
QoS	Quality of Service
QTES	Quantized Transform Expand Sample
rt-VBR	Real-Time Variable Bit Rate
SCR	Sustainable Cell Rate
SP	Spreading Parameter
SRD	Short Range Dependence
TDM	Time-Division Multiplexing
TGSP	Traffic Generator Software Package
UBR	Unspecified Bit Rate
UPC	Usage Parameter Control
VCPB	VirtualClock with Priority Buffer Algorithm
VC-VFTs	Virtual Clock Virtual Finishing Times
WAN	Wide Area Network
WFQ	Weighted Fair Queueing algorithm
WRR	Weighted Round-Robin

Chapter 1: Introduction

1.1 Background

Asynchronous Transfer Mode (ATM) is likely to be the principle transfer mode of future wide area networks (WAN) as well as local area broadband communication networks (LAN) due to its inherent flexibility with respect to the integration of services with widely differing bit rate characteristics. ATM's packet-oriented information transfer mode uses small, fixed size blocks called cells which provide any arbitrary bit rate up to the link capacity. In addition, cost-efficient operation of a network can be achieved by statistical multiplexing of variable bit rate traffic streams and the sharing of both bandwidth and buffer memory. On the other hand, the different services within a future Broadband Integrated Services Digital Network (B-ISDN) will demand a certain Quality of Service (QoS). Without proper resource management mechanisms, resource sharing can cause disruptions among different bursty traffic streams causing networks to fail to meet their commitments on QoS requirements [1].

The concept of statistical sharing implies that there are several sources competing for the same resources. Resources should be shared in such a way that clients obtain better service than if there was no sharing and are protected from the potentially negative effects of sharing [11]. To this goal, any resource management method should have two basic objectives: isolation and sharing. Isolation is the more fundamental goal; it provides guaranteed service for well-behaved sources and quarantines misbehaving sources. But sharing, if performed in the context of an encompassing isolation scheme, performs the very different goal of mixing traffic from different sources in a way that is beneficial to all; bursts are multiplexed so that the post facto jitter is smaller for everyone [11].

Bandwidth and buffer are the two types of resources to be statistically shared in networks. The mechanism used for bandwidth management is generally called scheduling. Scheduling

controls the order in which individual cells or packets are serviced (or whether they are serviced at all). Scheduling is the most direct control over how the network serves every user. There are two approaches to bandwidth enforcement algorithms. One approach is based on round-robin scheduling, the other one is based on virtual finishing times. In the first approach, channel bandwidth is divided into equal time frames; each frame has a fixed number of slots, which are assigned to an individual user; and each user can have a different number of slots. There are two versions in this approach: Weighted Round-Robin (WRR) and Hierarchical Round-Robin (HRR) [30]. In the second approach, one of such algorithms was the Fair Queueing algorithm (FQ) introduced in [12]. This algorithm was targeted at an equal share of link capacity. In addition, a weighted version of the Fair Queueing algorithm (WFQ), in which the α_j (bandwidth allocated to each source) does not need to be equal, was also briefly described in [12]. The VirtualClock algorithm, described in [13] involves an very similar underlying packet scheduling algorithm, but it was expressly designed for a context where resources were preapportioned and, thus, had, as a fundamental part of its architecture, the assumption that the shares α_j were arbitrary. In [14], the WFQ algorithm was reintroduced under the name of Packetized Generalized Processor Sharing (PGPS). It has been shown in [14] that, under the worst case scenarios, PGPS can deliver a guaranteed service.

However, scheduling alone does not prevent substantial packet losses [1]. Because network load can change rapidly, cells are likely to be dropped whenever a surge of cells occurs unless adequate buffer space is available to temporarily harbor the cells. Thus, buffer management is an important aspect of resource management and congestion control. Two of the most general schemes for buffer management are shared buffer pool and per-flow allocation [1]. The shared buffer pool scheme is sometimes called First Come First Use (FCFU) buffer management or Completely Sharing (CS) scheme. The per-flow allocation scheme allocates buffer space to each traffic stream. It can be further divided to two schemes: Complete Partitioning (CP) and Partial Sharing (PS) [34]. The former is a totally separated buffer while the second allows some sharing if the buffer is not used by other users.

The relationship between the resource management structure of a network and the desired

QoS of each transported traffic stream is typically defined in terms of a traffic descriptor. The traffic descriptor is attempt to translate in a straightforward manner the traffic characteristics and QoS requirements of a source into network resource (buffer and bandwidth) requirements. They also help the network to accurately predict its ability to maintain a certain level of performance and to make the decision on whether to accept or reject a new connection in order to prevent congestion.

The network manages its resources based on resource allocation. And it uses the supplied traffic descriptor and the required QoS to establish whether sufficient spare resources are available to admit the call through a given route without introducing congestion into the network. The proposed traffic descriptor for a traffic stream is its “effective bandwidth” [1]. It depends on both the statistical nature of the cell stream and the nature of the required QoS constraint. It attempts to translate the QoS requirement of a source into the network resources. Suppose that there exists a collection of sources J , where each of these sources is known to require an effective bandwidth α_j , $j \in J$. And these sources share a common link having capacity C and buffer size B . As in circuit-switched networks, one can easily check if sufficient bandwidth is available by considering whether

$$\sum_{j \in J} \alpha_j \leq C \quad (1.1)$$

A key point to bear in mind is that, when traffic streams are bursty and share a buffered link, Eq. (1.1) has no clear-cut counterpart if the different sources have different QoS requirements. The effective bandwidth of each source may depend on the characteristics and requirements of other sources to be multiplexed as shown in [10]. The interactions among traffic streams within shared resources are typically, not linear in the number of sources nor are they usually decoupled across the different classes of traffic streams.

Traditionally, the effective bandwidth approach is studied under the scenario of the First In First Out (FIFO) queueing system where heterogeneous traffic streams typically suffer similar cell

loss rate or delay and obtain similar QoS. By using newly developed scheduling algorithms, such as VirtualClock algorithm, networks can be extended to support different delays by guaranteeing different α_j . However, as long as FIFO buffering or FCFU buffering is still used, traffic streams will continue to suffer similar cell loss rates. In [1, 10], de Veciana and Kesidis extended the equivalent bandwidth approach to also include individual statistical constraints. Instead of using a shared buffer or FIFO queueing discipline, they proposed segregating statistically identical streams with similar QoS requirements in “logically” separated buffers, thereby eliminating the disruption within the shared buffer while sharing the output capacity via service policies. This approach is similar to per-flow allocation buffer management. While this may simplify the analysis of the queueing system, it also removes the benefit of shared buffer pool and results in under utilization of buffer. Therefore, a more efficient buffer management mechanism which can work with simple effective bandwidth algorithms and provides different QoS commitments should be developed.

1.2 Thesis Objectives

The ATM layer QoS is measured by a set of parameters which includes throughput, cell loss, cell delay, or cell delay variation. Parameters, like cell delay, are closely related to bandwidth management, while others, like cell loss, are related to buffer management. Many mechanisms for resource management proposed in the literature focus on either bandwidth management or buffer management, but not both.

Given that the VirtualClock algorithm can provide bandwidth enforcement, in this thesis, we propose a new algorithm called VirtualClock with Priority Buffer (VCPB) to provide both bandwidth enforcement and buffer management in order to meet the needs of different QoS requirements and to provide efficient resource usage. While the VCPB algorithm utilizes the same bandwidth enforcement algorithm as the VirtualClock algorithm, it uses priority strategy to manage buffer sharing. A key part of the VCPB algorithm is that we assign a higher loss priority

to the traffic class with lower cell loss rate. Therefore, different from other approaches, the VCPB algorithm can accommodate not only different delay requirements but also different cell loss requirements while maintaining maximal resource sharing.

To predict the performance of the VCPB algorithm, we follow the “effective bandwidth” approach which maps directly the performance requirements to resource allocations. Typically, the effective bandwidth of each stream in heterogeneous traffic environment depends not only on its traffic characteristics but also on the traffic characteristics of other sources to be multiplexed. Therefore, it is very difficult to determine the effective bandwidth in a network which supports multiple QoS requirements. Because the VCPB algorithm provides a special isolation mechanism, we can use the effective bandwidths which can be determined in their corresponding homogeneous environments to estimate the performance and allocate the resources. We will show that, by using our proposed algorithm, the effective bandwidth for each source does not depend on the characteristics of the sources to be multiplexed and, therefore, is a more stable and useful traffic descriptor.

For integrated services, our goal is to implement an appropriate QoS for each connection. But provisioning an exact QoS for each individual connection is typically intractable due to the formidable number of connections in a running network. Traffic streams with similar traffic characteristics usually have similar QoS requirements. It is convenient to separate services into classes and apply congestion control and traffic control procedures to each class. This makes congestion control much easier and allows the allocation of network resources to be optimized among a relatively small number of service classes. The VCPB algorithm can be applied to each traffic class instead of individual connections. We will follow the class-based approach in this thesis.

Although analytical approach can be applied to study certain networks, it quickly becomes too complicated and intractable when the source models and the congestion control schemes becomes complex. In this thesis, we will use simulation approach to study the proposed resource sharing algorithm.

Traffic over ATM network is typically bursty at all time scales [15]. While congestion states are typically described in cell level time scales (e.g. cell loss rate), congestion control such as Call Admission Control works on the connection (or session) level time scale. Therefore, different from traditional simulation approaches, which typically only consider either a connection level or a cell level model, we simulate various schemes using a hierarchical traffic model which considers both levels at the same time. Although more time-consuming, this approach gives more accurate results. This is especially important for effective bandwidth studies. In theory, all effective bandwidth results are correct in the asymptotically sense. It is important to gain insight into the difference between the theoretical results and the practical results when buffer size and cell loss are both far from their asymptotic values.

In summary, this thesis:

- Proposes VirtualClock with Priority Buffer as a new resource management scheme which can meet individual QoS requirements while preserving maximum resource sharing.
- Shows that, for VCPB, the performance of various traffic classes in heterogeneous environments can be upper bounded by their performance in homogeneous traffic environments. Therefore traffic descriptors for different traffic classes can be defined based on their performances under homogeneous traffic environments. This has greatly simplified the relationship between the characteristics of sources and network resources and made the effective bandwidth approach more useful.
- Uses a simulation approach to verify the proposed algorithm. In order to integrate different time scales into one simulation, a hierarchical model and simulation structure is built. Also, the large deviation approach for effective bandwidth is evaluated through simulation.

1.3 Thesis Organization

The remaining chapters of this thesis are organized as follows:

Chapter 2: Will identify Quality of Service and service architecture in ATM networks, and will discuss different resource management techniques including bandwidth scheduling and buffer management mechanisms.

Chapter 3: Will describe and comment on various mathematical traffic models, since the simulation approach will be used and traffic modelling is extremely important for simulation studies. A hierarchical traffic model is presented. Models for each hierarchical level will be discussed in detail. And the traffic descriptor effective bandwidth, which is based on a Markov model, ON-OFF model and QTES model, is also discussed.

Chapter 4: The discussion in Chapter 2 will form the basis of a proposal for a new resource sharing algorithm: VirtualClock with Priority Buffer (VCPB) algorithm. The implementation of this new algorithm and the network simulation schemes which are used to verify the new algorithm are also be discussed.

Chapter 5: Will verify the simulation model, and evaluate the large deviation approach for effective bandwidth. Simulation results are presented to verify the proposed algorithm and to form the basis for a comparison to the results of other resource management techniques.

Chapter 6: Will draw conclusions and make recommendations for further study.

Chapter 2: QoS and Resource Management

B-ISDN traffic is statistically heterogeneous (e.g., voice, video, data, and network signaling). This heterogeneity results in varied QoS requirements, ranging from deterministic to statistical bounds on cell loss probability and/or cell delay. Traffic streams are statistically multiplexed so that the resources required to satisfy the QoS demands for each stream can be reduced by sharing. However, resource sharing can cause disruptions among different bursty traffic streams and cause networks to fail to meet their commitments on QoS requirements

In this Chapter, we will first introduce the Quality of Service and service categories defined by the ATM forum. Then we will discuss the different mechanisms of resource management.

2.1 ATM Quality of Service and Service Architecture

This section will introduce the QoS and service categories defined by the ATM forum.

2.1.1 Quality of Service

The ATM layer QoS is measured by a set of parameters characterizing the performance of an ATM layer connection. These QoS parameters quantify end-to-end network performance at the ATM layer. In the ITU-T Recommendation I.356, they are referred to as network performance parameters [6]. There are three QoS parameters identified and negotiated which correspond to a network performance objective:

- Peak-to-peak Cell Delay Variation (peak-to-peak CDV)
- Maximum Cell Transfer Delay (max CTD)
- Cell Loss Ratio (CLR)

The measured Cell Transfer Delay (CTD) is defined as the elapsed time between a cell exit event and the corresponding cell entry event for a particular connection. The CTD between two measurement points is the sum of the total inter-ATM node transmission delay and the total ATM node processing delay between two measurement points. Peak-to-peak CDV and max CTV are two end-to-end delay parameter objectives which can be negotiated between the networks and the users.

In Figure 2.1, the peak-to-peak CDV is the $1 - \alpha$ quantile of the CTD minus fixed CTD that could be experienced by any delivered cell on a connection during the entire connection holding time. The term "peak-to-peak" refers to the difference between the best and worst case of CTD, where the best case is equal to the fixed delay, and the worst case is equal to a value likely to be exceeded with probability no greater than α .

The maximum Cell Transfer Delay (max CTD) specified for a connection is the $1 - \alpha$ quantile of CTD. The CLR at connection request time is used to place an upper bound on α . When a switch accumulates max CTD or CDV, it may choose a smaller α which may have the effect of over estimating the cumulative max CTD or CDV

The Cell Loss Ratio is defined for a connection as:

$$\text{CLR} = \frac{\text{Lost Cells}}{\text{Total Transmitted Cells}}$$

The CLR parameter is the value of the CLR that the network agrees to offer as an objective over the lifetime of the connection.

Figure 2.1: Cell transfer delay probability density model

2.1.2 ATM Service Architecture

Traffic carried over ATM networks can be classified into several service categories. According to the ATM Forum Traffic Management Specification [5] the following are the service categories in B-ISDN systems:

- Constant Bit Rate (CBR)
- Real-Time Variable Bit Rate (rt-VBR)
- Non-Real-Time Variable Bit Rate (nrt-VBR)
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)

The ATM service categories relate to traffic characteristics and QoS requirements of network behavior. Functions such as routing, CAC, and resource allocation are structured differently for each service category. Service categories are distinguished as being either real-time or non-real-time. For real-time traffic, there are two categories, CBR and rt-VBR, which are distinguished by whether the traffic descriptor contains only the Peak Cell Rate (PCR) or both PCR and the Sustainable Cell Rate (SCR) parameters. The three non-real-time categories are nrt-VBR, UBR, and ABR. These three service categories are intended for non-real-time applications. They differ as to the nature of the service guarantees provided by the network and the mechanisms which are implemented in end-systems and networks to realize them. Selection of an appropriate service category is application specific.

The CBR service category is used by connections that request a static amount of bandwidth that is continuously available during the connection lifetime. This amount of bandwidth is characterized by a PCR value. In the CBR capability, the source can emit cells at the PCR at any time and for any duration while the QoS commitments still pertain.

The rt-VBR service category is intended for real-time applications, i.e., those requiring tightly constrained delay and delay variation, as would be appropriate for voice and video applications. rt-VBR connections are characterized in terms of a PCR, SCR, and Maximum Burst

Size (MBS). Sources are expected to transmit at a rate which varies with time. Consequently the source can be described as “bursty”. Cells that are delayed beyond the value specified by max CTD are assumed to be of significantly reduced value to the application. rt-VBR may support statistical multiplexing of real-time sources.

The nrt-VBR service category is intended for non-real-time applications which have bursty traffic characteristics and which are characterized in terms of a PCR, SCR, and MBS. For those cells which are transferred within the traffic contract, the application expects a low cell loss ratio. nrt-VBR service may support statistical multiplexing of connections. No delay bounds are associated with this service category.

The UBR service category is intended for non-real-time applications. i.e., those applications not requiring tightly constrained delay and delay variation. Examples of such applications are traditional computer communications applications, such as file transfer and e-mail. UBR service does not specify traffic related service guarantees. No numerical commitments are made with respect to the CLR experienced by a UBR connection, or as to the CTD experienced by cells on the connection. A network may or may not apply PCR to the CAC and Usage Parameter Control (UPC) functions. In the case where the network does not enforce PCR, the value of PCR is informational only. When PCR is not enforced it is still useful to have PCR negotiated, since this may allow the source to discover the smallest bandwidth limitation along the path of the connection. Congestion control for UBR may be performed at a higher layer on an end-to-end basis.

ABR is an ATM layer service category in which the limiting ATM layer transfer characteristics provided by the network may change subsequent to connection establishment. The ABR service does not require delay bound or the delay variation experienced by a given connection. ABR service is not intended to support real-time-applications. Upon the establishment of an ABR connection, the end-system shall specify to the network both a maximum required bandwidth and a minimum usable bandwidth. These shall be designated as PCR, and the Minimum Cell Rate (MCR) respectively. The MCR may be specified as zero. The

bandwidth available from the network may vary but shall not become less than MCR.

In this thesis, instead of discussing provisioning QoS requirements for each individual connection, we will discuss supporting QoS requirements for each class.

2.2 Resource Management Mechanism

There are two ways in which the network can actively manage its own resources: bandwidth management (or bandwidth enforcement) and buffer management. Different mechanisms will be discussed in this section.

2.2.1 Two Fundamental Issues: Isolation and Sharing

Unlike traditional data networks, future B-ISDN WANs will be required not only to carry a broad range of traffic classes but to do so while providing a guaranteed QoS to some of these traffic classes. That is also the goal of the congestion control in B-ISDN networks. The actual QoS performance matrices of interest are likely to vary from one application to another, but are projected to include such measures as cell loss, delay, and delay jitter guarantees [29].

When different types of traffic and service classes are merged into a single network, distinguishing between the two basic principles of isolation and sharing is both fundamental and novel [11]. The concept of statistical sharing implies that there are indeed several sources using the bandwidth and buffer space: a single source can not share alone. Clearly, the resources available to a particular user will depend on whether other users are also trying to use the network at the same time, but the network must try to isolate the service provided to a user from the details of how other users respond to congestion. This renders the issue of cooperation moot; the service one user obtains should not depend on whether or not all users cooperate. Providing isolation entails enforcing limits on the resources made available to users. If one source injects excessive traffic into the network, this disrupts the service for everyone.

Figure 2.2: A Processor sharing node

2.2.2 Different Bandwidth Enforcement Algorithms

In recent years, two approaches for bandwidth enforcement algorithms have been developed. One approach is based on round-robin scheduling, the other one is based on virtual finishing time [30].

Consider the Processor Sharing Node (PSN) of Figure 2.2 which consists of a bank of synchronized, discrete-time FIFO queues sharing an output link with a service capacity on cell per unit time (c cell/s). Let N be the number of FIFO queues that have bandwidth guarantees.

The n^{th} FIFO queue has a bandwidth allotment of ρ_n cells per unit time where $\sum_{n=1}^N \rho_n \leq 1$, i.e.,

no “overbooking”. Let $a_i^n \in \mathbb{Z}^+$ be the arrival time of c_i^n , the i^{th} cell arriving to FIFO queue n , and assume all FIFO queues of the PSN are empty prior to time zero [30].

2.2.2.1 Round-Robin Schedulers

In round-robin scheduling, the sequence of cell departure epochs is divided into consecutive frames; each user or each traffic stream with a non-zero bandwidth allotment has a reserved number of slots (departure epochs) in each frame. There might be some unreserved departure epochs. If a user can send a packet or cell only by using its own slots, the round-robin scheduler is

Figure 2.3: WRR scheduling

said to be idling, i.e., not work-conserving. It is also known as “time-division multiplexing” (TDM). If the unreserved departure epochs, or departure epochs that are reserved for users or traffic streams that are currently empty, can be used by other users, the round-robin scheduler is said to be non-idling, i.e., work-conserving [30].

Assume that each user or traffic stream has its own queue, the server may visit each queue an equal number of time slots in a frame. This method provides the users with an equal allocation of network resources. In the version called Weighted Round-Robin (WRR), traffic streams can have different number of slots. Consider a frame size of f slots, under WRR, the server visits the n^{th} queue at most $\lceil \rho_n f \rceil$ times in each frame where ρ_n is the n^{th} queue’s bandwidth allotment. That is, in each frame, queue 1 uses the first $\lceil \rho_1 f \rceil$ slots, queue 2 uses the next $\lceil \rho_2 f \rceil$ slots, etc. To prevent overbooking, the following constraint on the bandwidth allotments must always hold:

$$\sum_{n=1}^N \lceil \rho_n f \rceil \leq f \quad (2.1)$$

For example, in Figure 2.3 the frame size is $f = 8$ and the bandwidth allocations are $\rho_1 = 1/10$, $\rho_2 = 1/4$ and $\rho_3 = 1/6$. Thus, under WRR, queue 1 receives $1/8$ cells per unit time, queue 2 receives $1/4$ cells per unit time, and queue 3 receives $1/4$ cells per unit time. In Figure 2.3, the reserved departure epochs are labeled with the queue’s index [30].

The bandwidth granularity of a round-robin scheduler is defined as the minimum amount of bandwidth it can allocate to a particular queue. Clearly, the bandwidth granularity of the WRR is f^{-1} cells per unit time. One can attempt to improve the bandwidth granularity of the WRR by increasing its frame size f . Because a WRR serves all of the queues in a round-robin manner

Figure 2.4: HRR frames

within each frame, increasing f would increase their minimum-bandwidth property parameters. A bandwidth scheduler's minimum-bandwidth property determines how well the scheduler allocates bandwidth to the queues. Another version called Hierarchical Round-Robin (HRR) attempts to allocate bandwidth more "evenly" than WRR and thereby reduce the minimum-bandwidth property parameters [30].

In HRR, there are several levels of frames. Let the number of levels of frames be L and the number of slots in the l^{th} level be f_l . A certain number, $n_l < f_l$, of slots from the l^{th} frame are devoted to levels $l+1, \dots, L$ where $n_L = 0$.

The operation of HRR is as follows. Each queue is assigned to a particular level. First-level queues are served during departure epochs

$$D_1 = \{i \in Z^+ \mid (j-1)f_1 \leq i \leq jf_1 - n_1 - 1 \text{ for some } j \in Z^+\} \quad (2.2)$$

The queue with bandwidth allotment ρ that belongs to level 1 is assigned $\lceil \rho f_1 \rceil$ consecutive slots from each frame of level 1.

Second-level queues are served during departure epochs

$$D_2 = Z^+ \setminus D_1 = \{i \in Z^+ \mid (i \notin D_1)\}. \quad (2.3)$$

Note that D_2 is the union of the indexes of the last n_1 slots (departure epochs) of each level 1 frame. D_2 is divided into consecutive frames each of size f_2 (as Z^+ was divided into consecutive

frames each of size f_1 above). A level 2 queue with bandwidth allotment ρ is assigned $\lceil \rho f_2 (f_1/n_1) \rceil$ consecutive slots from each frame of level 2. In other words, if a level 2 queue is assigned k consecutive slots from each level 2 frame, then it receives a service bandwidth of $(k/f_2) \times n_1/f_1$ cells per unit time.

Let D_3 be the union of the indexes of the last n_2 slots of each level 2 frame and etc.

An example HRR frame structure is illustrated in Figure 2.4. In this figure, the number of levels $L = 3$, the frame sizes are $f_1 = 14$, $f_2 = 9$ and $f_3 = 12$, and the number of slots reserved for subsequent frames are $n_1 = 3$ and $n_2 = 4$ (the slots reserved for subsequent frames are shaded in). Also, a certain number of slots reserved for queues 1, 2 and 3 are labeled with the queue index as shown in Figure 2.4. The service bandwidths that queue 1, 2 and 3 receive are $\rho_1 = 1/14$, $\rho_2 = (2/9) \times (3/14)$ and $\rho_3 = (4/12) \times (4/9) \times (3/14)$, respectively [30].

A queue belonging to level l with k slots reserved in that frame receives a bandwidth of [30]

$$\rho = k \frac{n_1 n_2 \dots n_{l-1}}{f_1 f_2 \dots f_l} \quad (2.4)$$

As discussed above, idling round-robin schedulers, also known as TDM systems, guarantee each user the prescribed transmission rate. It also completely eliminates interference among users, as if there were firewalls protecting individually reserved bandwidths. Users, however, are limited to transmission at a constant bit rate; the system tolerates no variations in the speed of data generation. In addition, capacities are wasted when a slot is reserved for a user that has no data to send at that moment. On the other hand, non-idling round-robin schedulers can also provide sharing while providing isolation. Departure epochs that are reserved for users or traffic streams that are currently empty, can be used by other users.

2.2.2.2 Schedulers Based on Virtual Finishing Times

The other approach uses local time stamps instead of using frames to determine how cells are

to be served. These time stamps are closely related to Virtual finishing times [30].

The definition of Virtual Finishing Times is given as follows [30]:

For a given n , the Virtual Clock Virtual Finishing Times (VC-VFTs) which are based on arrival times $\{a_i^n\}$ and bandwidth allotment ρ_n cells per unit time are defined by the following recursion:

$$F_i^n = \max\{F_{i-1}^n, a_i^n\} + \frac{1}{\rho_n} \quad \text{for } i \geq 1 \quad \text{and} \quad (2.5)$$

$$F_0^n = 0$$

Note that $\lceil F_i^n \rceil$ is the departure time of the i^{th} cell from a queue having arrival process $\{a_i^n\}_{i=1}^{\infty}$ and a constant service rate of exactly ρ_n cells per unit time.

The order in which cells depart the processor sharing node is typically determined by the numerical order of their VFTs. If some head-of-queue cells have the same VFT, then those cells are served in an arbitrary order with respect to each other.

The basic idea of this kind of scheduler was inspired by the TDM system. To make a statistical data flow resemble a TDM channel, we can imagine that arriving packets are spaced out by a constant interval in virtual time so that each packet arrival indicates that a one slot time period has passed.

These kinds of schedulers consider the network as a Flow Network. In the Flow Network model, a flow is a stream of packets that traverse the same route from the source to the destination and that require the same grade of transmission service. They consider a set of flows and a set of clock rates α_j . The clock rate of a flow represents the relative share of the link bandwidth this flow is entitled to; more properly, it represents the proportion of the total link bandwidth which this flow will receive when it is active. Fair queueing (FQ) is an example of a scheduler falling into this category. It is similar to round-robin scheduling, in that it provides all users with an equal

allocation of network resources. Demers et al. [12] have provided some extensive analysis and simulation work on the performance of fair queueing algorithm. This algorithm was targeted at the traditional data service architecture, and so involved no preallocation of resources; α_j is the same for all the flows. In addition, a weighted version of the Fair Queueing algorithm (WFQ), in which the α_j need not all be equal, was also briefly described in [12]. By assigning a clock rate α_j to each flow, the network commits to provide an effective throughput rate no worse than $(\mu\alpha_j)/(\sum_j \alpha_j)$, where the sum in the denominator is over all currently active flows to this flow. The Virtual Clock algorithm, described in [13], involves a similar underlying packet scheduling algorithm, but was expressly designed for a context where resources were preapportioned and thus has as a fundamental part of its architecture the assumption that the shares α_j were arbitrary. Parekh and Gallager, in [14], reintroduce the WFQ algorithm under the name of Packetized Generalized Processor Sharing (PGPS).

The above algorithms are effective for bandwidth isolation. For a particular example, we look at the behavior WFQ or VirtualClock algorithm induces. When there is a backlog in the queue, packets leave the queue at rates proportional to their clock rates. Consider a moment when all sources are transmitting uniformly at their clock rates except for one which emits a burst of packets. The WFQ algorithm would continue to send the packets from the informed sources at their clock rates, so their packets are not queued for any significant time whereas the backlog of packets from the bursty source will take a long time to drain. Thus, a burst causes a sharp increase in the delay seen by that source, and has minimal effects on the delays seen by the other sources. The mean delay will be rather low, assuming the network link is not over-committed, but a burst will induce jitter directly, and mostly, affecting only the source that emitted the burst. We can see WFQ provides for a great degree of isolation, so that sources are protected from other sources's bursts.

Consider what happens when we use the FIFO queueing discipline instead of WFQ. Now, when a burst from one source arrives, it passes through the queue in a clump while subsequent packets from the other sources are temporarily delayed; this latter delay, however, is much smaller

than the delay that the burst source would have received under the WFQ. Thus, the particular source producing the burst is not singled out for increased jitter; all the sources share all the jitter induced by the bursts of all the sources. Remember that when the packets are of uniform size, the total queueing delay in any time period (summed over all flows) is independent of the scheduling algorithm. The FIFO algorithm splits this delay evenly, whereas the WFQ algorithm assigns the delay to the flows that caused the momentary queueing (by sending bursts). The delays are shared as in the FIFO, in what it might be called multiplexing of bursts. Under the same link utilization, FIFO allows a number of sources to aggregate their traffic to obtain a lower overall delay jitter. The FIFO scheme is an effective sharing scheme.

FIFO is not the only interesting sharing method. Another sharing method is priority, which has a very different behavior from FIFO. The goal of FIFO is to let every source in a common class equally share in the jitter. In priority, one class acquires the jitter of higher priority classes, which consequently obtains a much lower jitter. In one direction priority is considered a sharing mechanism, but in the other it is an isolation mechanism, i.e. lower priority traffic can never affect the performance of higher priority traffic [11]. But, priority strategy can not guarantee bandwidth allocation. The higher priority traffic always obtains the bandwidth it required even though it might transmit data faster than the bandwidth which it specified. There is no protection for lower priority traffic. Furthermore, priority strategy can be used as a buffer sharing mechanism. It will be discussed in section 2.2.4

If the traffic is organized into classes of clients with similar service requirements, we find that this reasoning leads to a nested scheme in which the queuing decision is in two steps: the first step is to ensure isolation of classes, and the second is to provide a particular sharing method within each class [11]. That is exactly what is needed in a heterogeneous environment.

2.2.3 VirtualClock Algorithm and Implementation

In our proposed algorithm, we will use the VirtualClock algorithm as a bandwidth mechanism. That is not to say that all switches must implement the VirtualClock algorithm, but

Figure 2.5: Real time, VirtualClock and packet-processing order

rather that they must implement some scheduling algorithm that can enforce a bandwidth allocation policy. We will present a detailed implementation of the VirtualClock algorithm in this section.

The VirtualClock Algorithm is a scheduling algorithm based on Virtual Finishing Times. By assigning to each data flow a VirtualClock, which ticks at the interpacket gap rate (assuming a constant packet size for the moment). The value of the VirtualClock will denote the expected arrival time of the arrived packet. To imitate the transmission ordering of a TDM system, we can transmit packets according to the time stamp values, as if the VirtualClock stamp were the real-time slot number in a TDM system (see Figure 2.5). This provides firewall protection among individual data flows. If a flow transmits according to its specified average rate, its Virtual Clock reading should fluctuate about real clock time.

The VirtualClock algorithm described in [13] is more than a scheduling algorithm. After the flow is set up, the VirtualClock algorithm is designed both to ensure resource usage and to monitor flows. It supports the diverse performance requirements of various applications by enforcing the resources usage according to each individual flow's throughput reservation, while preserving the flexibility of statistical multiplexing of packet-switching networks. And it monitors the average throughput rate of data flows and provides measurement input to other network control functions, as well as providing feedback to flow sources whenever the actual data flow violate the agreement negotiated during the flow set-up.

The actual algorithm described in [13] is as follows. Two parameters are used to describe a statistical data flow are chosen: the average rate (AR) and the average time interval (AI). Each

packet along the path of a flow uses two control variables, a VirtualClock (*VirtualClock*) and an auxiliary VirtualClock (*auxVC*), to monitor and control the flow according to the specified *AR* and *AI* values. More specifically, each packet switch performs the following two basic functions: data forwarding and flow monitoring. Here, *real time* is referred as current time.

Data Forwarding: The switch has one packet-waiting queue in front of each outgoing link; it serves packets in the following way:

- i) Upon receiving the first data packet from $flow_j$, $VirtualClock_j \leftarrow auxVC_j \leftarrow real\ time$.
- ii) Upon receiving each packet from $flow_j$,
 - (a) $auxVC_j \leftarrow \max(real\ time, auxVC_j)$;
 - (b) $VirtualClock_j \leftarrow (VirtualClock_j + Vtick_j)$, and $auxVC_j \leftarrow (auxVC_j + Vtick_j)$;
 If all packets have a constant size, $Vtick_j = 1 / AR_j$. If packets vary in size, the value of AR_j should be computed from individual packet sizes.
- iii) Stamp the packet with the $auxVC_j$ value.
- iv) Insert the packet into its outgoing queue. Packets are queued and served in the order of increasing stamp values.

Flow monitoring: The switch computes a control variable, $AIR_j = AR_j \times AI_j$ at $flow_i$ set-up. Upon receiving every set of AIR_j data packets from $flow_j$, the switch checks the flow in the following way:

- If $(VirtualClock_j - real\ time) > T$, where T is a control threshold, a warning message should be sent to the flow source. Depending on how the flow source reacts, further control actions may be necessary.
- If $(VirtualClock_j < real\ time)$, $VirtualClock_j \leftarrow real\ time$.

At this time, the switch also synchronizes *VirtualClock* and *auxVC* provided doing so does not cause packets from the same flow to be serviced out of order: $auxVC_j \leftarrow VirtualClock_j$ if either the outgoing link for $flow_i$ is idle or the packet being transmitted has a stamp value greater than $auxVC_j$.

Although the basic concept of *VirtualClock* is simple, not all of the individual steps in the above description are obvious. First, *VirtualClock* and *auxVC* provide different functionalities. Both of them are advanced according to the flow's specified average transmission rate. *VirtualClock* plays the role of a "flow meter" driven by packet arrivals. The difference between the *VirtualClock* and the *real time* indicates how closely a running flow is following its specified transmission rate. The *VirtualClock* algorithm monitors each flow by periodically comparing its *VirtualClock* with the *real time* clock, providing feedback to the flow source whenever its actual throughput departs significantly from the reserved rate. That is the function of **Flow monitoring** as mentioned above. There is a full discussion on how to choose the threshold T and $AIR_j = AR_j \times AI_j$ in [13]. The *auxVC* provides firewall protection among flows and priority services. This is actually the Virtual Finishing Times we discussed in section 2.2.2.2. Queueing packets in the order of their stamp values makes packets from different flow maximally interleaved as in a round-robin service system. If a burst of packets from $flow_i$ arrives at a switch shortly after a burst from $flow_j$ and the two flows have the same average rate, then the packets of the two flows will be one-to-one interleaved in the waiting queue; if $flow_i$ is twice as fast as $flow_j$, then the packets will be two-to-one interleaved in the queue. Serving packets in the order of their stamp values assures that each flow will receive any resources that it has reserved. Therefore, in a network controlled by *auxVCs*, although an aggressive flow can consume idle resources, it cannot disturb network service to other flows. Priority service can be easily accommodated by the *VirtualClock* algorithm. A network can provide priority services to a flow simply by letting each switch replace "*real time*" by "*real time* - P " in the previous *VirtualClock* algorithm description, where P is a chosen value representing the priority. Note that in the **Data Forwarding** function, step (ii) (a), we replace $auxVC_j$ by the maximum value of *real time* and $auxVC_j$. When *real time* >

$auxVC_j$, it implies that a packet arrives later than the expected time, we stamp the packet with *real time* and advance the $auxVC_j$ based on real time, it makes sure that the flow obtains no credit because of its packet's coming late. The resource reserved for the flow has gone. If $real\ time < auxVC_j$, then the packet is stamped by $auxVC_j$. The farther the stamped time is beyond real time, the farther the packet will be placed from the head of the server queue, and the worse service the flow will receive.

L.Zhang [13] has done extensive analysis and simulation work on the performance of VirtualClock, and showed that the VirtualClock algorithm can provide guarantee bandwidth usage.

2.2.4 Buffer Sharing Mechanisms

Buffer allocation to provide an efficient and fair use of the available buffer spaces is of critical importance to ATM networks. Just as bandwidth is closely related to cell delay, buffer size is related to cell loss ratio. While there are many bandwidth sharing mechanisms, there are very few buffer sharing mechanisms.

There are basically three types of buffer allocation schemes: Complete Partitioning (CP), Complete Sharing (CS), and Partial Sharing (PS). It has been shown that, e.g., [36], under relatively balanced input condition, the CS scheme can achieve a lower blocking probability than the CP scheme. When the inputs are unbalanced, however, the buffer space may not be efficiently used by the users. The PS scheme provides a good trade-off between buffer utilization and blocking (loss) probabilities among the users [34].

We are going to compare these schemes by using Figure 2.6. In practice, cells from the same traffic class or connection are not necessarily stored in the memory in order. They can be anywhere inside the memory. Lists of pointers are maintained to indicate the positions of cells in memory and to form a set of virtual queues or logical queues. Assuming that there are J kinds of traffic classes or connection, the available buffer space for traffic class j is B_j , and total buffer size

is B .

For the CP scheme, B_j is pre-assigned for each traffic class, where $\sum_{j=1}^J B_j = B$. It guarantees buffer usage up to this value. If traffic class j has occupied the buffer space up to this value, the next arrival cell from traffic class j will be lost even though there is still unused buffer space.

In contrast, in the CS scheme, there is no pre-assigned value for the available buffer space B_j . B_j can be any value between 0 and B , which represents the unused buffer space. This means that cells from any traffic class can access the buffer as long as the buffer is not full. However, the drawback of CS scheme is that, the buffer space can be occupied by cells from any connection or traffic class, cells might be lost even though the connection to which the cells are attached has occupied few buffer space.

In the PS scheme, B_j includes two parts: a pre-assigned value B_{jmin} , and a dynamic value B_{ja} which is decided by $B_{ja} = B - \sum_{k=1}^J \max(B_{kmin}, B_{kcurrent})$ (where $B_{kcurrent}$ is the current queue size of traffic class k). Traffic class j will be guaranteed to have a minimum buffer usage of B_{jmin} . Cells from other traffic classes cannot access this buffer space even though the buffer space is not used by traffic class j .

Figure 2.6: A buffer system

Other than these three basic types of schemes, the priority strategy which we use for bandwidth allocation can be used as a method of buffer sharing. It can be considered as a modified CS scheme. The priority strategy divides traffic streams into classes and gives each class a priority level. The higher priority traffic has more privileged to access to the buffer than the lower priority traffic and, therefore, is better protected within a shared buffer. Unfortunately, the priority strategy is traditionally used both as a buffer sharing mechanism and as a bandwidth sharing mechanism at the same time [11]. Therefore, the higher priority traffic always obtains more bandwidth than the lower priority traffic. This makes it difficult to apply the priority strategy to real applications. For example, a VBR video stream may be more tolerant of cell losses than data traffic, therefore, it can be treated as low priority traffic in terms of buffer occupancy. But, on the other hand, video stream can be quite sensitive to delay, and requires a larger bandwidth guarantee than data traffic. The low priority traffic cannot obtain this kind of guarantee with a traditional priority strategy.

In [33], Elwalid and Mitra used a fixed priority and buffer control mechanism, with two types of priority: H and L, having two threshold values: A and B, with $A < B$. When the buffer reaches threshold B, only high priority cells are accepted. The low priority cells are not allowed to enter the buffer. The low priority cells are allowed to enter the buffer only when the buffer occupancy drops below threshold A. This scheme, thus, sacrifices L cells to protect H cells.

In [34], a Complete Sharing with Virtual Partition (CSVP) strategy for buffer management at a multiplexer or an output port of an output buffered switch is proposed and analyzed. The total buffer space is partitioned based on the relative traffic loads (measured or estimated). Virtual partitions allow a newly arriving cell belonging to an oversubscribed type to occupy the spare space of an undersubscribed type, and to be overwritten when necessary. In this scheme, priority is not fixed but dynamical. Each type is guaranteed a minimum usage. When the buffer becomes full, the type that over-subscribes will have to sustain a loss of its excess cells.

Associated with the CSVP scheme is a policy to admit cells into the buffer. The CSVP scheme has the following attributes:

- Let there be M users. Each user generates a stream of cells that will be stored in the buffer before being transmitted. The cells generated from user i are referred to as type i cells.
- Assuming that the total available buffer space is B . The buffer space B is virtually partitioned into M segments corresponding to M users, with the size: $B_1, B_2, \dots, B_M$, such that

$$\sum_{i=1}^M B_i = B.$$
- The entire buffer is shared by the M types of cells. When the buffer is not full, cells of any type are accepted upon arrival.
- When the buffer is full, if one type, say the i^{th} , occupies fewer spaces than B_i , then, at least one other type, say the j^{th} , must be occupying more than B_j spaces. The admission policy will admit a newly arriving type i cell by pushing out type j cell from B_i , but will reject a newly arriving type j cell.

A salient feature of the CSVP scheme is that it behaves as a complete sharing mechanism, but maintains fair allocation to participating users. Fair sharing is attained by means of a push-out mechanism [34]. It guarantees each user has its minimum buffer space usage. However, the drawback of this scheme is obvious: the allocation of buffer space B_i for each user is very complicated. It depends on how many users there are inside the system and their traffic characteristics. The joint state probability distribution of the aggregate traffic is governed by a set of partial differential equations, which is difficult, or perhaps, impossible to solve. It is not practical to use this scheme when the number of users is large. Even though we can divide the traffic into several service categories or classes, the joint state probability distribution of the aggregate traffic is still difficult to solve.

Chapter 3: Traffic Descriptors

3.1 Traffic Modelling

New generations of high capacity networks and switching systems based on the ATM principle are on their way both in the public and private domain. The capacity on-demand concept underlying ATM poses a wide range of challenges with respect to traffic engineering. The characteristics of the load put on the systems are partly unknown; new call handling and switching techniques are introduced and very strict requirements are put on QoS parameters like cell loss probability and cell delay. It is impossible to guarantee a high quality of service if we do not know what the traffic will be. Characterizing and modeling various sources (real time video, LAN to LAN data, etc.) via experiments and analytical models have been a major research area for many years. It is well-known that different types of traffic have to be captured by different models. Establishing suitable models for multimedia ATM networks is a crucial part of performance analysis and network control.

3.1.1 Mathematical Processes for Modeling

Commonly used traffic models are surveyed in detail in [20]. Such models can be used either as part of an analytical model, or to drive discrete-event simulations. Of particular interest to modelers are the interarrival times, batch sizes, and work-load processes, in either continuous or discrete (slotted) time. Models outlined in [20] include renewal models (e.g., Poisson processes, Bernoulli processes, and phase-type renewal processes), Markov and Markov-renewal models, Markov-modulated models (e.g., Markov-modulated Poisson processes, transition-modulated processes), fluid traffic models, and autoregressive models. In this section, we briefly describe the nature of several processes which will be used in our simulation. A more detailed discussion of various traffic models can be found in [20].

Poisson Process: This is the oldest process used in traffic modeling. It has only one parameter (mean arrival rate) and all interarrival times are i.i.d. (identical, independently distributed) with exponential distribution. It is very easy to use and has elegant analytical properties. Its dimensionality is excellent. Its drawback is that it is not flexible enough for modeling single traffic. Its usage is based on the theory that, under suitable, but mild, conditions, an aggregating of a large number of i.i.d. processes with reducing rate tends to get a Poisson process. This attribute can be used in modeling backbone traffic. Although a Poisson process is difficult to use to model single traffic, it can be integrated with other processes to model single traffic (e.g. MMPP: Markov modulated Poisson process).

Markov Process: A Markov process is more general than a Poisson process in that, its interarrival times are correlated. Although they may have different parameters, the interarrival times are still exponentially distributed. Due to the memoryless nature of exponential distribution, the Markov process has very good analytical properties and is widely known. But to build a Markov process requires knowledge of the transition matrix which is difficult to get when too many states are assumed. Therefore, to increase its flexibility and reduce its complexity, a Markov process with few states is integrated with other processes to form a new process (e.g. the MMPP, fluid model).

TES Process: TES processes provide another modeling approach geared towards simultaneously capturing both marginal and autocorrelation of stationary empirical records. It is very flexible and very easy to generate. The drawbacks of a TES process are that it can not generate Long Range Dependent (LRD) structure and it has a poor analytical nature.

Fluid Process: The fluid traffic process dispenses with individual traffic units. Instead, it views traffic as a stream of fluid, characterized by a flow rate (such as bits per second). A fluid model is appropriate for the continuative cells. Typical fluid models assume that traffic arrives deterministically at a constant rate. Modeling the larger units as discrete traffic and their transport as fluid flow would give us the best of both worlds: we can measure waiting times and loss probabilities and enjoy savings on simulation computing resources.

3.1.2 Layered Modeling Structure

Traffic over ATM network is typical bursty on all time scales [15]. Due to the diversity and complexity of the activities of ATM sources, it is very difficult to model real traffic on all time scales using a single process as mentioned in the last section. Typically, we prefer to split activities into hierarchical layers and build composite models.

Hui [7] was the first to introduce the notion of time-scale. A typical scale division scheme is as follows: (see Figure 3.1.) [35]

- Connection level: describing connecting and disconnecting of the virtual circuit.
- Dialogue level: describing the alternations between the A and B party in a connection.
- Burst level: describing the behavior of the active party.
- Cell level: describing the actual cell generation.

In the timescale, connections last for a “long” period of time (perhaps measured in seconds or minutes). Looking more closely, perhaps on a millisecond time-scale, it can be seen that a call is made up of bursts of activity interspersed with periods of inactivity. Moving in closer still, to a microsecond or cell level time scale, each burst will comprise a number of cells, each sent at the line rate, with intervening gaps.

For the dialogue and burst levels, different applications have different demands views. For example, interactive applications are more interested in the dialogue level time scale. It is widely believed that the dialogue level can be accurately modeled by ON-OFF models although the processes that model the ON and OFF activities may be quite different. For some applications,

Figure 3.1: Hierarchical traffic modeling

such as video transmission, it is not necessary to consider this level. Dialogue levels will not be used in this thesis as interactive applications are not considered.

3.1.3 Models for Each Level

In this section, models for each level are described.

3.1.3.1 Model for Connection Level

The connection level describes the series of connects and disconnects of the virtual circuit. Typical time constants are in the order of 100 seconds. This level is related to the command function associated with the physical source: when to communicate (the time when a call is established) and where to communicate (call destination). At the call or connection level, it is the “grade of service”: the fraction of calls that are denied access because of lack of network resource. There are two parameters for this level of model: call-arrival process and, statistics of call-holding time.

The connection level is modeled by a birth-death process with Poisson call arrival pattern and exponentially distributed holding time. Poisson process are fairly common in traffic applications that physically comprise a large number of independent traffic streams, each of which may be quite general.

3.1.3.2 Models for Burst Level

Burst level describes the behavior of the active party. The activities on this level are primarily governed by the physical source. We usually distinguish source type by its characteristics in this level, such as voice, data file, and video. Time constants are in the order of 100 milli-seconds.

Two models are used at this level in this thesis: the Variable Bit Rate ON-OFF model [16] and the QTES model [17]. These two models basically use the Markov processes. The Markov process is more general than Poisson process in that, its interarrival times are correlated.

Traffic Generator Software Package (TGSP), which was developed in our lab, is used in this thesis. The TGSP generates synthetic traffic streams at the burst level and cell level. Hence, we will describe the models for burst level and cell level based on the implementation in TGSP.

3.1.3.2.1 Bellcore Standard ON-OFF Model

The ON-OFF model was chosen based on the following considerations: (1) Most of the analytical results in literature are based on the ON-OFF model, and so by using the same burst level model, our simulation results are comparable to the analytical results; (2) The ON-OFF model has some typical features which exist in various models: they are bursty in nature and their burstiness are easy to set; (3) The model is easy to generate and widely used in various simulation studies.

Variable Bit Rate ON-OFF source models produce variable bit rate traffic according to the ON-OFF source models proposed by the Bellcore standards [16]. The inputs that can be specified by the user are:

- Peak rate: In number of cells per second
- Mean rate: In number of cells per second
- Mean burst size: In number of cells
- Line rate: In number of cells per second

Given the line rate we can determine the time needed for the transmission of an ATM cell i.e. the duration of the time slot. The user specified peak and mean rate along with the burst length

Figure 3.2: ON-OFF source state model

then uniquely define an ON-OFF model which is actually a two-phase Markov process consisting of an active phase and a silent phase. As shown in Figure 3.2, the two states are named ON and OFF. The source generates information carrying cells during the ON period and emit no cell during the OFF period. For a given specification, the duration of the active phases of all sources is identical, independently distributed (i.i.d) random variables. The duration of the silent phases of all sources are also i.i.d random variables. The duration of an active phase has an integer number of cell times with a geometric distribution with a mean Average_ON_time:

$$\text{Average_ON_time} = \frac{\text{mean burst size}}{\text{peak rate}} \times \text{line rate} \quad (\text{measured in time slots}) \quad (3.1)$$

The silent phase lasts an integer number of cell times, which is geometrically distributed with mean:

$$\text{Average_OFF_time} = \frac{\text{mean burst size}}{\text{mean rate}} \times \text{line rate} - \text{average_ON_time} \quad (\text{measured in time slot}) \quad (3.2)$$

All the results from divisions in the formulas above are assumed to be appropriately truncated so that the final results are integer valued [16].

According to the theory of geometrical distribution, if the probability in the ON and the OFF states is P_{on} and P_{off} respectfully, then

$$\text{Average_ON_time} = \frac{P_{on}}{1 - P_{on}} \quad (3.3)$$

$$\text{Average_OFF_time} = \frac{P_{off}}{1 - P_{off}} \quad (3.4)$$

Once we obtain the value of P_{on} and P_{off} , we can calculate the ON and OFF periods as follows:

The probability that the ON period has less than n slots is given as:

$$P_{n-on} = \sum_{i=0}^n P_{on}^i P_{off} \quad (3.5)$$

$$P_{n-on} = P_{off} \left(\frac{1 - P_{on}^{n+1}}{1 - P_{on}} \right) \quad (3.6)$$

$$P_{n-on} = 1 - P_{on}^{n+1} \quad (3.7)$$

P_{n-on} is a uniform random number corresponding to n . So Eq. (3.7) can be rewritten as:

$$n = \frac{\log R}{\log P_{on}} - 1 \quad (3.8)$$

The same steps can be applied to the OFF period:

$$n = \frac{\log R}{\log P_{off}} - 1 \quad (3.9)$$

The cell pattern inside each ON period will be discussed in Section 3.1.3.3

3.1.3.2.2 QTES-Modelled Video

While ON-OFF models are quite general, they lack the degree of realism required by some realistic applications. Of particular interest is video traffic which is destined to become the main component of the future B-ISDN network traffic. We can choose a video model which is based on an Autoregressive Modulo Process (AMP) process which can closely match the statistics of real video as shown in [19].

Any empirical traffic trace can be viewed as a stochastic time series. Therefore a correlated (i.e., bursty) time series (such as the number of ATM cells arriving at successive time intervals, or the size of successive LAN packets etc.) can be statistically modeled by an AMP (also known in this particular instance as Quantized Transform Expand Sample or QTES process) in the following sense [17]:

- The marginal probability density of the empirical time series and the modeling process are the same;
- The autocorrelation of the empirical time series and the modeling process are closely matched.

A QTES process is very flexible and very easy to generate. The drawbacks of the QTES process are that it can not generate LRD structure and it has a poor analytical nature.

The generation of an AMP process involves two parts: the generation of a background process with uniform marginal probability distribution in $[0,1]$ and a distortion stage that generates an appropriate foreground process [17]. More details about the generation of a QTES process can be found in [17, 18].

In TGSP, the traffic modeling procedure was initiated for a MPEG VBR video by first collecting data from different video clips. For this purpose we identified three clips namely, a fifteen minute *news segment*, a two-hour *action movie*, and a fifteen minute segment of *American football*. The news segment consists mainly of talking heads and no commercials. It exhibits relatively low activity and therefore can be used as a model of a newscast or teleconferencing session. The action movie and the American football segments exhibit higher scene activity (corresponding to higher burstiness) and in particular the frame sizes of the football segment exhibit the highest autocorrelation.

Using facilities at the Broadband Networks Laboratory (BNL) at Carleton University, the three video clips were encoded multiple times by employing a software MPEG encoder and using Group of Picture (GoP) sizes of 12 and 6 and two settings of the *quantization parameters* Q namely 3 (corresponding to high video quality) and 10 (corresponding to lower but acceptable video quality). Therefore, for each video clip we created four empirical traces each one corresponding to a different GoP size and visual quality.

For each empirical trace we concentrated on the sizes (in Kbytes) of the three different video frames that constitute an MPEG compressed video stream, namely I, B and P frames. Marginal histograms and autocorrelations of the frame sizes for each frame type were computed and an autoregressive modulo process (QTES) was developed for each I, B and P frame that closely matched the empirical marginal distributions and autocorrelations. The software modules that generate the frame size information can be summered in the order prescribed by the MPEG compression standard and appropriate correlation between the information content of different

frames can be introduced in a manner similar to [19].

The computational requirements of the procedure for synthetic MPEG video frame size generation are modest, since each frame module (be it I, B or P) requires only the generation of a random number from the innovation distribution f_v , and an inversion according to the desired marginal distribution.

The MPEG video traffic generator was integrated in the traffic generation software. It accepts as input the following parameters:

- Line rate: Specified as a number in cells per second.
- AAL type: AAL2, AAL3/4 or AAL5.
- Video quality factor: High quality (Q=3) or lower quality (Q=10).
- Group of Pictures (GoP) size: GoP=6 or 12.
- Spreading parameter (SP): It assumes values between 0.0 and 1.0 and it is directly related to the burstiness of the video output. Its usage is explained in the following paragraph:

Video frames are generated periodically every 33ms. Given the size of a particular frame (which is a random number generated by the software), and the type of AAL assumed, the frame can be parsed into a number of ATM cells which are referred as time slots. Furthermore, the pattern the ATM cells assume in the time interval of 33ms (which is the time between the generation of successive video frames) is left for further discussion in Section 3.1.3.3.

3.1.3.3 Models for Cell Level

Cell generation patterns model the cell stream inside each burst period. Normally, the peak rate of a physical source cannot exceed the line capacity. This implies that the total number of cells which can be transmitted in each burst period is less than the total number of time slots if the active burst period is divided into time slots for a given line rate. For example, cells can be

Figure 3.3: The cell pattern of ON-OFF model

transmitted in each time slot from the beginning of the burst period until all available cells have been exhausted, leaving the rest of the burst period idle; or the intervals between cells can be constant and leave no idle period in each active burst period; or the intervals between cells can be varied according to some process. The first method will result in a very bursty traffic stream while the second method result in a smooth traffic stream.

In this thesis, the cell level traffic is modeled by a fluid process which is not based on individual traffic units, rather, it views traffic as a stream of fluid, characterized by a flow rate (such as bits per second) [20]. A fluid model is appropriate for the constituent cells. Typical fluid models assume that traffic arrives deterministically at a constant rate. Modeling the larger units as discrete traffic and their transport as fluid flow would give the best of both worlds: we can measure waiting times and loss probabilities and enjoy savings on simulation computing resources.

In the Bellcore Standard ON-OFF model, shown in Figure 3.3, no cell will be generated in the

Figure 3.4: Video ATM cell transmission and corresponding spreading factor

OFF state. During the ON state the generator generates one ATM cell every $\frac{\text{line rate}}{\text{peak rate}}$ time slots.

In the QTES model, for a given line rate, the time interval of 33ms (which is the time between the generation of successive video frames) can be divided into a number of time slots as shown in Figure 3.4. The spreading parameter (SP) specifies the percentage of the time slots between successive frame generations that are allocated for transmission of a particular frame. If the number of slots happen to be less than the total number of cells in a frame then cell transmission automatically continues using the remaining slots until the generation of the next video frame (in this case the number of slots between successive ATM cells is the smallest, i.e., equal to 1, resulting in very bursty traffic).

3.2 Traffic Descriptor: Effective Bandwidth

3.2.1 Linearity Property and Subadditivity Property

ATM-based B-ISDN networks with heterogeneous applications requiring stringent performance guarantees will need appropriate service provisioning schemes including: /buffer/ bandwidth allocation. The relationship between the resource management structure of a network and the desired QoS of each transported traffic stream typically manifests itself in traffic descriptor. For traffic which tolerates queueing delays and some cell loss, “effective bandwidth” is a recommended traffic descriptor[1]. Effective bandwidth tries to translate the QoS requirement of a source into network resources requirements (buffer and bandwidth) of a connection in a straightforward manner.

It is known that for many Markovian sources, the effective bandwidth of a collection of sources is no more than the sum of their effective bandwidths (subadditivity property) and that, in certain asymptotic regimes, the two are nearly equal (near-linearity property) [8, 9]. When such subadditivity and near-linearity properties hold, the resource allocation and congestion control (by using CAC) significantly simplifies. In particular, a new call is accepted if, and only if, the

sum of the effective bandwidths of existing calls and new call (easy to calculate individually) does not exceed the link speed [21]. For example, for a collection of sources J , each source is known to require an effective bandwidth α_j , $j \in J$. All sources share a link with capacity C and buffer size of B . As in a circuit-switched network, one can easily check if bandwidth is available by considering whether

$$\sum_{j \in J} \alpha_j \leq C \quad (3.10)$$

Such a scheme, when extended to a computer network, would resemble traditional telephone systems where connections are set up if physical resources are available to link the source to the destination. Standard techniques used in telephony could then be adapted to packet-switched networks. When traffic streams are bursty and share a buffered link, Eq. (3.10) has no clear-cut counterpart. The interactions among traffic streams within shared resources is typically not linear in the number of sources nor is it usually decoupled across the different classes of streams.

3.2.2 Equivalent Bandwidth for Traditional Markovian Model

Different approaches to the estimation of the effective bandwidth may be found in [1, 7, 8, 9, 22, 23, 24, 25, 26, 27]. Elwalid and Mitra [22] show that the effective bandwidth of a Markovian source is the maximal real eigen value of a matrix derived from the source parameters, network resources and service requirements. Gibbens and Hunt [25] consider theoretical and numerically heterogeneous ON-OFF fluid sources that alternate between exponentially distributed periods of transmission at a peak rate and silence. Kelly [27], finds effective bandwidth for GI/G/1 queues and for M/G/1 queues with the performance criterion taken to be the buffer utilization or the mean workload. Kesidis [9], takes a large deviation approach in order to determine the effective bandwidth.

There exist several approaches, that typically, give similar or identical results for calculating effective bandwidth. The large deviation approach [1, 9] will be introduced here.

Consider a buffered link with capacity C (cell/second), supporting a stationary and ergodic arrival packet stream $A(t)$. Let X be distributed as the buffer's stationary workload. Our preliminary QoS aim is to ensure that cell loss probabilities at this link are in fact quite small. In order to do so, we require that

$$P\{X > B\} \leq \varepsilon := e^{-B\delta} \ll 1 \quad (3.11)$$

for a reasonably large ATM buffer size B if loss is of concern.

The parameter δ clearly determines the stringency of the QoS constraint.

Under mild conditions, for both continuous and discrete-time arrival processes, it has been shown, that for all $\delta \geq 0$

$$\alpha(\delta) < c \Leftrightarrow \lim_{B \rightarrow \infty} B^{-1} \log P\{X > B\} \leq -\delta \quad (3.12)$$

or equivalently

$$P\{X > B\} = \exp[-B\alpha(c) + o(B)] \quad (3.13)$$

where $o(B)$ is a function satisfying $\lim_{B \rightarrow \infty} (o(B))/B = 0$.

The function $\alpha(\cdot)$, shown in Figure 3.5, is given by

$$\alpha(\delta) := \delta^{-1} \lim_{t \rightarrow \infty} t^{-1} \log E \exp[\delta A(0, t)] \quad (3.14)$$

where $A(0, t]$ denotes the number of cell arriving at the buffer in the interval of time $(0, t]$. In light of Eq. (3.12), $\alpha(\cdot)$ is called the source's effective bandwidth. The effective bandwidth is a non-decreasing function, in δ , with the mean rate of the source being $\alpha(0)$ and the "peak" rate being $\alpha(\infty)$.

In [9], the formula for a two-state discrete-time Markov source is obtained. We call a buffer source a discrete-time Markov source if there is a discrete-time Markov chain Z and a real constant R such that the number of arrivals to the buffer in the interval of (continuous) time $(nR^{-1}, (n+1)R^{-1})$ is a function of Z_n . We take the state space of Z to be $1, 2, \dots, m$, and we let Q

Figure 3.5: Effective bandwidth curve

be its irreducible and periodic transition probability matrix. Let Λ_i be the number of cells that arrive in the interval $(nR^{-1}, (n+1)R^{-1})$ when $Z_n = i$. We assume $0 \leq \Lambda_i \leq \Lambda_{i+1} \leq \infty$ for all $i = 1, \dots, m-1$.

Therefore,

$$\alpha(\delta) = \delta^{-1} R \log \left[\rho \left(e^{\delta \Lambda} Q \right) \right] \quad (3.15)$$

For a Two-State Discrete-Time Markov Source:

$$\alpha(\delta) = \delta^{-1} R \log \left[\frac{1}{2} \left(a(\delta) + \sqrt{a(\delta)^2 + 4b(\delta)} \right) \right] \quad (3.16)$$

where

$$a(\delta) = Q_{1,1} e^{\delta \Lambda_1} + Q_{2,2} e^{\delta \Lambda_2}$$

$$b(\delta) = e^{\delta (\Lambda_1 + \Lambda_2)} (1 - (Q_{1,1} + Q_{2,2}))$$

$$Q = \begin{bmatrix} Q_{1,1} & Q_{1,2} \\ Q_{2,1} & Q_{2,2} \end{bmatrix}$$

For a given ON-OFF model which was described in Section 3.1.3.2.1, effective bandwidths exist and can be calculated by Eq. (3.11) and Eq. (3.16).

Recall that the traffic descriptors of the ON-OFF model are:

- Peak rate: In number of cells per second
- Mean rate: In number of cells per second
- Mean burst size: In number of cells
- Line rate: In number of cells per second

And we can calculate:

$$\text{Average_ON_time} = \frac{\text{mean burst size}}{\text{peak rate}} \times \text{line rate} \quad (\text{measured in time slot}) \quad (3.17)$$

$$\text{Average_OFF_time} = \frac{\text{mean burst size}}{\text{mean rate}} \times \text{line rate} - \text{average_ON_time} \quad (\text{measured in time slot}) \quad (3.18)$$

Because of the geometric distribution, we can obtain the probability of the ON and the OFF as the follows:

$$\text{Prob_ON_ON} = \frac{\text{Average_ON_time}}{1 + \text{Average_ON_time}} \quad (3.19)$$

$$\text{Prob_OFF_OFF} = \frac{\text{Average_OFF_time}}{1 + \text{Average_OFF_time}} \quad (3.20)$$

In Eq. (3.16)

$$R = \frac{\text{Line rate}}{\text{Line rate}} = 1 \quad (3.21)$$

$$\Lambda_1 = \frac{\text{Peak rate}}{\text{Line rate}}$$

$$\Lambda_2 = \frac{\text{Mean rate}}{\text{Line rate}}$$

$$Q_{1,1} = \text{Prob_ON_ON}$$

$$Q_{2,2} = \text{Prob_OFF_OFF}$$

For a given specified buffer size, B , and cell loss probability, δ should be calculated by Eq. (3.11). We can then calculate the effective bandwidth $\alpha(\delta)$, or, if we know $\alpha(\delta)$ in advance, we can calculate δ inverse.

3.2.3 Equivalent Bandwidth for Video Model

Despite many recent successes, calculation equivalent bandwidth of video models still remains an unknown. Several approaches are proposed in the literatures.

The approach in [28] is to substitute the original source distributions by ON-OFF distributions, which lend themselves to easy evaluation. In most of the cases, the Cell Loss Probability (CLP) is chosen as the QoS parameter, which is defined as

$$P_{\text{cell loss}}(Y, C) = \frac{\text{lost traffic}}{\text{offered traffic}}$$

where Y denotes the aggregate traffic and C represents the node capacity.

The behavior of real sources can be approximated with an ON-OFF type distribution, which yields a CLP (ON-OFF) given as follow:

$$P_{\text{cell loss}}(Y', C) = \frac{\text{lost traffic (ON-OFF)}}{\text{offered traffic (ON-OFF)}}$$

Here Y' denotes the aggregate traffic emitted by ON-OFF users. This approximation is only acceptable if the inequality below holds.

$$P_{\text{cell loss}}(Y, C) \leq P_{\text{cell loss}}(Y', C) \quad (3.22)$$

This inequality was proven in [20] in the case of general sources. More precisely, the CLP generated by any arbitrary source is always bounded by the CLP generated by the corresponding ON-OFF source. Unfortunately, the above approach can only be applied to zero buffer system which estimates the CLP as the probability of which the total cell arrival rate exceeding the link capacity.

The calculation of effective bandwidth for the video model is beyond the scope of this thesis. For the purpose of our studies in this thesis, we will use a simulation approach to find the relationship between resource requirements and QoS for QTES-modeled video.

Chapter 4: Proposed VCPB Algorithm and Simulation Scheme

4.1 Overview

The design of the congestion control mechanism is driven mainly by the nature of the carried traffic. It is expected that a wide variety of different traffic types over the network, with different requirements in terms of bandwidth, packet loss, delay and number of connections, will be supported over the network. The network will have to provide adequate levels of service to all these traffic types. Since the network has a finite capacity, some ability to allocate resources (link bandwidth, buffer space, switch capacity, processing.) among the contending traffic streams in a “fair” manner is required. The objective of resource management is to meet individual QoS requirements rather than aggregate performance constraints.

In Chapter 2, resource management mechanisms were discussed. Although combined bandwidth/buffer sharing is possible, most bandwidth management mechanisms proposed in recent literature focus attention on bandwidth sharing with CP buffering, which results in low utilization of buffers, or with CS buffering which does not support different cell loss rate requirements.

Furthermore, combined capacity/buffer sharing also makes resource allocation and performance analysis more difficult. The traditional effective bandwidth approach is difficult to applied to these algorithms due to their computational intractability. The interactions among different traffic streams within a shared buffer environment are extremely difficult to quantize.

To solve the above problem, we propose a new algorithm called the VirtualClock with Priority Buffer (VCPB) algorithm. The VCPB algorithm uses different mechanisms for both bandwidth management and buffer management. The two mechanisms are virtually separated and, therefore, allow greater freedom to manage delay and cell loss separately. The VCPB algorithm can provide isolation among different traffic classes having different QoS

requirements, which simultaneously share the available network resources. Thus different QoS requirements can be satisfied at the same time. In addition, the isolation mechanism that the VCPB algorithm provides the traditional effective bandwidth approach to be applied as in homogeneous cases. For the VCPB algorithm, the performance of various traffic classes in heterogeneous environments is upper bounded by their performance in homogeneous traffic environments. Therefore, we can predict the QoS of a traffic class in heterogeneous environments based on their QoS in homogeneous environments which can easily be characterized by the effective bandwidth approach.

4.2 Proposed VirtualClock with Priority Buffer Algorithm

In the VCPB algorithm, the VirtualClock algorithm is used as a bandwidth enforcement mechanism while a priority loss strategy is used as a buffer management mechanism. Traditional priority strategy approaches typically use priority, both as a buffer and as a bandwidth scheduling mechanism. As a result, higher priority traffic streams not only have higher priority buffer access but also higher priority bandwidth access. Thus, lower priority traffic streams cannot have guaranteed bandwidth allocations. A key aspect of the VCPB algorithm that differentiates it from previous approaches is that a higher loss priority is assigned to the traffic class with the lower cell loss rate, assuming that, the cell loss requirements for each class are widely spaced (i.e. differ by at least one order of magnitude). This approach effectively eliminates the disruptions within a shared buffer in terms of cell loss rates. For example, it is easy to see that a traffic stream with a cell loss rate of, say 10^{-9} , has little influence on another traffic stream with a higher cell loss rate, say 10^{-6} , although they share the same buffer. But this is not true in reverse. By providing a higher priority to the traffic stream with 10^{-9} cell loss rate, we also protect it from the disruption of the traffic stream with 10^{-6} cell loss rates. Therefore, both maintain their own cell loss rates. Furthermore, the above mentioned priority mechanism is only used as a buffer management

mechanism and does not affect how and when the cells are serviced. Some typical application requirements are shown in Table 4.1 [35] where the cell loss rates are, in general, at least one order of magnitude different from class to class. This validates our assumption above.

Generic service type	Virtual bandwidth	Tolerable error rate	Acceptable maximum delay	Tolerable delay variation	Maximum burst length
High-quality real-time voice	≤ 0.20 Mb/s	$\leq 10^{-3}$	~ 300 ms	≤ 30 ms	O(bytes)
Real-time video	≥ 4 Mb/s	$\leq 10^{-6}$	~ 100 ms	~ 5 ms	O(kilobytes)
Web browsing	≥ 0.25 Mb/s	$\leq 10^{-5}$	~ 100 ms	~ 10 ms	O(megabytes)
Multiparty network games	≥ 0.10 Mb/s	$\leq 10^{-5}$	~ 50 ms	~ 5 ms	O(kilobytes to megabytes)
Working from home	≥ 1 Mb/s	$\leq 10^{-4}$	~ 1 min	≤ 500 ms	O(megabytes)

Table 4.1: Common broadband to the home services and the required transmission characteristics

As shown in Eq. (4.1), traffic streams with similar traffic characteristics and similar QoS requirements, such as cell loss rate and delay bound (P, D) are defined as homogeneous traffic. Usually, traffic streams which have similar traffic characteristics will have very similar QoS requirements. However, in some cases, the QoS requirements may be different. In these cases, the traffic streams can be decomposed into different types of homogeneous traffic. Several kinds of homogeneous traffic which commit to the requirements of (P_j, D_j) , are carried through a processor sharing node with capacity C and a buffer pool with size B . Each type of homogeneous traffic has a “logical” FIFO queue. Since a FIFO queuing discipline is ideal for homogeneous traffic, it allows a number of sources to aggregate their traffic to obtain a lower overall delay jitter. Each homogeneous traffic stream j can be treated as a single traffic stream consuming bandwidth C_j .

$$\sum_{i=1}^{n_j(t)} C_i^j \leq C_j \quad (4.1)$$

where $n_j(t)$ is the source number of traffic stream j at time t , C_i^j is the effective bandwidth of

the i^{th} source of traffic stream j .

The number of individual connections inside each FIFO queue may vary from time to time. C_j can be adjusted according to the traffic load.

By using the VirtualClock algorithm, the order in which cells depart the Processor Sharing Node (PSN) is determined by the numerical order of their virtual finishing times. The structure here is similar to Figure 2.2, but time here is real time rather than cell slots. The structure of the PSN node using VCPB algorithm is shown in Figure 4.1. The j^{th} aggregated traffic forms a FIFO “logical” queue and has bandwidth C_j . Let $a_i^j \in R^+$ be the arrival time of the i^{th} cell arriving to FIFO queue j . Therefore, Eq. (2.5) can be rewritten as:

$$F_i^j = \max\{F_{i-1}^j, a_i^j\} + \frac{1}{C_j} \quad \text{for} \quad i \geq 1 \quad \text{and} \quad (4.2)$$

$$F_0^j = 0$$

Priority loss strategy is used as a buffer management mechanism.

The “logical” queues in Figure 4.1 are not physically separated. Let B_j be the queue size for the homogeneous traffic stream j , such that $\sum_{j=1}^J B_j \leq B$. In the VCPB algorithm, we assign the loss priority of each traffic class or connection according to its cell loss requirement. Assume that, the cell loss requirements P_j for each class j are widely spaced (i.e. differ by at least one order of

Figure 4.1: A processor sharing node with VCPB

magnitude). We assign the priority of each class according to its cell loss requirement P_j . The higher the cell loss rate, the lower the cell loss priority. If $P_1 < \dots < P_j < \dots < P_J$, traffic class 1 has the highest priority, while traffic class J has the lowest. For the purpose of cell loss strategy, each cell will be associated with a parameter W_i^j , which indicates whether the cell is overflow or underflow when it arrives at the buffer. If $F_{i-1}^j > a_i^j$ in Eq. (4.2), and $W_i^j = 1$, then the traffic stream which the cell belongs to is overflow. Cells are arriving faster than the reserved bandwidth rate. Otherwise, when $W_i^j = 0$, the cell rate is underflow. When the buffer is not full, every cell can access the buffer upon arrival. When the buffer is full, the cell's ability to access the buffer depends on its cell loss priority. If there is any cell in the buffer which has a lower loss priority than the newly arrived cell, and its parameter W_i^j indicates that it is an overflow cell, it can be pushed out of the buffer by the newly arrived cell; otherwise, the newly arrived cell will be lost. The searching order should start from the lowest priority traffic class and from the last cell in each "logical" queue.

If a cell is pushed out of the buffer, the traffic stream to which this cell is attached must recalculate its Virtual Finishing Times without reference to the removed cell.

The final algorithm can be implemented as follows:

Each homogeneous traffic stream has two variables: *VirtualClock_j* and; *auxVC_j* (which are consistent with the variables in the *VirtualClock* algorithm). Let $vrick_j = 1/C_j$ (second/cell). Recall that we assign a sub-total capacity C_j as the available bandwidth to each homogeneous traffic class. And *real time* is referred as current time.

Any switch will serve cells in the following manner:

- i) Upon receiving the first cell from traffic class j :

- (a) $VirtualClock_j \leftarrow auxVC_j \leftarrow real\ time$;
- (b) $W = 0$;
- (c) Stamp the cell with the $auxVC_j$ and W values;
- (d) put the cell in the head of logical queue which belongs to traffic class j .

- ii) Upon receipt at a cell from traffic class j when the buffer pool is not full:

- (a) $auxVC_j \leftarrow \max(real\ time, auxVC_j)$;
- (b) if $real\ time \geq auxVC_j$, $W = 0$; otherwise, $W = 1$;
- (c) $VirtualClock_j \leftarrow (VirtualClock_j + Vtick_j)$, and $auxVC_j \leftarrow (auxVC_j + Vtick_j)$;
- (d) Stamp the packet with the $auxVC_j$ value and W value;
- (e) Put the cell at the end of the logical queue which belongs to traffic class j .

- iii) Upon receipt of a cell from traffic class j when the buffer pool is full: Check the time stamp of the last cell in logical queue of class $J, J-1, \dots, j+1$ (whose priority is lower than traffic class j), starting from class J .

If we find that the last cell in the logical queue of class i is associated with $W = 1$ when $i \geq (j + 1)$, stop checking and do the following:

- (a) push out the last cell in logical queue i , $VirtualClock_i \leftarrow (VirtualClock_i - Vtick_i)$, $auxVC_i \leftarrow$ time stamp of the last cell current in the logical queue i ;
 - (b) $auxVC_j \leftarrow \max(real\ time, auxVC_j)$;
 - (c) if $real\ time \geq auxVC_j$, $W = 0$; otherwise, $W = 1$;
 - (d) $VirtualClock_j \leftarrow (VirtualClock_j + Vtick_j)$, and $auxVC_j \leftarrow (auxVC_j + Vtick_j)$;
 - (e) Stamp the packet with the $auxVC_j$ value and W value;
 - (f) Put the cell at the end of the logical queue which belongs to traffic class j ;
- otherwise:
lose the newly arrived cell of traffic class j .

- The switch always serves the cells at the head of logical queues, in the order of their time stamp.

The description above highlights the difference between the VirtualClock algorithm and the VCPB algorithm. The difference is how to insert the cells in the queues. In the VirtualClock algorithm, the cell is inserted into the outgoing queue according to its timestamp. When the buffer is full, cells are lost in the order of their timestamps. In the VCPB algorithm, whether the cells are lost is not decided by its time stamp or the time it arrives, but by its priority. The VCPB algorithm maintains several logical queues. Newly arrived cells can push out the cells with lower loss priority when the buffer is full. Also, since the switch always serves the cells at the head of logical queues in the order of their timestamps, it is similar to having an outgoing queue according to its timestamp. Logical queues can be maintained by using pointer lists in implementation.

4.3 Traffic Descriptor

The network can guarantee a different QoS requirements for different traffic classes by using the VCPB algorithm as a resource management mechanism. It can also maintain different QoS requirements for individual connections inside each traffic class due to homogeneous nature of the traffic stream. However, the network also needs to use the requested QoS and supplied traffic descriptor to establish whether sufficient resources are available to admit any new connection in order to prevent congestion. Recall that effective bandwidth is the traffic descriptor which relates to the desired QoS and required resources

There exists a remarkable collection of results for multiple types of streams sharing a FIFO buffer with deterministic service rate, leading to an effective bandwidth and the accompanying linear constraint shown in Eq. (3.10). However, all these results were generated under the scenario of FIFO queueing systems which cannot meet individual QoS requirements.

In [10], de Veciana and Kesidis extended the equivalent bandwidth approach to also include individual statistical constraints. Instead of using completely shared buffers, they assumed physically segregated buffers which eliminate the disruption within the shared buffer. Through the large deviation approach, they found that, for each buffer, there exists a decoupling bandwidth

α_j^* for all independent traffic streams $j \in J$. When all α_j^* are smaller than the corresponding guaranteed bandwidths supported by service policies, the traditional effective bandwidth constraint (i.e. Eq. (3.10)) will be valid, otherwise, the firewall set up by the PGPS service policy comes into play and the requirement depends on the service rate and weighting parameters. While de Veciana and Kesidis's results have shown one possible solution of applying the effective bandwidth approach to satisfy individual QoS requirements, some problems still remain to be solved. One important issue is that, in their solution, physically separated buffers are assumed rather than a shared buffer. While this may simplify the analysis of the queueing system, it also removes the benefit of sharing the buffer pool and results in low usage of the buffer. The other problem is that, the effective bandwidth of each source may depend on the characteristics and requirements of other sources to be multiplexed. This makes the effective bandwidth approach difficult to apply to real applications and, therefore, less useful.

In networks using the VCPB algorithm as a resource management mechanism, performance prediction can be based on the bandwidth requirement C_j and buffer size B . The bandwidth management mechanism in the VCPB algorithm guarantees the bandwidth allocation of C_j . The priority strategy in buffer management provides different cell loss requirements for different traffic classes even though they share the same buffer pool. Although it may be difficult to determine the exact effective bandwidth value for each class in a heterogeneous environment, the traffic performance should be bounded by the performance predicted in a homogeneous environment. Our simulation will show that, in most cases, the bound is tight. Therefore, we can treat the effective bandwidth derived in the homogeneous environment as the effective bandwidth in the heterogeneous environment. This has also greatly simplified the relationship between the characteristics of sources and network resources and makes the effective bandwidth approach more useful

4.4 Network Simulation

4.4.1 Overview of Simulation Tools

Optimized Network Engineering Tools (OPNET) is a comprehensive engineering system capable of simulating large communications networks with detailed protocol modeling and performance analysis capabilities. We will use it as our simulation tool. OPNET features include: graphical specification of models; a dynamic, event-scheduled Simulation Kernel; integrated data analysis tools; and hierarchical, object-based modeling. OPNET's hierarchical modeling structure accommodates special problems such as distributed algorithm development. OPNET delivers open systems methodology and an advanced graphical user interface known as the MIL 3 User Interface. The package consists of a number of tools, each one focusing on particular aspects of the modeling task. These tools fall into three major categories that correspond to the three phases of model development: *Specification*, *Data Collection and Simulation*, and *Analysis*. More detail about OPNET can be found in [31].

When a simulation involves several layers, especially with connection level modelling, dynamic source creation and deletion are necessary. OPNET only supports dynamic processes, the lowest level of the hierarchical structure. However, synthetic traffic streams are required for simulation purposes. The Traffic Generator Software Package (TGSP), which was developed by the Broadband Networks Laboratory at Carleton University, satisfies the requirement. The TGSP generates synthetic traffic streams on the burst level and the cell level. Based on the hierarchical modeling approach, the TGSP is designed in an object-oriented style and implemented in C++. This allows the user to use the model at different layers for different applications. The TGSP can be used either independently or in conjunction with other user-developed software modules.

By writing an interface module in C++, which also models the connection level, we obtain a hierarchical simulator, which integrate three time scales discussed in Chapter 3.

4.4.2 Simulation Structure

The structure of the simulation model is shown in Figure 4.2. It contains two parts: the traffic generator and the network simulator. To simplify the simulation scheme, two types of traffic will be generated by the traffic generator: ON-OFF traffic and, QTES-modeled video traffic. The ON-OFF model as a burst level model can be considered as a model for data traffic. We can also consider it as non-real-time application, which has bursty traffic characteristics and is characterized in terms of the PCR, SCR, and MBS. For those cells which are transferred within the traffic contract, the application expects a low cell loss ratio. There is no delay bound associated with a non-real-time service category. On the other hand, the QTES-modeled video was modeled based on modeling an MPEG video trace of the movie "Last Action Hero". It can be considered as a real-time application, which requires constrained delay, but can tolerate cell loss in some degree.

Sources within each traffic class are identical (with the same parameter settings). Therefore, the performance of homogeneous aggregated traffic stream can be considered to be the same as the performance of individual sources of this traffic class. Also, the call admission control procedure (which is used for congestion control, since connection level is considered.) for individual sources is performed within homogeneous traffic. Aggregated traffic in Figure 4.2 can be decomposed into two types of homogeneous aggregated traffic streams.

The multiplexer configuration for each homogeneous traffic is shown in Figure 4.3. Single traffic source data is generated by the Traffic Generator Software Package, according to the

Figure 4.2: Simulation structure

Figure 4.3: The analyzed multiplexer configuration (implemented by C++ function code)

parameter setting for the connection level. When the source is generated, the CAC procedure will decide whether or not this connection can be set up. Because all sources are identical, and the effective bandwidth for each connection is C_o^j , Eq. (4.1) can then be re-written as:

$$n(t)C_o^j \leq C_j \quad (4.3)$$

Since there is no buffer space in the multiplexer, there is no delay in Traffic Generator. The output from the Multiplexer will become input traffic to the network simulator as shown in Figure 4.4. Individual sources have an equal cell loss priority and they have equal access to the shared buffer to form a FIFO logical queue for this traffic class.

We use mean delay and cell loss probability as our performance matrices. The easiest way to determine the cell loss rate is to examine the actual cell loss rate in a fixed buffer size. However, it is time-consuming if we want to determine the cell loss rate in different buffer size, since we have

Figure 4.4: Network simulator structure in OPNET

to repeatedly run simulations. Hence, an infinite buffer size is used instead to calculate the probability that each buffer size has been reached and that no cell has been dropped. In order to compare the difference between results of infinite buffer size and finite buffer size, we run simulations for one specified buffer size.

4.4.3 Simulation Scenarios and Parameter Settings

First, we define homogeneous and heterogeneous in the context of this thesis. When we say that the simulation is homogenous, we are referring to the structure shown in Figure 4.5. The network still has a total capacity C and a total buffer space B . However, each traffic class has a fixed capacity with no sharing among traffic classes. Therefore, the performance of each single traffic source in this structure is the same as the performance in the homogeneous case. In contrast, the heterogeneous case refers to the structure similar to that shown in Figure 4.4. Bandwidth is shared among all traffic classes, and there is only one server. The performance of each traffic source in this structure is the same as the performance in heterogeneous case.

For parameter setting in a hierarchical traffic model, we follow two principles:

- Maintain at least one order of difference in time scales for different model levels so that the hierarchical traffic modeling is assumed. Higher order time scale differences are desired but not necessary for traditional models which typically show short range dependent structure.

Figure 4.5: Homogeneous traffic structure

- Simulation time must be reasonably long. It must be long enough to let the system enter a steady state, which is necessary for obtaining statistical results.

4.4.3.1 Parameter Settings of Burst Level and Cell Level

We usually distinguish source types (voice, data file transfer and video) by their characteristics in the burst and cell levels. Hence, the parameter settings for these two levels remain unchanged for all simulation scenarios.

According to these principles, the parameters in our simulation are set as follows:

Total simulation time: 2000 seconds

ON-OFF source model:

- Line speed: 96,000 cells/second
- Burst level:

peak rate: 4,800 cells/second,

mean rate: 960 cells/second

mean burst length: 10 cells

- Cell level:

Cell interarrival time (in ON period): $\text{line speed} / \text{peak rate} = 20$ (cell slots)

QTES-modeled video model:

- Line speed: 96,000 slots/second
- Burst level:

Video prototype: MPEG-I (I frame only) from the movie of “Last Action Hero”

AAL type: AAL5

Group of Picture: 6

Video quality factor: $Q=10$

- Cell level:

Cell interarrival time (in ON period): constant within one frame, varies from frame to frame.

SP=1

4.4.3.2 Parameter Settings for Connection Level in Homogeneous Simulations

There are two goals in homogeneous simulations. The first goal is to verify our simulation structure and evaluate the equivalent bandwidth approach. The second is to predict traffic performance.

4.4.3.2.1 Verification of Simulation Structure

The first goal is to verify our simulation structure and evaluated the equivalent bandwidth approach. In this scheme, only the ON-OFF model is considered, since its analytical results exist.

For comparison purposes, the connection level is not considered, rather a constant source number is used.

- Connection level:

constant source number:35

- Available bandwidth for ON-OFF class: 48,000 cells/second

In our study, we intend to consider the traffic behavior at the connection level. That is, in Eq. (3.10), sometimes we have $\sum_{j \in J} \alpha_j < C$, and at the other times we have $\sum_{j \in J} \alpha_j = C$. Network utilization in this condition will drop compared to that without considering the traffic behavior in the connection level. In order to compare performances, we will change the network utilization in different cases while keeping $\sum_{j \in J} \alpha_j \leq C$ valid. The network utilization can be changed by changing the mean source number. This is accomplished by adjusting the two parameters in the

connection level: mean call interarrival time and mean holding length. There are two cases in our study.

Case I:

- Connection level:
 - mean source number: 28.5, maximum source number: 35
 - mean call interarrival time: 13,340 cell slots (about 0.14 second);
 - mean holding length: 400,000 cell slots (about 4 seconds)
- Available bandwidth for ON-OFF class: 48,000 cells/second
- Total bandwidth: 48,000 cells/second

Case II:

- Connection level:
 - mean source number: 31, maximum source number: 35
 - mean call interarrival time: 11,428 cell slots (about 0.12 second);
 - mean holding length: 400,000 cell slots (about 4 seconds)
- Available bandwidth for ON-OFF class: 48,000 cells/second
- Total bandwidth: 48,000 cells/second

For the ON-OFF traffic models in the simulation, effective bandwidths exist and can be calculated by using Eq. (3.11) and Eq. (3.16).

The traffic descriptors for the ON-OFF models are:

- Peak rate: 4,800 cells/second
- Mean rate: 960 cells/second
- Mean burst size: 10 cells
- Line rate: 96,000 cells/second

By using the results from Eq. (3.17), Eq. (3.18), Eq. (3.19), Eq. (3.20) and Eq. (3.21), Eq. (3.16) can be re-written as:

$$\alpha(\delta) = \delta^{-1} \log \left[\frac{1}{2} \left(0.999 + 0.995e^{0.05\delta} + \sqrt{\left(0.999 + 0.995e^{0.05\delta} \right)^2 - 4 \times 0.996e^{0.05\delta}} \right) \right] \quad (4.4)$$

Using the parameter settings specified in the last section, bandwidth with 48,000 cells/second is consumed by 35 ON-OFF sources. The equivalent bandwidth for each source is given by:

$$\alpha(\delta) = \frac{48,000}{35} = 1371$$

The theoretic δ value can be obtained by solving Eq. (4.4), where

$$\delta = 0.0296$$

The δ value from the simulation can be obtained if we determine the slope of the cell loss probability and buffer size In Eq. (3.11).

4.4.3.2.2 Prediction of Traffic Performances

The second goal is to predict traffic performance. In these simulations, different traffic loads and different mixed traffic loads will be examined on the network in heterogeneous traffic simulations.

Case I: Prepare results for performance comparison of homogeneous to heterogeneous simulations of balanced traffic loads. There will be 50% ON-OFF traffic and 50% QTES-modeled video traffic in the network.

For the purposes of this thesis the above traffic load ratio will be referred to as a “balanced traffic load”. The required bandwidth for the ON-OFF traffic is the same as that for the QTES-modeled video traffic. The two traffic classes have the same utilization.

Parameter settings for the ON-OFF source model:

- Connection level:

mean source number: 28.5, maximum source number: 35
 mean call interarrival time: 13,340 cell slots (about 0.14 seconds);
 mean holding length: 400,000 cell slots (about 4 seconds)

- Available bandwidth for ON-OFF class: 48,000 cells/second
- Total bandwidth: 48,000 cells/second

Parameter settings for the QTES-modelled video source:

- Connection level:

mean source number: 8.6, maximum source number: 10
 mean call interarrival time: 1,111,333 slots (about 11.6 seconds);
 mean holding length: 9600,000 slots (about 100 seconds)

- Available bandwidth for Video class: 48,000 cells/second
- Total bandwidth: 48,000 cells/second

Case II: Prepared for performance comparison of homogeneous to heterogeneous simulations of unbalanced traffic load. There will be 25% ON-OFF traffic and 75% QTES-modeled video traffic in the network.

The above traffic load ratio will be referred to as a “unbalanced traffic load”. The required bandwidth for ON-OFF traffic is the one-third of that for QTES-modeled video traffic. The two traffic classes have the same utilization.

Parameter settings for the ON-OFF source model:

- Connection level:

mean source number: 13.75, maximum source number: 18
 mean call interarrival time: 26,680 cell slots (about 0.28 seconds);
 mean holding length: 400,000 cell slots (about 4 seconds)

- Available bandwidth for ON-OFF class: 24,000 cells/second
- Total bandwidth: 24,000 cells/second

Parameter settings for the QTES-modelled video source:

- Connection level:
 mean source number: 13, maximum source number: 15
 mean call interarrival time: 680,000 slots (about 7.08 seconds);
 mean holding length: 9600,000 slots (about 100 seconds)
- Available bandwidth for Video class: 72,000 cells/second
- Total bandwidth: 72,000 cells/second

4.4.3.3 Parameter Settings for Connection Level in Heterogeneous Simulations

In these simulations, we multiplex the ON-OFF traffic and video traffic injected into the network. Recall that the parameter settings for bust level and cell level remain unchanged.

Case I: Balanced traffic load. There is 50% ON-OFF traffic and 50% QTES-modeled video traffic in the network.

- Connection level:
 using the same settings in section 4.4.3.2.2 Case I for ON-OFF traffic and QTES-modeled video traffic.
- Total bandwidth: 96,000 cells/second

Case II: Unbalanced traffic load. There is 25% ON-OFF traffic and 75% QTES-modeled video traffic in the network.

- Connection level:
 using the same settings in section 4.4.3.2.2 Case II for ON-OFF traffic and QTES-modeled video traffic
- Total bandwidth: 96,000 cells/second

The performance will be different when we implement different bandwidth management and buffer management mechanisms. It also differs from the performance predicted in the homogeneous case.

4.4.3.4 Parameter Settings for Finite Buffer Simulation

The previous simulations use infinite buffer size. Cell loss probability is the probability that the buffer overflows at certain buffer sizes. Instead of losing the cell, it is put at the end of the queue. Queue tail distribution will be obtained in one replication. Otherwise, different replications have to be run for different buffer sizes.

In order to compare the difference in simulation results between the two schemes, we chose one finite buffer size to run the simulations under the proposed the VCPB algorithm using the parameter settings in section 4.4.3.3 Case I (balanced traffic load) and:

- Buffer size: 80

The reason for choosing this value is that the valid data for ON-OFF traffic can only be collected up to buffer size 80. This will be seen from the simulation results presented in Chapter 5.

Chapter 5: Simulation Results

5.1 Results for Homogeneous Simulations

Simulation results for homogeneous traffic are presented and discussed in this section.

5.1.1 Verification of Simulation Models

To verify the simulation structure we described in Figure 4.2 and to evaluate the equivalent bandwidth approach, simulation results are compared with the analytical results for the ON-OFF model obtained by using the large deviation approach.

In this case, burstiness at the connection level is not considered. Therefore, we actually determine the equivalent bandwidth for a single source by specifying capacity and source number. The analytical value of the convergence rate δ , has already been given in Chapter 4. The value of δ will be the slope in figure of loss probability vs. buffer size.

Using the parameter settings in section 4.4.3.2.1, with a total simulation time of 2000 seconds, approximately 6.7×10^7 cells are generated. It is not unreasonable to report the accuracy of cell loss as 10^{-5} . The simulation results are given in the form of cell loss probability versus buffer size. Ten replications are run in order to give 95% confidence interval for the simulation results. The comparison of the two results are given in Figure 5.1

In Figure 5.1, the two lines which represent analytical results and simulation results are parallel and the slope of the two lines are nearly identical. This shows that the analytical results and the simulation results have the same convergence rate. Effective bandwidth as a traffic descriptor can translate the QoS requirement of a source into network resources. However, the two lines do not overlap and there is an order of difference between them. The reason for that is probably due to the assumption made in the analytical method: that the result is obtained when buffer size is infinite. If the buffer size is very large, cell loss probability $\log(P(Q > b))$ will reach

Figure 5.1: Queue tail distribution comparison of simulation result and analytic result

a larger negative value. Compared to this value, one order of difference is acceptable. This figure has verified our simulation structure and traffic model.

If we consider the burstiness at the connection level, as discussed in the hierarchical model in Chapter 3, the network performance will be different from it without considering the burstiness in connection level. When we consider the burstiness in the connection level, the number of sources within the network is changing. That is, in Eq. (3.10), sometimes we have $\sum_{j \in J} \alpha_j < C$, and at

other times we have $\sum_{j \in J} \alpha_j = C$. When we have $\sum_{j \in J} \alpha_j < C$, each source can use more capacity than its effective bandwidth. Therefore, better performance would be expected. Also, network utilization will be lower. Under the same parameter settings at the burst level and the cell level, the network utilization is the highest in the case without burstiness in the connection level (i.e.

Figure 5.2: Queue tail distribution comparison of analytical, static and dynamic source (different utilization) results for ON-OFF traffic

$\sum_{j \in J} \alpha_j = C$) where the maximum number of source is limited in the case with burstiness in connection level (i.e. $\sum_{j \in J} \alpha_j \leq C$).

The results in Figure 5.2 show the performance for different network utilization with the same parameter settings in the burst level and the cell level. The performance in the case which does not consider the burstiness in the connection level is conservative. It can be the upper bound for any other case which considers the burstiness in the connection level. Of course, the maximum number of source should be limited. Hence, the QoS predicted by the effective bandwidth is based on parameters in the burst level and the cell level is conservative.

5.1.2 Performance Comparison of Two Traffic Models

There are two profiles of mixed traffic loads in the heterogeneous traffic simulations: 50% ON-OFF traffic and 50% QTES-modeled video traffic (the balanced case); 25% ON-OFF traffic and 75% QTES-modeled video traffic (the unbalanced case). In order to predict the traffic performance, we will examine the performance in terms of cell loss probability and mean delay when there is no sharing, as shown in Figure 4.5. Ten replications are run in order to give 95% confidence interval for the results.

Results of cell loss probabilities are shown in Figure 5.3 and Figure 5.4. Although the ON-

Figure 5.3: Queue tail distribution comparison of ON-OFF traffic and QTES-modeled traffic in balanced traffic load case

Figure 5.4: Queue tail distribution comparison of ON-OFF traffic and QTES-modeled traffic in unbalanced traffic load case

OFF model and the QTES model generate similar SRD traffic, their performance is quite different. For same size buffers as shown in Figure 5.3, the cell loss probability for the QTES-modeled video is much higher and the decrease is much more gentle compared to that in the ON-OFF model traffic due to its bursty characteristic. When the buffer size is small, let's say up to 300 cells in our simulations, the cell loss probability vs. buffer size is not linear for the video traffic modeled by the QTES process as it is for the ON-OFF traffic. This effect is the result of the stronger autocorrelation of the video traffic. When the buffer size increases, the cell loss probability is again linear since traffic generated by QTES is still SRD traffic. However, the non-linearity of the QTES-modeled video is not clearly shown in Figure 5.4. By comparing Figure 5.3 with Figure 5.4, the cell loss probabilities for ON-OFF are seen to remain almost the same while the cell loss probability for QTES-modeled video in Figure 5.4 is lower than that in Figure 5.3. Since the traffic loading in the two schemes are different, the QTES-modeled video traffic has a larger multiplexing gain due to its bursty characteristic when there are more sources multiplexed together. Also, cell loss probability vs. buffer size shows more linearly. Compared to the QTES-modeled video traffic, the ON-OFF traffic is less bursty. Therefore, there is not much difference shown in cell loss probability when the numbers of sources vary.

In addition to cell loss probability performance matrix, we give the values of mean delay. Mean delay shows a smaller value in Table 5.2 compared to Table 5.1 for QTES-modeled video traffic. The benefit is due to the multiplexing gain, since more sources are multiplexed in the

Traffic Type	Mean Delay	95% Confidence Interval
ON-OFF	2.94×10^{-5}	(+/-) 2.0×10^{-7}
Video	8.78×10^{-4}	(+/-) 4.9×10^{-5}

Table 5.1: Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) in balanced traffic load case

Traffic Type	Mean Delay	95% Confidence Interval
ON-OFF	1.24×10^{-4}	(+/-) 2.4×10^{-6}
Video	1.32×10^{-4}	(+/-) 3.9×10^{-5}

Table 5.2: Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) in unbalanced traffic load case

unbalanced traffic load case. For ON-OFF traffic, multiplexing gain are observed on the mean delay too. The mean delay in Table 5.1 is smaller when there are more sources.

We are going to compare this performance with that of the heterogenous traffic case using the same parameter settings

5.2 Results for Heterogeneous Simulations

Simulation results for heterogeneous traffic are presented and discussed in this section.

5.2.1 Performance of FIFO Queueing discipline (FIFO Scheduling and FCFU Buffer Management)

As we discussed in Chapter 2, the FIFO queueing discipline supports single QoS requirement. Traffic sources which generate more cells will consume more resource. Traffic flows have equal access to the buffer and cells are served in the order of the time they arrive at the switch. In the following simulation scheme, we use a balanced traffic load.

Simulation results confirm that the two types of traffic suffer the same mean delay and the same cell loss probabilities in the heterogeneous case even though they have different

performances in the homogenous case. In Figure 5.5, the cell loss of the ON-OFF traffic and the cell loss of the QTES-modeled video traffic are equal to the cell loss of the aggregated traffic. QTES-modeled video traffic improves cell loss performance by benefitting from multiplexing with ON-OFF traffic, which has a low cell loss requirement in the homogeneous case. However, the ON-OFF traffic is sacrificed. Its performance in terms of cell loss rate is degraded even though there should be a multiplexing gain for both traffic streams. The QTES-modeled video is more bursty than the ON-OFF traffic. A burst of video traffic can consume a lot of buffer space and cells from ON-OFF traffic could be lost. A FIFO queueing discipline cannot guarantee the required QoS.

Traffic Type	Mean Delay	95% Confidence Interval
ON-OFF	1.31×10^{-5}	$(+/-) 1.9 \times 10^{-6}$
Video	1.72×10^{-5}	$(+/-) 1.8 \times 10^{-6}$

Table 5.3: Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) while using FIFO discipline in balanced traffic load case

Mean delay has been calculated separately for each traffic as shown in Table 5.3. Although the values of the two types of traffic are not exactly the same due to the accuracy of the simulation, they can be considered to be very close in terms of the 95% confidence interval value. In comparison with the values of the homogeneous case in Table 5.1, the mean delay of both traffic types is not degraded. FIFO queueing allows a number of sources to aggregate their traffic to obtain a lower overall delay jitter. Therefore, QTES-modeled video traffic which has a higher

Figure 5.5: Queue tail distribution of aggregated traffic compared with homogeneous case in balanced traffic load case

mean delay in the homogeneous case has a lower mean delay when it is multiplexed with ON-OFF traffic. But there is no proof that the mean delay for the ON-OFF traffic will also be lower compared to that in the homogeneous case since it is multiplexed with the traffic with the higher mean delay. From the view-point of ON-OFF traffic, the total capacity is larger than that in the homogeneous case, but the queueing length is also longer than in the homogeneous case. These two factors affect the change in the mean delay value at the same time. However, with the parameter settings used in our simulations, the ON-OFF traffic mean delay is also lower.

5.2.2 Performance of VirtualClock Scheduling with FCFU Buffer Management

In this simulation scheme, we use the VirtualClock as a scheduling algorithm instead of using FIFO scheduling where the buffer management mechanism remains unchanged.

First, we use a balanced traffic load scheme in order to compare the performance against that in a FIFO queueing discipline.

Traffic Type	Mean Delay	95% Confidence Interval
ON-OFF	7.24×10^{-6}	(+/-) 6.8×10^{-7}
Video	2.07×10^{-5}	(+/-) 1.8×10^{-6}

Table 5.4: Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) while using VirtualClock scheduling with FCFU buffer in balanced traffic load case

In Table 5.4, ON-OFF and QTES-modeled video traffic are seen to have different delays, even though they require the same capacity and have the same utilization. The QTES-modeled video traffic suffers a larger delay because it is more bursty than the ON-OFF traffic. When using VirtualClock as a scheduling algorithm among different traffic classes, it controls the average transmission rate of each traffic class, enforcing its average resource usage according to the specified throughput, thus providing firewall protection between different traffic classes. Both traffic classes have the same guaranteed bandwidth usage as they have in the homogeneous traffic case, and they might obtain additional bandwidth when they are multiplexed together. Therefore,

the mean delay values are smaller when compared with those in Table 5.1. That is the benefit of multiplexing gain. Furthermore, the mean delay for the QTES-modeled video traffic is larger in Table 5.4 compared to Table 5.3; this shows that QTES-modeled video traffic consumes more bandwidth under a FIFO scheduling scheme than it reserved, and the ON-OFF's mean delay is smaller as shown in Table 5.4. A burst of video traffic could use more capacity under a FIFO queueing discipline, it can even steal the capacity which belongs to ON-OFF traffic. With the VirtualClock algorithm, video traffic can only use more capacity when ON-OFF traffic has no cell to send.

Unfortunately, the cell loss probabilities for both traffic streams still cannot be separated. They will remain unchanged to what are shown in Figure 5.5, because the same buffer management mechanism (FCFU) is used. This is independent of the scheduling algorithm. Whether the cell is lost or not depends on the total number of cells in the buffer. Using a separated buffer approach can certainly guarantee the cell loss probability as determined in the homogeneous case. Since VirtualClock has already guaranteed each traffic's average bandwidth usage, and there is additional bandwidth multiplexing gain among traffic classes. Cells are served no later than that in the homogeneous case. Therefore, queue sizes for each traffic stream are no longer than in the homogeneous case. However, in the completely separated buffer scheme, the total buffer size is equal to the sum of each individual buffer. Due to the multiplexing gain, buffer occupied utilization drops. In order to improve buffer utilization, we intend to use a shared

Figure 5.6: Queue tail distribution of aggregated traffic compared with homogeneous Case in unbalanced traffic load case

buffers, but an additional buffer management mechanism has to be used in order to provide different guaranteed cell loss rates.

The similar performance of VirtualClock scheduling with FCFU buffer management is obtained in the unbalanced traffic load case. Results are shown in Figure 5.6 and Table 5.4.

Different mean delays are maintained for the ON-OFF traffic and the QTES-modeled video traffic. The mean delay values are smaller than that in Table 5.1 while obtaining the uniform cell loss probability for both traffic streams.

Traffic Type	Mean Delay	95% Confidence Interval
ON-OFF	1.38×10^{-5}	(+/-) 6.0×10^{-7}
Video	2.88×10^{-5}	(+/-) 2.8×10^{-6}

Table 5.5: Comparison of mean delay (ON-OFF traffic and QTES-modeled video traffic) while using VirtualClock Scheduling with FCFU buffer in unbalanced traffic load case

5.2.3 Performance of The VCPB Algorithm

As shown in the previous simulation results, the FIFO queueing discipline, and FIFO scheduling plus FCFU buffer management, can provide a uniform QoS to all traffic streams. Traffic streams in the network suffer the same mean delay and the same cell loss rate. Furthermore, different delay requirements can be maintained by substituting the VirtualClock algorithm for a FIFO scheduling even though the FCFU buffer management remains unchanged. Therefore, FCFU buffer management still results in a uniform cell loss rate for all traffic streams.

In the VCPB algorithm, a priority strategy is used as a buffer management mechanism in order to provide different cell loss requirements for different traffic streams. Loss priority is assigned according to the cell loss requirements.

In the simulation implementing VCPB algorithm, the balanced traffic load and the unbalanced traffic load were used to verify the proposed algorithm.

Figure 5.7: Queue tail distribution for ON-OFF traffic and QTES-modeled video with VCPB algorithm in balanced traffic load case

5.2.3.1 Balanced traffic load case

The simulation results are also presented for the QoS in terms of mean delays and cell loss rates vs. buffer size. Mean delays are the same as in Table 5.4, since the VirtualClock algorithm is used as scheduling mechanism in the VCPB algorithm. For cell loss probability, differential cell loss rates are obtained instead of a uniform cell loss rate due to the priority strategy in buffer management as shown in Figure 5.7. Therefore, a different QoS can be provided in the networks using the VCPB algorithm as a resource management method.

Furthermore, the traffic descriptor “effective bandwidth” is used to predict the network performance or to translate the QoS requirements into the network’s resource. The traditional approach to effective bandwidth is under FIFO queueing discipline. With an unchanged buffer size B , the required capacity for maintaining the network performance is the sum of the effective bandwidths of all sources. The estimated effective bandwidth under these conditions is the same as that in homogeneous environments due to the identical QoS requirements. To estimate the effective bandwidth in a heterogeneous environment is difficult due to the differing QoS requirements and different traffic characteristics. When other scheduling algorithms than FIFO scheduling are used in order to provide different delay requirements, the estimation becomes even more difficult. Furthermore, to maintain different cell loss requirements, assumptions, such as a separated buffer, have to be made.

The VCPB algorithm can provide different cell loss requirements for different traffic streams in a shared buffer with priority strategy. This simplifies the estimation of effective bandwidth in heterogeneous environment. However, the impact of different traffic characteristics on the estimation of effective bandwidth is still difficult to calculate.

Case Type	Mean Delay	95% Confidence Interval
Homogeneous	2.94×10^{-5}	$(+/-) 2.0 \times 10^{-7}$
Heterogeneous	7.24×10^{-6}	$(+/-) 6.8 \times 10^{-7}$

Table 5.6: Comparison of mean delays for ON-OFF traffic with VCPB algorithm in balanced traffic load case

Case Type	Mean Delay	95% Confidence Interval
Homogeneous	8.78×10^{-4}	$(+/-) 4.9 \times 10^{-5}$
Heterogeneous	2.07×10^{-5}	$(+/-) 1.8 \times 10^{-6}$

Table 5.7: Comparison of mean delays for video traffic with VCPB algorithm in balanced traffic load case

In comparing the performance of each traffic class under the VCPB algorithm with that in the homogeneous case, in Figure 5.8, Figure 5.9, Table 5.6 and Table 5.7, we can see that the performance of each traffic class under VCPB algorithm is bounded by its performance in the homogeneous environment. First, different delay and cell loss requirements are met under the VCPB algorithm. The mean delay of each traffic class is smaller compared to that in the homogeneous case, and; it will not become larger than that in the homogeneous case due to the VirtualClock algorithm and the multiplexing gain, since the bandwidth usage is guaranteed. As

Figure 5.8: Queue tail distribution for ON-OFF traffic with VCPB algorithm in balanced traffic load case

Figure 5.9: Queue tail distribution for QTES-modeled video traffic with VCPB algorithm in balanced traffic load case

for cell loss rate, since the higher loss priority is given to the ON-OFF traffic which has the low cell loss rate, it seems that the buffer is used by ON-OFF traffic only from its view-point. It will not get worse, rather, better due to the bandwidth sharing. Hence, the cell loss rate of ON-OFF traffic can be bounded by that in the homogeneous case. For QTES-modeled video, its cell loss rate also decreased due to the bandwidth sharing. By giving away the higher priority of accessing the buffer, the cell loss rate in QTES-modeled video traffic would increase to some degree, but the influence of ON-OFF traffic on video is not critical since their cell loss requirements have more than one order of difference. Therefore, the cell loss rate of QTES-modeled video traffic will also be bounded by that in the homogeneous case. The comparisons for ON-OFF traffic and QTES-modeled video traffic are shown in Figure 5.8 and Figure 5.9.

5.2.3.2 Unbalanced traffic load case

The VCPB algorithm is also verified under unbalanced traffic load case.

The simulation results are shown in Figure 5.10, Figure 5.11, Figure 5.12, Table 5.8 and Table 5.9. These results show, again, that different delay and cell loss requirements can be met by the VCPB algorithm. The performance of each traffic stream under VCPB algorithm in the heterogeneous case is bounded by that in homogeneous case.

Case Type	Mean Delay	95% Confidence Interval
Homogeneous	1.24×10^{-4}	$(+/-) 2.4 \times 10^{-6}$
Heterogeneous	1.38×10^{-5}	$(+/-) 6.0 \times 10^{-7}$

Table 5.8: Comparison of mean delays for ON-OFF traffic with VCPB algorithm in unbalanced traffic load case

Case Type	Mean Delay	95% Confidence Interval
Homogeneous	1.32×10^{-4}	$(+/-) 3.9 \times 10^{-5}$
Heterogeneous	2.88×10^{-5}	$(+/-) 2.8 \times 10^{-6}$

Table 5.9: Comparison of mean delays for QTES-modeled video traffic with VCPB algorithm in unbalanced traffic load case

Figure 5.10: Queue tail distribution for ON-OFF traffic and QTES-modeled video with VCPB algorithm in unbalanced traffic load case

Figure 5.11: Queue tail distribution for ON-OFF traffic with VCPB algorithm in unbalanced traffic load case

Figure 5.12: Queue tail distribution for QTES-modeled video traffic with VCPB algorithm in unbalanced traffic load case

5.2.3.3 Further Discussion on The VCPB Algorithm

As the simulation results show with both balanced and unbalanced traffic loads, the VCPB algorithm can provide different QoS requirements in a shared environment. The performance of the various traffic classes in the heterogeneous environments is upper bounded by their performance in the homogeneous traffic environments. Therefore, we can predict the QoS of a traffic class in heterogeneous environments based on its QoS in homogeneous environments which can easily be characterized by the effective bandwidth approach. However, the estimation of effective bandwidth based on a homogeneous environment is conservative since the impact of different traffic characteristics on multiplexing is not considered. Furthermore, the different traffic loads can also effect the network performance.

5.2.4 Using Finite Buffer in Simulations

The results in the previous simulations were obtained by using an infinite buffer size. The cell loss probability is the probability that the buffer is overflows at a certain buffer size. Instead of losing the excess cell, we put it at the end of the queue. Queue tail distribution can be obtained in one replication, otherwise, we have to run different replications for different buffer sizes.

In order to compare the simulation results produced by the two schemes, we chose a finite

buffer size to run the simulations under the proposed the VCPB algorithm and balanced traffic load.

The buffer size is chosen to be 80, because we can only obtain valid result up to this buffer size for the ON-OFF traffic from the previous simulation. Cell loss probability in the case of a finite buffer is expected to be smaller than that in an infinite buffer size case.

Case Type	Mean Delay	95% Confidence Interval
Finite Buffer Size	7.10×10^{-6}	(+/-) 4.9×10^{-7}
Infinite Buffer Size	7.24×10^{-6}	(+/-) 6.8×10^{-7}

Table 5.10: Comparison of mean delays under finite and infinite buffer size schemes for ON-OFF traffic with balanced traffic load case

Case Type	Mean Delay	95% Confidence Interval
Finite Buffer Size	1.68×10^{-5}	(+/-) 2.0×10^{-6}
Infinite Buffer Size	2.07×10^{-5}	(+/-) 1.8×10^{-6}

Table 5.11: Comparison of mean delays under finite and infinite buffer size schemes for QTES-modeled video traffic with balanced traffic load case

Comparison for mean delay values are shown in Table 5.10 and Table 5.11. In Table 5.10, values for ON-OFF traffic shows no difference in terms of 95% confidence interval. In Table 5.11, there is some difference in the two schemes since the video traffic is more bursty compared to ON-OFF traffic. Queueing length might be longer for the QTES-modeled video than that for ON-OFF traffic. Keeping the cell in the queue, instead of dropping it, will affect the results for mean delay when the cell should have been dropped.

Cell loss probabilities are compared in Table 5.12 and Table 5.13.

For both traffic classes, results from the finite buffer size case are smaller. Hence results from the infinite buffer size scheme in previous simulations are conservative.

Case Type	Cell Loss Probability	95% Confidence Interval
Finite Buffer Size	-6.12	(+/-) 0.21
Infinite Buffer Size	-5.37	(+/-) 0.25

Table 5.12: Comparison of mean delays under finite and infinite buffer size schemes for ON-OFF traffic with balanced traffic load case

Case Type	Cell Loss Probability	95% Confidence Interval
Finite Buffer Size	-3.24	(+/-) 0.08
Infinite Buffer Size	-2.67	(+/-) 0.13

Table 5.13: Comparison of mean delays under finite and infinite buffer size schemes for QTES-modeled video traffic with balanced traffic load case

Chapter 6: Conclusions and Recommendations for Further Study

6.1 Conclusions

In this thesis, we proposed the VCPB algorithm as a new resource management scheme which can meet individual QoS requirements while preserving maximum resource sharing. In addition to the bandwidth sharing mechanism introduced by the VirtualClock algorithm, the VCPB algorithm also implemented a new buffer management scheme based on a new priority strategy. This allows VCPB to commit to both delay and cell loss requirements for different traffic streams in a shared environment. Furthermore, we have shown that, for the VCPB algorithm, the performance of various traffic classes in heterogeneous environments is upper bounded by their performance in homogeneous traffic environments. Therefore, we can predict the QoS of a traffic class in heterogeneous environments based on its QoS in homogeneous environments -- which can be easily characterized by the effective bandwidth approach. Traffic descriptors (e.g. effective bandwidth) can now be treated as the attributes of a traffic class which does not depend on other traffic classes to be multiplexed and, therefore, can be pre-calculated. Congestion control approaches based on traffic descriptors (e.g. CAC) can become simple and straightforward by comparing the aggregate effective bandwidth with the available link rate as in a circuit-switching network.

Cost-efficient operation of the network can be achieved by a statistical multiplexing of variable bit rate traffic streams and sharing of the resources. When different types of traffic and service are merged into a single network, distinguishing between the two basic principles of isolation and sharing is both fundamental and novel. They both play important roles when sources are bursty and resources are limited. Isolation is fundamental and mandatory. The network cannot make any commitments if it cannot prevent the unexpected behavior of one

source from disrupting others. Sharing is important but not fundamental. If bandwidth resources were plentiful, the effective behavior could be guaranteed by allocating to each source its required maximum resources usage; in this case sharing need not be considered. However, plentiful resources do not eliminate the need for isolation, and still need to ensure that each source does not use more than its allocated portion of resources. Hence resource management with these two objectives is necessary to prevent disruptions among different bursty traffic streams and assure that the networks meet their commitments on QoS requirements.

The objective of resource management is to meet individual QoS requirements while preserving maximum resource sharing. The network has to allocate finite resources among the contending traffic streams in a “fair” manner. Resource management includes bandwidth management and buffer management. Although combined bandwidth/buffer sharing is possible, most bandwidth management mechanisms proposed in literature focus attention on bandwidth sharing with simple completely separated buffers or completely sharing a buffer; it simplifies the network analysis. The others focus on buffer sharing with a fixed bandwidth allocation which also eliminates the bandwidth sharing benefit. In our proposed VCPB algorithm, we use different mechanisms for bandwidth management and buffer management, one cannot affect the other. The VirtualClock algorithm is used as a bandwidth enforcement mechanism while the priority strategy is used as a buffer management mechanism. Neither the VirtualClock algorithm nor the priority strategy are new, the contribution of this thesis is to combine both of them as a new resource management scheme to provide individual QoS requirements while preserving maximum resource sharing. L. Zhang [13] has done extensive analysis and simulation work on the performance of the VirtualClock algorithm, and showed that the VirtualClock algorithm can provide guarantee bandwidth usage. A key part of the VCPB algorithm is that, as opposed to previous approaches, we assign a higher loss priority to the traffic with a lower cell loss rate. In this sense, it is new. This approach effectively eliminates the disruptions within a shared buffer in terms of cell loss rates.

The relationship between the resource management structure of a network and the desired QoS of each transported traffic stream typically manifests itself in the corresponding traffic

descriptors. "Effective bandwidth" is one of the proposed traffic descriptors. It tries to translate the traffic characteristics and QoS requirement into network resource requirements, and helps to accurately predict the network performance level. Performance prediction in heterogeneous environments is complicated. The interactions among traffic streams within shared resources is typically not linear in the number of sources, nor can it usually be decoupled across the different traffic streams, since the FIFO queueing discipline is assumed in traditional approaches to effective bandwidth. We proposed that the performance prediction is based on a homogeneous environment: that a single traffic stream will consume the same bandwidth in a heterogeneous environment as in a homogeneous environment when based on the same buffer space, and as if there were no sharing with the other traffic streams.

The proposed VCPB algorithm is verified via simulation with a hierarchical traffic model built in OPNET interfaced to an external traffic generator under different mixed traffic loads. Our simulations involved several layers in the traffic model, especially the connection level, dynamic source creation and deletion are necessary. OPNET only supports the dynamic processes the lowest level of the hierarchical structure. Also, synthetic traffic streams are required. The Traffic Generator Software Package, which was developed by the Broadband Networks Laboratory at Carleton University, can be used to generate synthetic burst level and cell level traffic streams which are based on existing traffic models. The software package is designed in object-oriented style and implemented in C++. This allows the user to use the model at different layers for different applications. Unfortunately, this software package does not include the connection level. Furthermore, interface modules are needed to integrate this software package with OPNET which is written in C codes. The Traffic Generator Software Package has been updated to include the connection level models and to work with OPNET through the interface modules during the research phase of this thesis.

With this updated software package, extensive simulations were run for this thesis under different homogeneous and heterogeneous traffic loads using different resource management mechanisms. The simulation results show that the VCPB algorithm meets both the delay and the cell loss requirements for different traffic streams in a shared environment under different mixed

traffic loads. Furthermore, the performance of various traffic classes in heterogeneous environments is upper bounded by their performance in homogeneous traffic environments. Therefore, resources required to meet a desired QoS can be estimated based on homogeneous environments and does not depend on the other multiplexed traffic streams. In a homogeneous environment, the desired resources can be easily characterized by the effective bandwidth approach. Also the large deviation approach for effective bandwidth which is based on the Markov model is evaluated through simulation.

6.2 Recommendations for Further Study

1) Additional functions in the VirtualClock algorithm could be adapted into the VCPB algorithm. In the VirtualClock algorithm, there are two basic functions: data forwarding and flow monitoring. The function of data forwarding assures that each flow or connection will receive the resources that it has reserved. From another viewpoint, VirtualClock plays the role of a “flow meter” driven by cell arrivals. Thus it can be conveniently used for flow measurement. Because the VirtualClock is advanced according to the flow’s specified average transmission rate or reserve bandwidth, the difference between the VirtualClock and the real time indicates how closely a running flow is following its specified transmission rate. We can monitor each traffic stream by periodically comparing its VirtualClock with the real-time clock, providing feedback to the flow source whenever its actual throughput departs significantly from the reserved rate. In the VCPB algorithm, we use the function of data forwarding as our bandwidth management mechanism and assume that the traffic is well-behaved, therefore, it is not necessary to provide feedback to the source. Advanced study on the VCPB algorithm would take the flow monitoring function into account.

2) A more complicated network topology could be built by using OPNET. The network topology of the simulations for this thesis is a single switch. However, OPNET is designed to allow modeling with varying levels of detail, and the model developer must select an appropriate

model complexity. Provided that a model is sufficiently representative of an actual system, OPNET simulation models can provide accurate estimates of true system performance and realistic observations of true system behavior by executing one or more simulation runs of the model.

3) The VCPB algorithm could be implemented in the network with multiple traffic classes with different QoS requirements. There were two types of traffic models used in the simulation models developed for this thesis: ON-OFF traffic and, QTES-modeled video traffic. The Traffic Generator Software Package is capable of generating four types of traffic, where each type of traffic can also be decomposed into different sub-traffic classes by adjusting the parameter settings. This will give us the flexibility to generate a number of homogeneous traffic classes with different QoS requirements for simulation purposes.

4) Analytical study of the VCPB algorithm. Analytical study of the VCPB algorithm is difficult and complicated because of the combined sharing of bandwidth and buffer. The proposed VCPB algorithm was verified via simulations in this thesis, and the analytical study is left for future work.

References:

- [1] C. Lefelhocz, Bryan Lyles, S. Shenker and L. Zhang, "Congestion Control for Best-Effort Service: Why We Need a New Paradigm", *IEEE Network*, vol. 10, no. 1, January/February 1996, pp. 10-19
- [2] G. Veciana, G. Kesidis and J. Walrand, "Resource Management in Wide-Area ATM Networks Using Effective Bandwidths", *IEEE JSAC*, vol. 13, no. 6, August 1995, pp. 1081 - 1090
- [3] J. J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks", *Proceedings of IEEE*, vol. 79 no.2, February 1991, pp. 170 - 188
- [4] The ATM Forum, *User-Network Interface Specification Version 3.0*, Prentice Hall.
- [5] The ATM Forum, "Traffic Management Specification Version 4.0", *ATM Forum/af-tm-0056.000*, April 1996.
- [6] ITU-TSS, "Traffic control and congestion control in B-ISDN", *Recommendation I.371*, Geneva, June 1992.
- [7] J. Y. Hui, "Resource allocation for broadband networks", *IEEE JSAC*, vol. 6, no. 9, December 1988, pp. 1598-1608
- [8] A. Elwalid and D. Mitra, "Analysis, approximations and admission control of a multi-service multiplexing system with priorities", *Proc. of IEEE INFOCOM'95*, Boston, April 1995, pp. 463 - 472
- [9] G. Kesidis, J. Walrand and C. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources", *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, August 1993, pp. 424-428
- [10] G. Veciana and G. Kesidis, "Bandwidth Allocation For Multiple Qualities of Service Using Generalized Processor Sharing", *IEEE Globecom'94 Conference Record*, San Francisco, November 1994, pp. 1550-1554

- [11] D. D. Clark, S. Shenker and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", *ACM SIGCOMM Computer Communications Review*, vol. 2, no. 4, October 1992, pp. 14-26
- [12] A. Demers, S. Keshav and S. Shender, "Analysis and simulation of a fair queueing algorithm", *Proceedings of SIGCOMM Computer Communications Review*, vol. 19, no. 4, September 1989, pp. 1-12
- [13] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet-Switched Networks", *IEEE/ACM Trans. on Computer Systems*, vol.9, no. 2, May 1991, pp. 101-124.
- [14] A. K. Parekh and G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case", *IEEE/ACM Trans. on Networking*, vol.1, no.3, June 1993, pp. 334-357
- [15] W. E. Leland, M. S. Taggu, W. Willinger and D. V. Wilson, "On the Self-Similar nature of Ethernet Traffic", *ACM/IEEE Trans. on Networking*, vol. 2, no. 1, February, 1994, pp. 1-15
- [16] Bell Communications Research, "Broadband Switching System (BSS) Generic Requirements", *GR-1110-CORE*, Issue 1, September 1994.
- [17] S. Attal, M. Devetsikiotis, I. Lambadaris and C. Huang, "Traffic Generation for Broadband Network Testing Using the GN Nettest interWATCH 95000 Test Set", *Participants Proceedings of 4th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, Ilkley UK, July 1996.
- [18] B. Melamed, Q. Ren, and B. Sengupta, "Modeling and Analysis of a Single Server queue with Autocorrelated Traffic" *Proc. IEEE INFOCOM'95*, Boston, April 1995.
- [19] M. R. Ismail, I. Lambadaris, M. Devetsikiotis and A. R. Kaye, " Modeling Prioritized MPEG Video Using TES and a Frame Spreading Strategy for Transmission in ATM Networks", *Proc. IEEE INFOCOM '95*, Boston, April 1995.
- [20] V. S. Frost and B. Melamed, "Traffic Modeling for Telecommunications Networks", *IEEE Communications Magazine*, March 1994, pp. 70-81

- [21] B. T Doshi, "Deterministic rule based traffic descriptors for broadband ISDN: Worst case behavior and concept equivalent bandwidth", *IEEE GLOBECOM'94*, San Francisco, November 1994, pp. 1759-1764
- [22] A. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks", *IEEE/ACM Trans. on Networking*, vol.1, no. 3, June 1993, pp. 329-343
- [23] A. Elwalid, D. Mitra and R. Wentworth, "A New Approach for Allocating Buffers and Bandwidth to Heterogeneous Regulated Traffic in an ATM Node", *IEEE JSAC*, vol.13, no.6, August 1995, pp. 1115-1127
- [24] Z. Gand and A. Acampora, "Effective Bandwidth for Heterogeneous Sources in ATM Networks", *Proc. IEEE ICC'94*, New Orleans, May 1994.
- [25] R. Gibbens and P. Hunt, "Effective bandwidths for the multi-type UAS channel", *Queueing Systems*, 9, pp. 17-28, 1991.
- [26] R. Guerin, H. Ahmadi and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks", *IEEE JSAC*, vol.9, no.7, September 1991, pp. 968-981
- [27] F. P. Kelly, "Effective Bandwidths at Multi-class Queues", *Queueing Systems*, 9, pp. 5-16, 1991.
- [28] J. Leventovszky, E. C. van der Meulen, S. Imre and B. Varga. "Call Admission Control for ATM Networks by Source Distribution Transformations", *Participants Proceedings of 4th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, Ilkley UK, July 1996.
- [29] Ramesh Nagarajan and James F. Kurose, "On Defining, Computing and Guaranteeing Quality-of-Service in High-Speed Networks", *IEEE INFOCOM'92*, Florence Italy, May 1992, pp. 2016-2025
- [30] G. Kesidis, *ATM network Performance, A First Course*, Chapter 3, 1996.
- [31] MIL3, Inc, *OPNET 2.5A manual*, 1989-1994.

- [32] Buarne E. Helvik, Ole Melteig and Leif Morland, "The Synthesized Traffic Generator: Objectives, design and capabilities" Paper presented at *Integrated Broadband Communication Networks and Services (IBCN&S) Copenhagen*, Denmark, April 1993.
- [33] A.I. Elwalid and D. Mitra, "Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic", *Proc. IEEE INFOCOM'92*, Florence Italy, May 1992, pp. 415-425
- [34] G. Wu and J. W. Mark, "A Buffer Allocation Scheme for ATM Networks: Complete Sharing Based on Virtual Partition", *IEEE/ACM Trans. on Networking*, vol.3, no.6, December 1995, pp. 660-670
- [35] B. Khasnabish, "Broadband to the Home (BTTH): Architectures, Access Methods, and the Appetite for it", *IEEE Network*, January/February 1997, pp. 58-69
- [36] F. Kamoun and L. Kleinrock, "Analysis of shared finite storage in a computer network node environment under general traffic conditions", *IEEE Trans, on Commun.*, vol. COM-28, no.7, July 1980

IMAGE EVALUATION
TEST TARGET (QA-3)

APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved